Real-Time Processing and Visualization of 3D Time-Variant Datasets

Mai H. Elshahali

Submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Applications

Denis Gračanin, Chair
Hicham G. Elmongui
Yong Cao, co-chair
Chris L. North
Krešimir Matković

July 31^{st}, 2015

Blacksburg, Virginia

Keywords: Visualization, User Interaction, Parallel Processing

Real-Time Processing and Visualization of 3D Time-Variant Datasets

Mai H. Elshahali

(ABSTRACT)

Scientific visualization is primarily concerned with the visual presentation of three-dimensional phenomena in domains like medicine, meteorology, astrophysics, etc. The emphasis in scientific visualization research has been on the efficient rendering of measured or simulated data points, surfaces, volumes, and a time component to convey the dynamic nature of the studied phenomena. With the explosive growth of the size of the data, interactive visualization of scientific data becomes a real challenge. In recent years, the graphics community has witnessed tremendous improvements in the performance capabilities of graphics processing units (GPUs), and advances in GPU-accelerated rendering have enabled data exploration at interactive rates. Nevertheless, the majority of techniques rely on the assumption that a true three-dimensional geometric model capturing physical phenomena of interest, is available and ready for visualization. Unfortunately, this assumption does not hold true in many scientific domains, in which measurements are obtained from a given scanning modality at sparsely located intervals in both space and time. This calls for the fusion of data collected from multiple sources in order to fill the gaps and tell the story behind the data.

For years, data fusion has relied on machine learning techniques to combine data from multiple modalities, reconstruct missing information, and track features of interest through time. However, these techniques fall short in solving the problem for datasets with large spatio-temporal gaps. This realization has led researchers in the data fusion domain to acknowledge the importance of human-in-the-loop methods where human expertise plays a major role in data reconstruction.

This PhD research focuses on developing visualization and interaction techniques aimed at addressing some of the challenges that experts are faced with when analyzing the spatio-temporal behavior of physical phenomena. Given a number of datasets obtained from different measurement modalities and from simulation, we propose a generalized framework that can guide research in the field of multi-sensor data fusion and visualization. We advocate the use of GPU parallelism in our developed techniques in order to emphasize interaction as a key component in the successful exploration and analysis of multi-sourced data sets. The goal is to allow the users to create a mental model that captures their understanding of the spatio-temporal behavior of features of interest; one which they can test against real data measurements to verify their model. This model creation and verification is an iterative process in which the user interacts with the visualization, explores and builds an understanding of what occurred in the data, then tests this understanding against real-world measurements and improves it.

We developed a system as a reference implementation of the proposed framework. Reconstructed data is rendered in a way that completes the users' cognitive model, which encodes their understanding of the phenomena in question with a high degree of accuracy. We tested the usability of the system and evaluated its support for this cognitive model construction process. Once an acceptable model is constructed, it is fed back to the system in the form of a reference dataset, which our framework uses to guide the real-time tracking of measurement data. Our results show that interactive exploration tasks enable the construction of this cognitive model and reference set, and that real-time interaction is achievable during the exploration, reconstruction, and enhancement of multi-modal time-variant three-dimensional data, by designing and implementing advanced GPU-based visualization techniques.

Dedicated to Khaled, Karma, and Adam

Acknowledgments

It is hard to find enough words of gratitude to thank my PhD advisor Dr. Denis Gračanin for his invaluable support throughout this PhD research. It has been both an honor and a pleasure to work with him over the years, first for my Master degree, then my PhD. During this time, Dr. Gračanin has taught me a lot not by lecturing so much as by doing. On an academic level, he allowed me to grow as a researcher, taught me how to think independently, and how to conduct effective research. On career level, his guidance and advice have been crucial in giving me a sense of direction on what my skills are, and where they can be put to use. Finally, on a personal level, I have learned so much from his charisma and the amount of respect he treats everyone with. I am forever grateful for his support, and I hope that one day, I will be able to teach what I learned from him to my students back home.

Dr. Hicham Elmongui has been an asset to my PhD research. He always had the most uplifting and encouraging words to push me forward in my research and in my career. When I first met Dr. Hicham in Egypt, I was excited to know that we could both speak French. As he became my co-advisor, I have learned that there is much more in common background between us, which has led to a very fruitful advisor-advisee relationship.

It has also been my pleasure to work with Dr. Yong Cao and his lab when I first arrived in Blacksburg. He generously allowed me to attend his group meetings, which have been an incredibly rich resource for me, that kept me up to date on Computer Graphics and Visualization research. The fast pace of this extremely energetic group was a major driving force in my research. Dr. Cao always made sure to challenge each and every one of his students and bring out the best in each one of us. Every meeting with him was a brainstorming session, for which the outcome was many creative ideas.

Dr. Kresimir Matković has been involved in my research from day one. He always had very insightful comments that helped me think deeply about the assumptions I made and the hypotheses I tested. His feedback and questions have been priceless in pinpointing potential pitfalls and focusing on ensuring accuracy of results. I am very thankful to have him on my committee and to have had the opportunity to collaborate with him in different projects.

I am also especially grateful for having Dr. Chris North on my committee. Dr. North's feedback has been instrumental in shaping the final outcome of my research. His focus on user experience helped steer my efforts in the right direction to ensure system usability. His guiding comments and the discussions I have had with him have been a wonderful learning experience.

In addition, I would like to express my deepest gratitude to Dr. Mohamed Gad of the Civil Engineering Department in Ain Shams University. As a domain expert in atmospheric data and geoscience, Dr. Gad played a tremendous role in conveying domain challenges, and opportunities to overcome those challenges. His positive feedback has constantly given momentum and motivated research and implementation progress.

Last but not least, I would like to thank Matthew Carter, Junpeng Wang, Run Yu, James McClure, and Peter Radics for several brainstorming discussions that helped me better advocate my ideas and hunt down implementation issues. PhD research is a challenging endeavor that was made more enjoyable with the help of these wonderful people.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"To know means to record in one's memory; but to understand means to blend with the thing and to assimilate it oneself."

The Temple of Karnak. Luxor, Egypt. 3200 BC.

Scientific visualization deals with the *exploration* and *analysis* of datasets acquired from measurements or simulation of real-world phenomena. While the purpose of visualization may be creating images, the goal of *exploration* is to build understanding [LRTM07], and the goal of *analysis* is to provide insight [SBM92]. Creating visualization and interaction techniques that enable user's understanding of physical phenomena and provide insight is a challenging research endeavor.

Research in general is often divided into three categories: *basic*, *transitional*, and *applied*. Visualization is a field that primarily aims at introducing novel techniques as well as exploring new phenomena. Therefore, the NIH/NSF report by Jonhnson et al. [JMM+06] describes visualization research as a field that creates cyclic relationships between these divisions, as shown in Figure 1.1. In *basic* visualization research, areas of interest include, for example, cognitive psychology, vi-

sual perception, psychophysics, etc. Such areas inform visualization research by shaping Human-Computer Interaction (HCI) paradigms, and formulating guidelines to visualization system design.

On the opposite end of the spectrum, *application-driven* research is steered by data and domain-specific tasks. These tasks specify priorities and requirements for the system and set validation criteria for the developed techniques. Application areas that are frequently served with visualization include computational fluid dynamics, medical imaging, meteorology, and software engineering as examples.

The bulk of work for visualization researchers, however, lies between these two extremes, namely, in the *transitional* space, in which techniques are created and refined. The attention of transitional visualization research is focused on incrementally refining a narrow set of techniques. Informed by the basic sciences, and the application domain requirements, visualization is almost never a stand-alone process. It is often necessary but not sufficient to solve problems. Visualization research is, therefore, a cross-disciplinary approach, in which other analytic tools and techniques, such as statistics, data mining, parallel processing, and image processing, must be combined together to facilitate both qualitative and quantitative analysis.

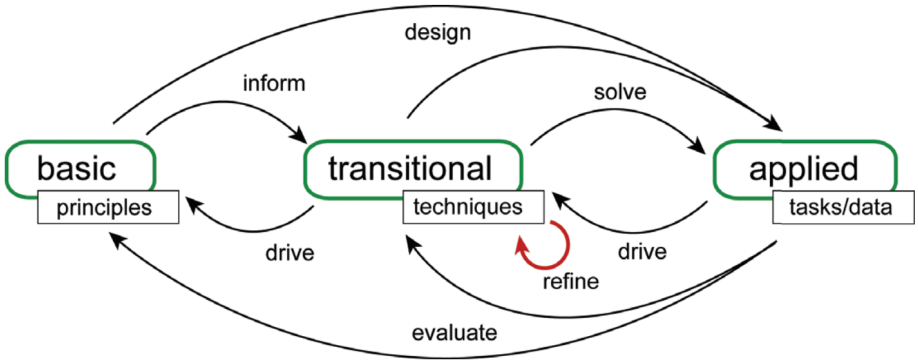


Figure 1.1: The cyclic relationships between the basic, transitional, and applied research [JMM+06].

As a sub-field of visualization research, scientific visualization includes all cycles in the diagram shown in Figure 1.1, targeting basic, transitional, and application-driven research. However, some of the widely adopted definitions do not quite capture this cyclic process; consequently, missing important challenges in the field. For example, according to Friendly et al. [FD08], Scientific Visualization is *"primarily concerned with the visualization of three-dimensional phenomena (architectural, meteorological, medical, biological, etc.), where the emphasis is on realistic renderings of volumes, surfaces, illumination sources, and so forth, perhaps with a dynamic (time) component"*. This description sets the focus almost exclusively on the technique-driven space of transitional visualization research, in which rendering techniques are developed and refined to achieve realism or to improve performance. With little or no regard to the basic science that informs the design of visualization systems or to application domains that drive the need for such systems, visualization can be characterized as merely a channel for data communication [VRFW14].

In contrast, an earlier characterization by Springmeyer et al. [SBM92] described scientific data analysis as *"the process of distilling potentially large amounts of measured or calculated data into a few simple rules or parameters which characterize the phenomenon under study"*. They argued that the more appropriate focus for scientific data analysis interfaces should reflect the broader process which includes how scientists interact with the data before, during, and after image generation. This broader view of the field is also adopted by Wright [Wri07]. She presents a clear distinction between the field of computer graphics, for which the main outcome may be pictorial representations of data, and that of scientific visualization, for which the outcome is an *understanding* of what produced the data rather than just a means to present it.

We adopt a similarly broad definition of scientific visualization as: "a branch of computer science with the purpose of graphically representing scientific data in a way that enables the scientist to *understand*, *illustrate* and *gain insights* into the phenomena of interest". In light of this latter statement, we argue that the emphasis is not on realistic rendering, contrast to Friendly's definition. Rather the focus is on the ability of the visualization to tell the story behind the data and enable the

user to incorporate it into their knowledge of the real world. In some cases, this requires photo-realism but in other cases illustrative techniques can convey the message better and, therefore, expensive realistic rendering calculations can be avoided.

To support users' understanding of the dynamic evolution of physical phenomena, time-enabled visualization illustrates data taken over *time*. Once captured, this evolution helps scientists explain the dynamics of such phenomena and aids the construction of reference and predictive models. Examples include cloud formation, weather predictions, hurricane warnings, environmental studies about ozone hole changes over time, air motion around an aircraft for better design and control, as well as medical scans taken over time. In many areas of research, time-varying data is collected through sensors at sparsely distributed spatio-temporal locations. In such cases, displaying data directly does not provide enough information to gain that understanding.

The problem of reconstructing the complete model describing a physical phenomenon from sensor data is theoretically insoluble just like the problem of obtaining a $3D$ scene from a $2D$ image. However, we use certain assumptions about, for example, the size of people, the shades, etc. from our knowledge of the physical world around us to interpret the scene. We start by understanding what is depicted, then link this understanding to prior knowledge, and fill in any gaps with assumptions to construct an as complete as possible cognitive model of the scene. In this dissertation, we propose ideas, inspired by this cognitive model construction process, to overcome the scarcity of collected data in the spatio-temporal domain. By advocating exploratory visual analysis as a means to *understand* the data, link it to prior experience, and form assumptions, we support the creation of a cognitive model that encompasses this knowledge. The next step is then to communicate this cognitive model back to the visualization in the form of a reference dataset, that can be used to create a visual representation with high information content about the events in question.

In this Chapter, we start by characterizing scientific data visualization in terms of three axial data characteristics in Section 1.1. Next, we motivate the problem with an example and present our rationale for tackling it from a high performance graphics perspective in Section 1.2. The proposed research methodology, visualization goals, research hypotheses, and questions are then described in Section 1.3. Finally, the contributions of this PhD research are stated in Section 1.4.

1.1 Data Characteristics

The problem of visualizing scientific data can be characterized in light of three axial data characteristics. These characteristics classify work in the literature, and impose different priorities on research in the field. Specifically, scientific data quantifies physical phenomena that occur in the three-dimensional world we live in. Data is collected or generated over time, and at every point, several variables may be measured by different modalities or calculated by simulation. Therefore, the three axial characteristics of scientific data are:

1. Volumetry
2. Time-variance
3. Multidimensionality and multimodality

Each one of these data characteristics poses a number of challenges to visualization approaches. In this Section, we explore the challenges that arise from each of these characteristics, and their implications on research.

1.1.1 Volumetry

Several challenges arise from the fact that the human eye perceives the world in three dimensions. A question then becomes: how to develop graphics techniques that project 3-Dimensional

structures onto 2-Dimensional displays, while perceivably conveying the correct spatial and temporal correlations among dynamic volumetric objects? Correct depth perception is crucial in understanding spatial relations among structures. For example, in pre-surgical planning doctors need to infer spatial relations between a tumor and surrounding structures to avoid mistakes during surgery. Traditionally, volume visualization techniques provided extra illustrative depth cues such as halos and shades to deliver correct depth order. Recent advances in the field, seek to avoid the overhead of rendering these extra cues through the development of appropriate enhancement techniques [GRT13].

User interaction is another way to alleviate the problem through rotation and free navigation in the scene. However, the freedom offered by interaction techniques can be a double-sided coin as it gives the user the ability to fully explore the details of a visualization, but can be overwhelming and confusing. Generally speaking, interaction in scientific visualization techniques is affected by users' level of expertise in the field, their familiarity with the dataset as well as with the visualization strategies available. Special care needs to be given to the design and implementation of interactions in the exploration phase of the analysis in order to steer users' efforts toward generating informative visualizations.

1.1.2 Time-Variance

The study of dynamic behavior of a volume's features of interest can uncover several mysteries in today's scientific challenges. In some applications (e.g. medical scans), time-varying datasets are assumed to be scalar values assigned to points or geometries in $3D$ space. A sequence of such scalar fields can be registered and processed in a way to track changes that features of interest (e.g. tumors) undergo over time. Visualization efforts for vector fields make a slightly different assumption [HH89]. Measurements of magnitude and orientation are known at each point in $2D$ or $3D$ space. The space is then sampled either randomly or using more sophisticated seeding strategies to decide where representative geometries (e.g. streamlines, pathlines, vectors, or tubes)

should be rendered in order to convey the field's trends and features of interest while reducing clutter. Vector fields represented by these streamlines or pathlines are commonly generated by simulation.

Scientists perform large-scale numerical simulations to understand models underlying the data and to investigate phenomena. Such simulations tackle increasingly ambitious yet demanding problems that require modeling of complex $3D$ geometries or incorporate inherently unsteady $3D$ physics. A challenge arises from the need to optimize simulation parameters and engineering designs that target the underlying phenomena. Several simulation iterations are run and insights are acquired on how to vary boundary conditions (independent variables) to reach optimum results. Visual analysis can be of great benefit in creating hypotheses and rules that guide the optimization. Research in visualization of time-varying multi-volume datasets is motivated by the need to aid this iterative process by providing insight on current simulation design and setting up hypotheses about the correlations among different measurements and simulation runs.

The main aim of time-varying visualization techniques is then to provide the scientist with a set of tools to identify and highlight the evolutionary patterns of physical phenomena. To that aim, the tools must enable tasks like identifying regions of interest, abstracting them for simplified representations, and tracking their evolution over time. Augmenting simulation results with ground truth measurements obtained via scanning devices can increase the accuracy of this tracking process, and highlight shortcomings of the current simulation model to pinpoint improvement opportunities.

1.1.3 Multidimensionality and multimodality

The effective visual presentation of multidimensional multi-sourced data while supporting visual analytic tasks at interactive rates is no simple task. Several challenges emerge from the large size of the data, its multi-dimensional nature, in addition to its spatio-temporal complexity. Further, data

obtained from multiple sources at different scales and resolutions creates gaps in the visualization and introduces the problem of missing information. The main goal of data fusion is to fill the gaps in data obtained from different sources, in order to present a more complete visual representation of the physical phenomena in question with the highest possible information content. Nevertheless, the generation of such representation is not trivial.

For years, fusion research has adopted machine learning techniques to integrate multimodal information. However, researchers in the field have highlighted the shortcomings of these techniques alone in coping with the increased complexity of multimodal data, and have called for a human-in-the-loop fusion model [HHT00]. Consequently, research has witnessed a shift of interest toward "high level" fusion which addresses the problem from a human-computer interaction perspective [FN13]. This has been reflected on technical discussions in the field of information fusion, as can be seen in Figure 1.2. A similar trend has paralleled this one in the scientific visualization community, as state-of-the art research has targeted more datasets involving multiple modalities to create information-rich visualizations.

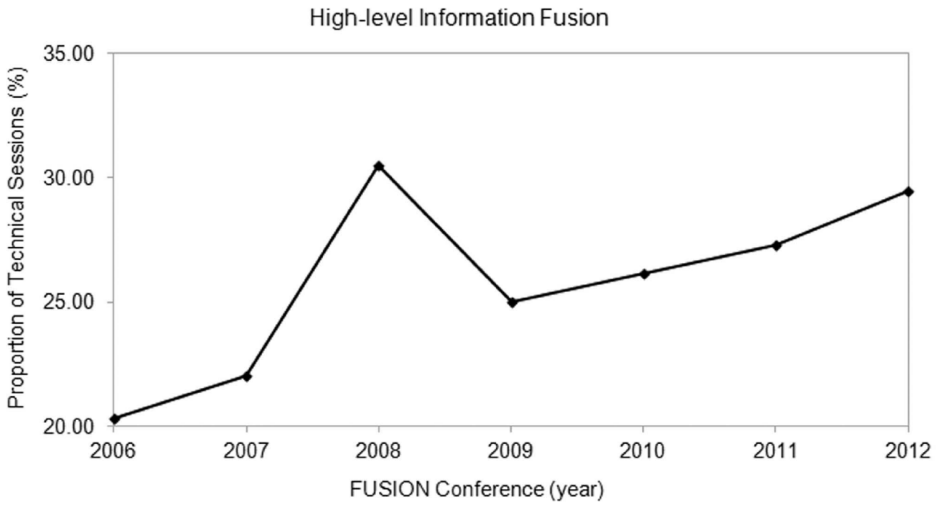


Figure 1.2: The proportion of technical sessions on high level information fusion at FUSION conferences [FN13].

1.2 Motivation

Computer graphics have brought several rewards to the modern society. In addition to dazzling entertainment, and quality information visualization, human-computer interaction techniques have evolved significantly over the past years. However, we still have a long way to reach the point where the amazing rewards of visual computing reflect on every aspect of our daily lives; one that can only be made shorter by simultaneously deploying existing techniques, while developing new ones to make better use of recent high-end graphics processing units. The challenges described in Section 1.1 and the realization of how crucial visualization is to the fusion of multiple data sources to understand physical phenomena, strongly motivate us to propose a complete solution framework that aims at the interactive visual fusion of heterogeneous datasets to support the user-aided reconstruction of missing information. Before we go into the details of the proposed framework, however, let us first answer a few motivating questions.

1.2.1 Why Three Dimensions?

Volume and Multi-volume data are becoming more and more important in several domains:

1. **Scientific Visualization:**
   As noted earlier, scientific visualization deals primarily with the exploration and analysis of datasets arising from measurements or simulation of real-world phenomena. Scientists perform large-scale numerical simulations to understand models underlying the data and investigate phenomena. Examples include computational fluid dynamics (CFD), geological and seismic simulations. Research in multi-volume rendering and visualization of time-varying volumes is motivated by the need to aid the iterative process of refining the simulation parameters, through providing insight on the current design and setting up hypotheses. A visualization technique should offer limited visual overlap, fast learning, and good recall.

2. **Medical Imaging:**

   Scans of the interior of the human body such as computed tomography (CT), Magnetic Resonance Imaging (MRI), and other modalities result in large sets of $3D$ data. The correct interpretation and understanding of the results of such scans can save lives. Visualization in medicine, or, for short, medical visualization is a special area of scientific visualization [PB07]. Given that data exploration, hypothesis testing, and presentation of results are the general goals of scientific visualization, many examples in medical visualization target these very same goals. Whether a patient suffers from a certain disease is a hypothesis to be tested. Adequate data representation is crucial to the correct exploration and interpretation of the scanned structures. For example, in pre-surgical planning and image-guided surgery, the physician's correct inference of spatial relations between body structures (e.g. important vessels and tumors) is critical to avoid surgical mistakes that could lead to severe complications. The multi-structure nature of the human body and the fine features of complex structures, like the brain, have motivated a large body of research in both direct and indirect volume rendering techniques (see Section 4.4) aiming to offer adequate $3D$ visualizations and interaction techniques that accurately and correctly convey information acquired through different medical imaging modalities.

3. **Computer Graphics:**

   The game industry has been a driving force for research in high performance graphics, taking Graphics Processing Units (GPUs) from fixed function devices aiming to produce quality images on a computer screen to general purpose parallel devices capable of performing massive calculations in real-time. Also, an important factor in this progress is the large body of game users population and their constantly increasing expectations for a faster, more interactive, and more convincing graphics experience. Subsequently, as more computational power becomes available, sophisticated rendering techniques have also become more feasible, leading to higher fidelity and better perception of the visualized $3D$ models. Volume data is ideal to model fluids, gases, and natural phenomena such as fog, fire, clouds, etc. In such cases, surface-based approaches, also called indirect volume rendering, lack the abil-

ity to account for light interaction taking place in the interior of an object. The choice of graphics representations of the data in $3D$ is therefore an active area of research that seeks to develop novel rendering techniques while providing high information content.

1.2.2 Why Multiple Sources?

The motivation for combining data collected from multiple sources (e.g. satellites, sensors, simulation, etc) emerges from the limitations of measuring instruments. Coverage in space and time, natural variability, and the instrument's ability to focus on a particular Region Of Interest (ROI), are all examples of such limitations. To illustrate with a real-world situation, let us describe a physical phenomenon example that is of interest to experts in the atmospheric science domain; and let us explore the available options for scientists to study this phenomenon and its dynamic behavior.

Example: Calbuco Volcanic Eruption 2015

On April 22nd 2015, Chile's Calbuco volcano erupted for the first time since 1972 resulting in an ash plume that was ejected at least 15 kilometers above the volcano. Thousands of people were evacuated from areas within 20 kilometers radius around the eruption location.

Volcanic ash constitutes of jagged pieces of rocks, minerals, and volcanic glass the size of sand and silt [USG]. The typical size of an ash particle is less than 2 millimeters (1/12th of an inch). Very small particles can be less than 0.001 millimeters (1/25,000th of an inch). Once in the air, ash particles can be carried by the wind for thousands of kilometers away from the eruption location. These hard particles do not dissolve in water, and conduct electricity when wet. They pose severe danger to aircraft. Therefore, the airline industry seeks regularly updated reports and maps of ash cloud location, altitudes, and concentration every 6 hours to make informed decisions about flight restrictions or cancellations. So how do they obtain this information?

Satellite Imagery makes the most important source of information for ash cloud localization and tracking. Other sources of information include public reporting of ash fall and news agencies. Visible light images provide a good source for experts to observe the shape of ash cloud at specific times when the satellite covers the affected regions, and when daylight is available at the time of coverage. Figure 1.3 shows an example image captured by the Moderate Resolution Imaging Spectroradiometer (MODIS) instrument aboard two different satellites at two different capture times. NASA's Terra satellite caught the phenomenon at 14:20 UTC on April 23rd, while NASA's Aqua satellite provided coverage about four hours later at 18:35 UTC of the same day[1].

What happened in the real world before and after the times of satellite coverage of the region of interest (ROI) and during the night cannot be deduced from these images alone. Therefore, despite their usefulness for static imaging, experts cannot rely on them for dynamic behavior analysis. This scarcity of information makes visible light images alone a poor choice for the study of time-variant phenomena such as volcanic ash clouds.

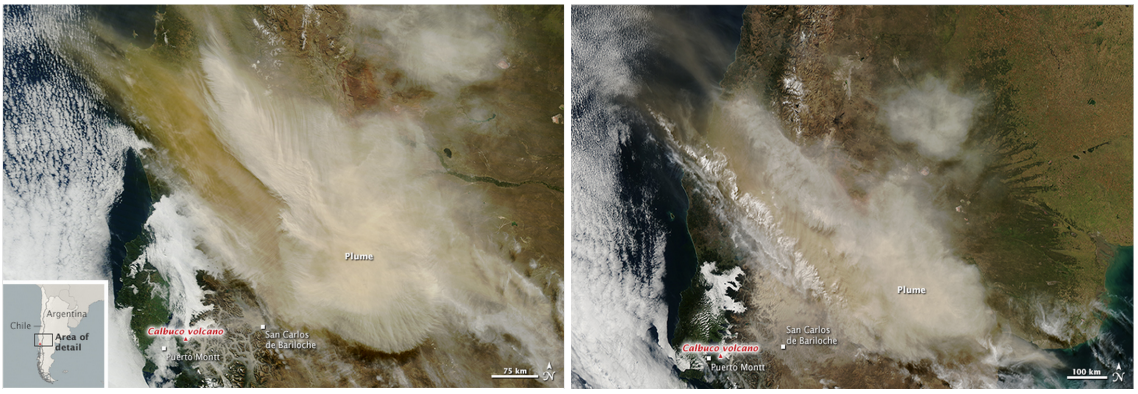


Figure 1.3: Visible light image of the Calbuco ash cloud, taken at 14:20 UTC on April 23rd by MODIS instrument on NASAs Terra satellite (left), and one acquired at 18:35 UTC of the same day by the MODIS instrument on NASAs Aqua satellite. Source: NASA Earth Observatory.

[1]Note that the second image is at a wider scale than the first.

In contrast to visible light images, Infrared (IR) sensors on satellites are sensitive to the infrared radiation waves that are generated by all object on Earth and in the atmosphere. These radiations are generated at all times, which makes infrared measurements available for both night and day. The joint NOAA/NASA Suomi NPP satellite observed atmospheric waves above Calbuco and its plume. An image from the Visible Infrared Imaging Radiometer Suite (VIIRS) on NASA's Suomi NPP satellite is displayed in Figure 1.4, and shows the heat signature of the hot ash in longwave infrared (11.45 micrometer channel).

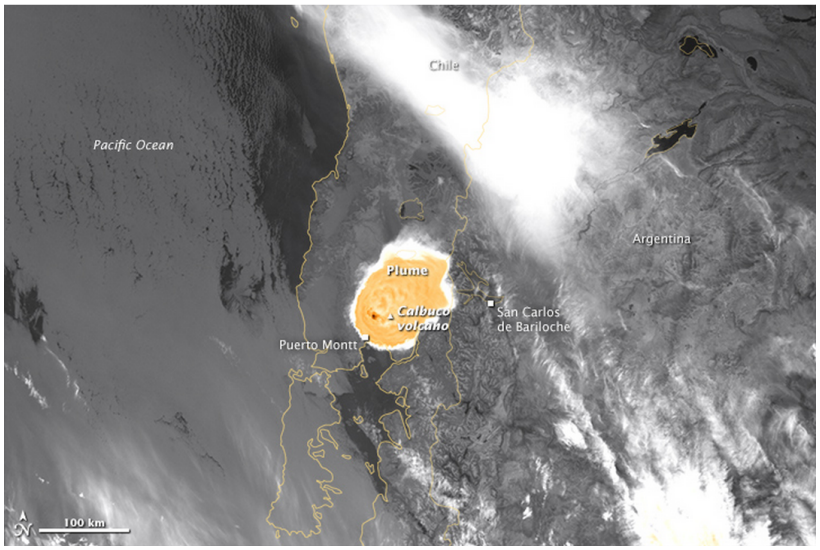


Figure 1.4: Nighttime view of the eruption in the early morning hours of April 23rd captured by VIIRS instrument aboard NASA's Suomi NPP satellite. Source: NASA Earth Observatory.

Despite the availability of infrared data, a different set of limitations is introduced. As Stevenson [SLF+13] argued, monitoring and tracking ash from infrared data is extremely difficult. First of all, the data has no altitude information. Domain experts make certain assumptions from their knowledge about the real world to estimate altitude information. For example, if the experts knows how the temperature of the atmosphere changes with altitude, e.g. from weather balloon (radiosonde) data, and if she assumes that the ash cloud is at the same temperature as the atmosphere around it, then she can convert the Brightness Temperature (a value that is based on the amount of

energy an object radiates) into an estimate of the altitude of the cloud.

Other techniques for altitude estimation include the integration of information from laser firing satellites like CALIPSO which create vertical profiles at specific locations (see Figure 1.5 for an example). Combining the two information sources does not create a full three-dimensional plume but can help the expert build a cognitive model based on their assumptions and based on information obtained from these different modalities.

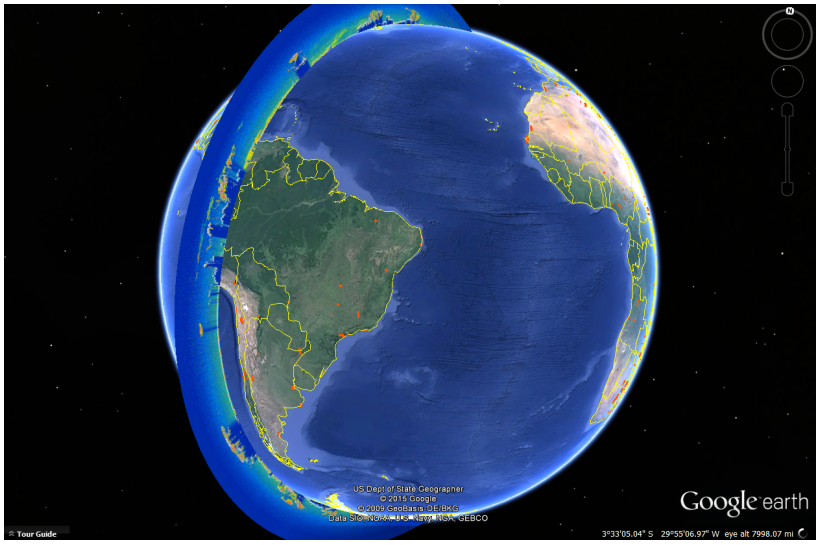


Figure 1.5: Vertical profiles near the eruption on April 23rd 2015. Google Earth file. Source: NASA LIDAR LEVEL 1 BROWSE IMAGES - 2015-04-23 06:00:00Z - SECTION 1

A second limitation arises from the orbiting patterns of satellites. Instruments are installed on polar satellites that orbit the Earth and capture measurements directly below the satellite. At that time, ash particles that exist in the atmosphere away from the area covered by the satellite's orbit are missing. However, an expert can make assumptions about the existence of detections in different regions of interest if she has an understanding of the cloud's dynamic behavior. An animation depicting cloud movement can help build understanding of cloud dynamics [SSEH03].

Figure 1.6 displays sample snapshots from an animation created from a sequence of images captured by NASA's Aqua satellite between April 23rd and May 4th of 2015.

The data shown in Figure 1.6 has several discontinuities, that are due to spatial gaps between orbits. Each one of the images represents 12 hours of coverage. If ash detections (shown in red) were captured at the same time, the expert would probably be able to mentally fill those gaps and build their own cognitive image of what the atmosphere looks like in those out of coverage areas. However, this is not the case. In fact the measurements taken in one of the orbits can have time stamps that are order of hours away from the next orbit, making the temporal gaps between them an even tighter limitation. This is due to the fact that the satellite covers the same location on Earth every 12 hours. A lot may have changed during this time, making it very hard to track the plume evolution.

Other motivational examples of visual fusion can be found in the biomedical domain, where combining multiple volumes to create a visualization can be of great value. In genomic imaging, for instance, gene expression energy volumes need to be combined for thousands of genes in order to contrast expression levels and locations relative to one another. Combining volumes from multiple genes with reference volumes can answer questions about structural and directional patterns and their consistency across time.

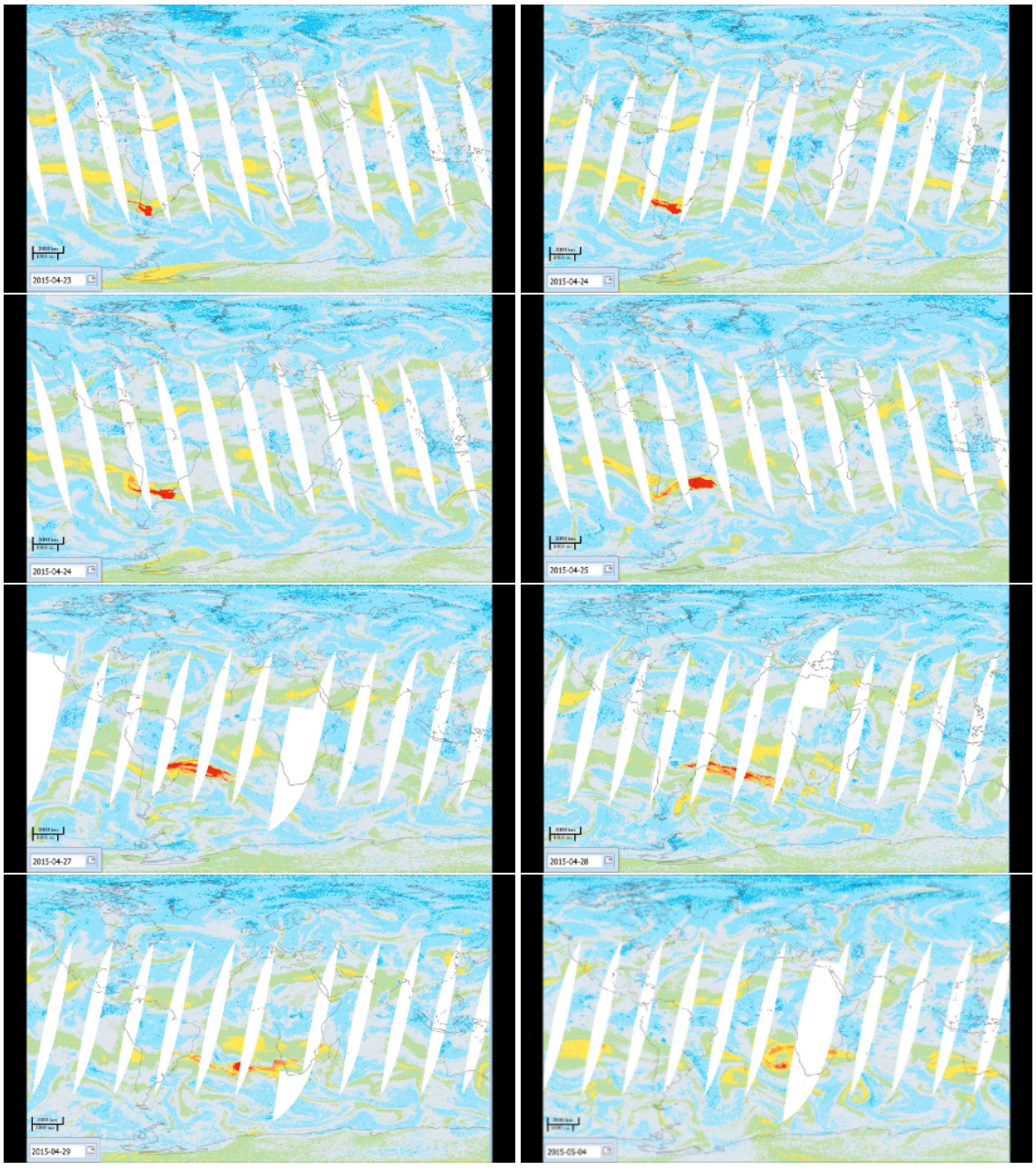


Figure 1.6: Sample snapshots from an animation sequence captured by NASA's Aqua satellite covering the period from April 23rd to May 4th of 2015. Ash cloud resulting from the Calbuco volcanic eruption is shown in red.

Table 1.1: Review of Themes and Goals of IEEE Scientific Visuslization contest from 2004 to 2014

Year	Theme	Goal	Dataset(s)
2004	Visualization Fusion	Combine multiple Vis. techniques to communicate dataset behavior	Single
2005	Rendering Evolution	Non-photorealistic visualization, global illumination, and multi-modal depictions	Single
2006	See what's shaking	Answer real domain-science questions on real datasets	Single
2008	Multifield 3D scalar data	Design a vis. that effectively answers domain-science questions	Multi
2010	Neurosurgical Planning	Identify risk structures in the brain and how they interact	Single
2011	Stability Visualization of Fluid Dynamics	Explore stability of fluid dynamics simulation w.r.t. models of turbulences	Multi
2012	Computational material science	Detect and analyze phase transitions	Single
2013	Mouse brain atlas	Identify spatial and temporal gene expression patterns	Multi
2014	Volcanic eruptions and atmospheric aftermath	integrate advantages from different modalities to provide holistic view	Multi
2015	Visualizing the Universe	Create a framework for the visualization of cosmological simulations	Multi

Over recent years, a paradigm shift in scientific visualization research has set more focus on the idea of visual fusion, where multiple data sources contribute to the generation of the visualization in order to reconstruct data and provide higher information content. We conclude from the leading scientific data visualization contest datasets and tasks over recent years (see Table 1.1) that an observable shift has taken place from single-volume or single-source data to data obtained from multiple sources that complement one another to accomplish visual analytic tasks. A challenge then remains of how to effectively combine data from these multiple datasets in a way that does not overload the user's cognitive perception and that delivers the message about the phenomena in question. This challenge leads us to pose our third motivational question.

1.2.3 Why GPU Acceleration?

The problem of visualizing data with large spatio-temporal gaps and missing attributes requires the involvement of domain experts at different stages of the fusion and analysis process. Keeping the human in the loop is a challenge that is not yet overcome by existing mainstream visualization tools. In a typical scenario, the expert would write a query to filter the data, then click a button

to execute the query. After that, the wait time for the results to be retrieved and rendered grows proportionally with the size of the data.

User involvement in data reconstruction and tracking requires the construction of a cognitive model that the user builds and refines from her understanding of the data, based on exploration. Interaction with the data, generating hypotheses and testing them in real-time has been proven to be a crucial part of exploratory analysis to enable this understanding [KLM+08] [TLLH13]. The question of whether such interaction is feasible with larger sizes of scientific datasets is central in Computer Graphics research.

Parallel programs running on multi-core CPUs and many-core GPUs tend to outpace the More's law of exponential speed growth [KWm12]. As of 2015, some of the fastest multi-core CPU processors, like Intel Xeon E5 v3 family, have a theoretical peak performance of 992 GFLOPS; while GPUs are considerably more powerful. For example, NVidia Titan Z GPU processors perform around 8.74 TFLOPS in single precision calculations. One might think that at this pace, we can rely solely on this significant increase in processing powers and let complex model visualization research questions be answered by such increase. This is a misleading idea, because this performance boost alone can never transform scientific data visualization problems into memories of the past. One reason for this is the fact that increased computing power also allows users to produce more and more complex simulation datasets. Another reason is that memory bandwidth and data access speed grow at significantly slower rates than processor speed, which makes memory access a major bottleneck. In general, the typical access times of different levels of memory hierarchy vary by orders of magnitude, and the growth rate in access times for main memory and disk has not exceeded 7-10% in the last two decades. Due to this computational gap between processing power and access time, the impact of GPU performance increase is amortized on the application domain, and massive data cannot be interactively rendered by brute force methods. Research in the visualization of complex dynamic three-dimensional datasets aims to develop techniques for the careful management of bandwidth requirements, control of working set size, and coherent access.

1.3 Methodology

Following the cyclic relations among visualization research divisions depicted in the diagram in Figure 1.1, our research methodology in this dissertation exhibits a similar pattern. Specifically, we collaborate with domain scientists to identify problems (*applied*), look into foundational areas to set principles and guidelines that inform our framework design (*basic*), then present novel visualization techniques (*transitional*) as components of a reference system that implements the framework and helps address the challenges to solve a real problem.

At a high methodological level, our research approach can be described as **problem-driven** visualization research that follows a similar methodology to that of a design study [SMM12]. The theory behind this methodology is described in detail in Section 5.1.

Starting with a precondition phase, we initiate several interactions with domain experts to identify promising collaboration opportunities and domain challenges (Section 1.3.1). In the core phase, we deduce hypotheses, identify research questions, and propose a framework that defines a process model for the solution to the problem of time-variant multimodal three-dimensional data analysis. Then, we design and develop a system as a reference implementation of the framework (Section 1.3.2). Finally, the analysis phase encompasses our evaluation efforts and reflects on the usability of the framework and the system. This Section gives a summary overview of our solution approach, and describes how different components fit in parts of this dissertation document.

1.3.1 Precondition Phase

As Sedlmair et al. [SMM12] noted, the precondition phase is further divided into three steps: *learn*, *winnow*, and *cast* (a diagram of these phases is depicted in Figure 5.3). In the *learning* phase, we conducted an extensive literature review that categorizes the field of scientific data visualization in light of the axial data characteristics that we described in Section 1.1. Part I of this

dissertation is dedicated to explaining background and related work that has addressed each one of these characteristics.

The *winnow* phase presented the first challenge in this research. Prior to formulating the problem description that our research addresses, we had a few discussions with domain experts from medical, neuroscience, and civil engineering backgrounds. Throughout these discussions, we were able to identify a pattern of challenges that are faced by experts in different domains, all of which appeared to be promising directions for fruitful collaboration.

In the medical domain, scientific data of time-variant volumetric nature is prevalent. Early brainstorming discussions with a radiologist revealed a dire need for interactive visualization techniques that combine multiple sources of such data in order to build understanding of physical phenomena happening in human body, support hypothesis generation about health and disease, and enable testing of the generated hypotheses with treatment plans while monitoring changes. For example, chronic patients frequently undergo serial (sequential) imaging studies in order to assess the progression of disease and response to treatment. Radiologists have a critical role in the process of comparing images of serial studies to detect subtle changes in imaging findings. Whether the disease is progressing, regressing, or stable can be judged by detailed description of imaging parameters such as size of the abnormality or its behavior in terms of enhancement after contrast administration, change in texture, effect on adjacent structures, etc. The process of visually identifying these subtle changes can be very time consuming and inaccurate when dealt with by the human eye alone. Timely detection of such changes is critical to aid clinical decision making regarding drug modification, possible intervention, and predicting outcome especially when dealing with cancer patients, who need long term follow up and early detection of disease fluctuations. Such timely detection calls for extending the measurements provided by scanned images, into three-dimensional volumetric representation of size and shape. Further, continuous animations depicting change with high information content are mandatory to avoid confusion on the clinicians' side.

Biomedical collaboration opportunities were explored from discussions with engineers working on functional neuroimage datasets. One example is the source reconstruction of electromagnetic brain signals measured by the magnetoencephalography (MEG) modality. A functional neuroimaging technique that maps brain activity by recording magnetic fields produced by electrical currents occurring naturally in the brain, using very sensitive magnetometers. Among the available functional imaging techniques, MEG and EEG uniquely have temporal resolution below 100 ms. This temporal precision allows us to explore the timing of basic neural processes at the level of cell assemblies. The challenge posed by MEG is to determine the location of electric activity within the brain from the induced magnetic fields outside the head. Problems such as this, where model parameters (the location of the activity) have to be estimated from measured data (the signals) are referred to as inverse problems. The primary difficulty is that the inverse problem does not have a unique solution (i.e., there are infinite possible "correct" answers), and the problem of defining the "best" solution is itself the subject of intensive research. Possible solutions can be derived using models involving prior knowledge of brain activity. For example, if one's goal is to estimate the current density within the human brain with say a 5mm spatial resolution then it is well established that the vast majority of the information needed to perform a unique inversion must come not from the magnetic field measurement alone, but rather from the constraints applied to the problem by the domain expert.

Functional magnetic resonance imaging (fMRI) is another functional neuroimaging modality that provides a higher spatial resolution but at a lower temporal resolution than MEG. State of the art research in neuroscience seeks to combine the strengths of the two modalities to visualize brain activity at a high spatio-temporal resolution. The estimated MEG source locations can be combined with MRI images to create magnetic source images (MSI). The two sets of data are combined by measuring the location of a common set of fiducial points marked during MRI with lipid markers and marked during MEG with electrified coils of wire that give off magnetic fields. The locations of the fiducial points in each data set are then used to define a common coordinate system so that superimposing the functional MEG data onto the structural MRI data ("coregistration") is possible.

The optimization problem inherent in the inverse model of the MEG beamformer calculates powers based on all sensor readings at each assumed dipole setup. This computational intensive phase is a bottleneck that restricts the scientists' ability to interactively explore the effect of modifying simulation parameters or reconstruction assumptions in real time. Further, once brain sources are configured, one needs to project these results back into sensor space to obtain an understanding of how sensor readings evolve over time due to the calculated active regions. During the *winnow* phase of our solution approach, we implemented a GPU-based prototype to parallelize the MEG beamformer and achieved significant speedup over CPU versions. However, only simulation data was available at the time of this collaboration. Data collection from real subjects using different modalities presented a challenge for timely progress, which led to more search for further collaboration opportunities.

A crucial set of brainstorming discussions we conducted during the *winnow* phase was with an atmospheric scientist from a hydraulics civil engineering background. We explored the problem of visualizing physical phenomena that occur in the atmosphere, and the many complexities that arise from the heterogeneity of the available data, the lack of colocation of information obtained from different satellites, and the need to reconstruct missing information to track the dynamic behavior of interesting phenomena, e.g. ash clouds, over time. Section 1.2.2 described an example of this type of problem.

Having covered a set of domains that can benefit from research in time-variant volumetric heterogeneous data visualization, several factors were taken into account while deciding which application direction to proceed in. We summarize these factors as follows:

- Can we establish a long-term collaboration with domain experts? Experts' location, schedules, and willingness to pursue novel visualizaiton techniques to serve their analysis needs were considered in answering this question.
- Is enough data available? Given that our research focuses on the fusion of multiple sources of

data that serve the analysis of a given phenomenon, the choice of application was restricted by the availability of datasets obtained from different modalities and from simulation.

- Does the visualization framework solve the problem? The proposed framework is of generic nature therefore, in principle, it has the potential to solve a wide range of domain problems. Therefore, this question was the least limiting to our choice, but had to be considered.

1.3.2 Core Phase

The core phase is where the bulk of framework design and reference system implementation took place. We started with a concrete definition of the goals that the system seeks to achieve and the visualization tasks of visual fusion that can help satisfy the goals. We then deduced research hypotheses and formulated questions that need to be answered in order to test these hypotheses. We explain these components in what follows.

The Goals of Visual Fusion

An effective time-variant data visualization aims to help users and domain experts answer a number of questions about: *(i)* whether a data element exists at a specific time, *(ii)* whether two events are correlated, *(iii)* what is the length of the time span from beginning to end of a feature of interest, *(iv)* what is the rate of change, and *(v)* what is the sequence of events. Consequently, dynamic, time-varying data present a big challenge to visualization researchers. We need to enable the user to discriminate between the different data features and to identify and understand changes that those features undergo over time. As a result of this understanding, the user can generate hypotheses on how to solve the problem of data reconstruction then test these hypotheses to verify the correctness of their understanding.

Providing these abilities through a visualization system sets certain requirements on design and implementation. The system seeks to support user's comprehension of the problem, the datasets

available and their shortcomings; and to support hypothesis generation and testing. In order to address these requirements, we identified a set of goals that a visualization system seeks to achieve and derived a number of user tasks needed to satisfy these goals. Figure 1.7 visualizes the mapping from problem solving requirements to visualization goals that the system seeks to achieve. In the following we describe each.

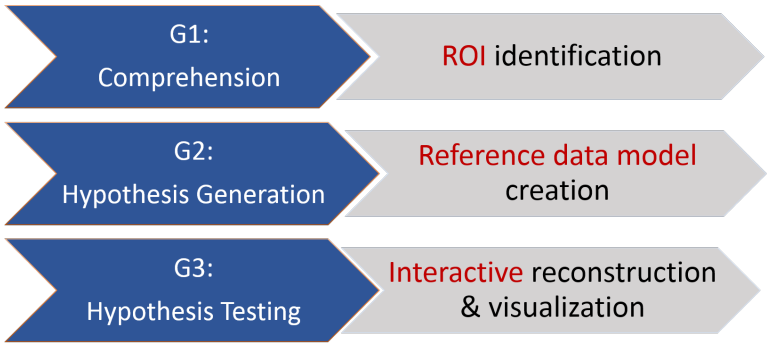


Figure 1.7: Mapping requirements of problem solving to visualization goals. Keywords shown in red map these goals to research questions.

G1: Region of Interest identification

In every domain, there is a set of interesting objects or features that are usually defined by *regions* in the dataset that satisfy certain constraints. An example from meteorologic data can be an area of low pressure defining an oncoming storm. Sets of points in a spatio-temporal neighborhood can be tested for inclusion in the ROI. The test for inclusion depends on the type of feature we are looking to extract. In the following we describe two different types of features that can be of interest in scientific visualization:

- **Isovalued Features.** The simplest definition of a *feature* can be conveyed through highlighting isovalued clusters in a visualization. Coherently colored areas are a natural way to draw user's attention, which made this a standard way in many areas of research to visualize homogeneous features. That is, features consisting of sets of neighboring points above or below a certain threshold value. The extraction of such regions is often implemented using

$3D$ segmentation or region growing algorithms. The seeds around which a region is grown can be automatically generated based on certain criteria of relevance, like local maxima if we are looking for high-valued regions, or they can be selected by the user. The shape and size of the regions can be controlled by multidimensional thresholding, topological parameters that are known about the domain, or a gradient filter that defines the edges. The set of points constituting an object are stored in a data structure for efficient manipulation. Hierarchical structures, such as octrees, are sometimes used to accelerate access to the objects' underlying nodes.

- **Flow Field Features.** Areas of interest in a flow field can reveal important flow behavior (e.g. vortices). In visualizing simulation results, such areas of interest are not readily visible due to being cluttered by a dense representation of the entire flow field. Several techniques exit in the literature to reduce clutter and highlight flow features of interest. These include different seeding strategies that generate flow lines representing only interesting behavior. In a post-seeding stage, the task of clutter reduction and ROI identification is up to the visualization system to enable. We have deduced from our discussions with domain experts that visual enhancement techniques such as opacity optimization [GRT13] can be helpful up to a certain extent. However, such automated techniques and their generated static representations lack the ability to test different ROI selection criteria that are hypothesized by the expert as she builds up her understanding of the data.

We focus our attention on interactive clutter reduction and thresholding techniques to enable real-time hypothesis generation and testing about what constitutes a ROI. The idea is to empower the expert to generate a multitude of visual encodings for the data, and test whether they present an identifiable object that can be linked to the expert's existing cognitive model. This is an iterative process, in which data is visualized, perceived by the user, linked against existing knowledge, and then new visual encodings are generated to test other alternatives. The user tasks involved in this iterative process can be stated as:

- **T1:** Overview, browsing and clutter reduction
- **T2:** Estimation of relevant attribute ranges through interaction
- **T3:** Time-enabled exploration and integration of multiple datasets

G2: Reference model Creation

Reconstructing dynamic behavior from sensor data can be theoretically insoluble just like the problem of obtaining a 3D scene from a 2D image is insoluble. In the latter, we use certain assumptions about, for example, the size of people, the shades, etc. to interpret the scene. These assumptions are inferred from the viewer's prior knowledge of what the world is like in real life. Such prior knowledge is not always complete in the case of scientific phenomena. Therefore, the visualization system must provide data exploration capabilities that help the user build a cognitive model of what happened in the real world, even if merely a hypothesized one, in light of what the data reveals. In order to achieve that, the user must be given the ability to manipulate the data, focus on specific regions and eliminate the clutter caused by uninteresting ones. This interactive process helps speed up hypothesis testing and aids users' decision of what constitutes a "most likely" scenario of what happened in the data. Once this cognitive model is achieved, the system seeks to capture it in the form of a reference dataset that covers the spatio-temporal Regions of Interest (ROIs) at a higher resolution than the existing modalities.

User tasks relevant for this reference model creation process were identified through interviews with domain experts and can be summarized as follows:

- **T4:** Investigation of interactions among different spatio-temporal regions and whether they can together be linked to phenomena of interest
- **T5:** Drilling down on specific ROIs and deriving a description of expected behavior

G3: Data Reconstruction

Once a reference model is identified, it is used for feature tracking in any individual dataset through matching extracted points $X^i_{t_0}$ from the dataset in question to a neighborhood in the reference model N_{t_0}, then extrapolating through time to obtain an updated location $X^i_{t_d}$ where t_d is an arbitrary destination point in time at which we would like to find the location of a feature that was detected at time t_0. This tracking process can be done offline on the CPU for a limited number of regular time steps, creating multiple frames of instantaneous images that fill out the spatial gaps in all of these frames. This can be useful for a high quality animation of the physical phenomena under investigation. In cases where the user is interested in tracking the data at variable and arbitrary points in time, however, it is crucial to perform tracking calculations on the fly as user slides through time. In order to do that, we develop GPU-based techniques and we investigate their potential to speed up neighborhood matching and tracking through acceleration data structures.

This reconstruction process is semi-automated. As the user scrolls a time slider, the reconstructed data is calculated and visualized in real-time. System-enabled user tasks that aim at aiding the testing and evaluation of the hypothesized reference model and reconstruction mechanism can be stated as:

- **T6:** Obtaining details on a specific region of interest, answer questions such as whether objects or features of interest existed at a certain time and place, how long they persisted, and what happened next
- **T7:** Determining whether reconstruction results match hypothesized behavior, and modifying hypotheses accordingly

In an ideal visualization, the initial view of the world must give an overview of the context while highlighting regions of interest to guide user interaction. Rendering adequate representation of time-variant $3D$ datasets at interactive rates, while providing abilities to perform the user tasks described above is the main challenge we address in this work. We combined the identified goals

in a unified visualization framework, which we describe in detail in Part II of this dissertation. We designed and implemented a system to test the framework and enable the user tasks in the context of atmospheric data visualization.

Hypotheses

Based on the identified goals and user tasks that characterize the problem of interactive visualization of multi-sourced time-variant three-dimensional data, we state our research hypotheses as follows:

H1. Exploration tasks enable the construction of a cognitive model representing the expert's comprehension of physical phenomena. This cognitive model can then be embodied by a reference dataset.

H2. Real-time interaction is achievable during the exploration, reconstruction, and enhancement of complex three-dimensional data, by leveraging GPU-based techniques and advanced acceleration data structures.

Research Questions

In order to address the problem and test the stated hypotheses, we have identified the following research questions:

1. **How to adequately communicate the spatio-temporal structure of *regions of interest* via user-selected levels of abstraction?**
   Given the differences in size, scale, and resolution of the different data sources, the decision of what constitutes an adequate level of abstraction for each source becomes a nontrivial one, and is best made by the human expert. In answering this question, the proposed interactive visualization techniques aim to convey information about whether a particular modality covers a spatio-temporal region of interest or not, and answer questions like: how dense is

the coverage within ROIs, and what combinations of data attributes contribute to the identification of such ROIs. Addressing this question is very data-dependent, and answers can be formulated differently for every modality. Nonetheless, we developed techniques to address general user tasks for the ad hoc exploration of data attributes and of the resulting visual encoding. Positive expert feedback was received about the flexibility of interactive visual mappings and their intuitive use during the ROI identification and data understanding phase.

2. **How to effectively incorporate data characteristics and domain knowledge to create a *reference data model*?**
   The goal of this question is to assess the ability of the proposed visualization framework to combine data from measurement sources and simulation results, while enabling the real-time tracking and reconstruction of missing location and attribute information. Ideally, particles of interest (e.g. from satellite detections) are advected through a simulated flow field that is constructed as a reference dataset embodying the user's cognitive model of the phenomenon in question. The aim is to reconstruct data positions at arbitrary target time instances that are selected by the user. This advection and position reconstruction needs to be performed in real-time to support critical decision making and to strengthen the scientists' understanding of the story behind the data. In answering this questions, we evaluated the effectiveness of the proposed tracking methods in terms of their ability to accurately model the evolution of the physical phenomenon in question, while maintaining interactive rates.

In addition to these human- and data-centered research questions, a lower level implementation-related question was kept into consideration in order to serve and maintain system interactivity.

3. **How to leverage the power of modern GPU computing capabilities to speed up computations and achieve *interactive* performance?**
   The challenge of information fusion arises from the lack of structure in many of the available datasets including simulation results. The ability to match existing data pieces from multiple modalities while reconstructing the non-existing ones using spatio-temporal inter-

polation methods, can be computationally very expensive due to massive numbers of neighborhood lookups and interpolation calculations. This question seeks to assess the possibility of achieving real-time performance at interactive rates, through efficient GPU-parallel algorithms and acceleration data structures.

1.3.3 Analysis Phase

The analysis phase of our proposed methods was multi-tiered. Reflections on the data led to a characterization that guided our classification of the literature according to three axial data characteristics in the targeted scientific domains, and a prioritization of the challenges herein. In addition to data characterization, guidelines for real-time data reconstruction and for analyzing the accuracy of the results were useful outcomes of this reflection.

At the technique level, presenting novel interactive techniques to the visualization research community has further highlighted the importance of analyzing the accuracy of the reconstructed information in order to validate the ability of our interactive methods to produce visualizations that are a true reflection of what happened in the real world. Feedback on the proposed methods confirmed the goals and guidelines we had set for our framework, and highlighted the value of real-time interaction in supporting expert's understanding and cognitive model construction.

At the implementation level, reflections on our developed GPU-parallel programs enabled the identification of performance bottlenecks; which guided us to introduce novel and improved acceleration data structures optimized for real-time performance.

1.4 Contribution

Following the methodology described in Section 1.3 to address the above stated research questions and test hypotheses, we propose a framework for the real-time processing and visualization of data relevant to a physical phenomenon, given measurements collected from multiple sources and a simulation dataset. The contributions of this work can be summarized as follows:

1. State-of-the-art literature review of heterogeneous spatio-temporal data visualization in light of three axial characteristics (volumetry, time-variance, and multimodality) and use of this characterization to classify visualization literature.

2. A characterization of the visualization problem from cognitive science, human-computer interaction, data fusion, and computer graphics perspectives; and a discussion of design rationale in light of these different perspectives.

3. A statement of goals for visualization that are informed by cognitive psychology to aid the expert users' cognitive model construction process.

4. An integration of the goals into a generic framework that can guide future problem-driven research that aims at visualizing heterogeneous spatio-temporal data.

5. A translation of the stated goals into concrete task specifications and design requirements for visualization systems.

6. A demonstration of the reference implementation of the framework and related interaction techniques to support the stated goals and tasks.

7. An evaluation of the results in terms of reconstruction accuracy (quantitative) and system usability (qualitative) in light of the proposed interactive techniques.

This Chapter defined the problem of Scientific Visualization in light of three axial data characteristics, and summarized the related challenges. In Part I of this dissertation, we elaborate on

each of these data characteristics and provide a classification of the relevant literature. In addition, having summarized our methodology, the questions we seek to answer, and the contributions of our work in this Chapter, we have paved the way to explain the details of the proposed solution approach, which we elaborate on in Part II. Last but not least, in Part III, we discuss results of testing the usability of the proposed framework and accuracy of its reference implementation.

Part I

Part I: Background and Literature Review

Scientific data characteristics, as described in Chapter 1, have been a major driving force in visualization research. Frameworks and techniques have been developed to address the challenges inherent in each one of these characteristics. Historically speaking, there has been little overlap among the data characteristics that were targeted by the solutions proposed in the literature. Research in Volumetric data visualization, for example, did not fundamentally address time-variance or multimodality issues and vice versa. This trend has changed in more recent years, where a shift of paradigm in Scientific Visualization exhibited a merge of these — once considered separate — research directions.

In this part of the dissertation, we elaborate on the historical background of each one of these research directions, and provide a classification of the literature based on them. We discuss recent trends and highlight areas of overlap.

Chapter 2

Time-Variant Data

"Has there come upon man a time when he was a thing unremembered?"

Quran(76:1)

The analysis of time-variant data is one of the most common problems in several application areas, including science, engineering, and business. Data acquired through sensor networks which monitor natural phenomena, as well as simulation data that imitate time-identified events, have fueled the need for interactive techniques to quickly analyze and understand trends and patterns across time.

Time-varying data can be categorized by the different temporal behaviors that they exhibit [WYM08]:

- *Regular* data involves a certain phenomenon that grows, persists, and declines in several distinct stages. The rate of change at each stage can vary dramatically in space and time. Natural phenomena such as earhtquakes fall into this category.
- *Periodic* data includes recurring patterns. In this case, features of interest that the visualization seeks to emphasize are space-time abnormalities. For example, meteorological data

normally follows some temporal patterns (e.g. daily, monthly, yearly). It is the abnormality in such patterns and fluctuations out of expected ranges that requires further investigation.

- *Turbulent* data is characterized by the ubiquitous presence of spontaneous fluctuations distributed over a wide range of spatial and temporal scales. Examples of this type are usually found in computational fluid dynamics (CFD).

Figure 2.1 shows example depictions of the three types of time-varying data.

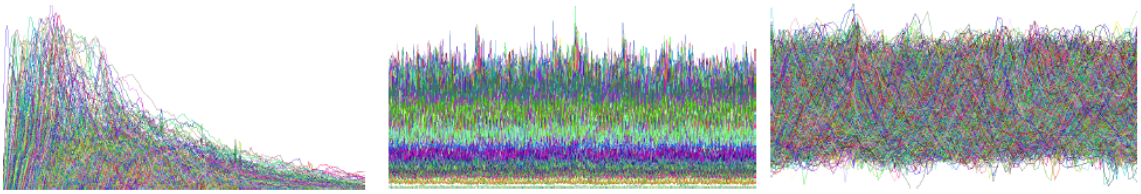


Figure 2.1: Temporal Behaviors of Time-varying data [WYM08].

2.1 Historical Background

For a long time, visualization has proven to be very useful in analyzing time-varying data. This is due to the ability of the human eye to quickly detect patterns and structural changes in images. Early attempts to study time-variant phenomena were pioneered by Étienne-Jules Marey, a French scientist and photographer who developed animated photography (also known as chronophotography) techniques in the 1880's to display and interpret data from cardiology, aviation, and physiological movement [Sel06]. Figure 2.2 depicts an example of a bird captured at different stages of flying. Marey's revolutionary idea was to record several phases of movement on one photographic surface.

A 1912 painting called *"Nude Descending a Staircase"* by Marcel Duchamp, a French artist, also marks one of the early attempts to capture motion and time-variance through still images

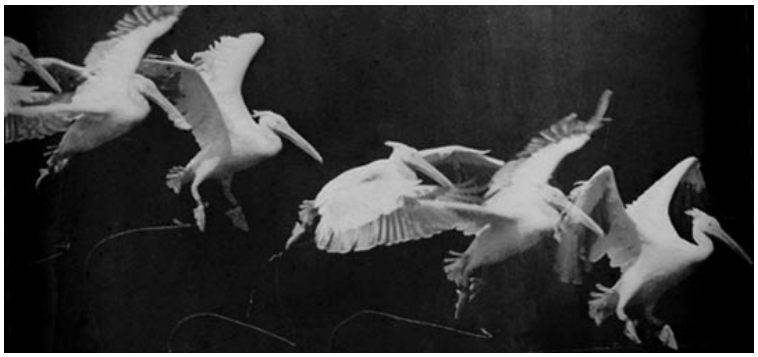


Figure 2.2: Flying Pelican, by Étienne-Jules Marey (1882), capturing several phases of movements in one photo [Sel06].

on one display, as shown in Figure 2.3. His work presented an artistic appreciation of non-photorealistic illustration as a means to depict physical movement. In one interview, he once stated: *"My aim was a static representation of movement, a static composition of indications of various positions taken by a form in movement, with no attempt to give cinema effects through painting."*

Visual and sequential art inspired more modern graphical techniques to represent activities and events by an iconic language that abstracts from photo-realism. Such symbolization simplifies the perception of dynamics through a vocabulary of symbols including arrows, strokes, bubbles, and signs. For example, Motion Lines, or Speedlines [MSS99], indicate moving objects by well-placed strokes. Ghost images mark past, present, and future positions of objects by drawing multiple images of the original objects with fading contours. In addition, partial contours and motion arrows are more examples of visual aids that were adopted for motion depiction on static displays.

Nienhaus et al. [ND05] described smart depictions, which effectively present dynamics and narrate sequential processes. The main advantage is that these depictions can symbolize in a single static image past, ongoing, and future activities and events. Further, non visual information (non-motion related) e.g tension can also be depicted. The $3D$ scene is augmented by dynamics glyphs,

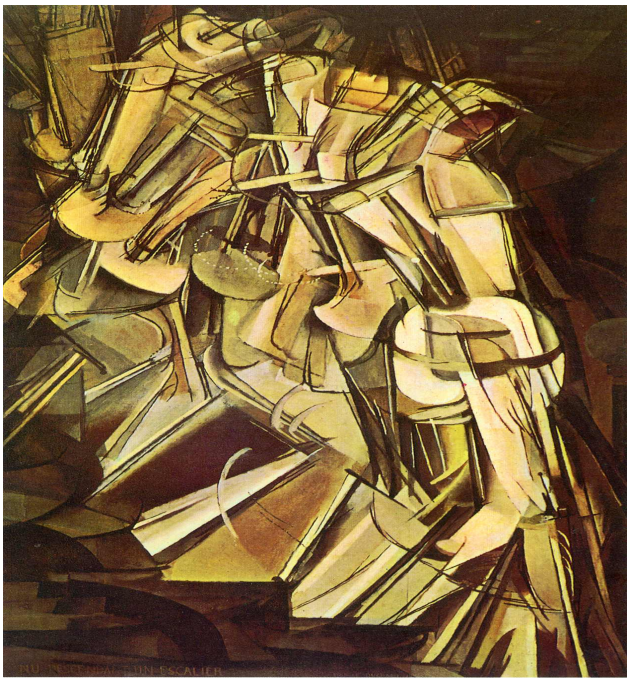


Figure 2.3: Nude descending a staircase by Marcel Duchamp 1912 [WWS03b].

which are additional graphics elements whose appearance can be edited by the visualization designer.

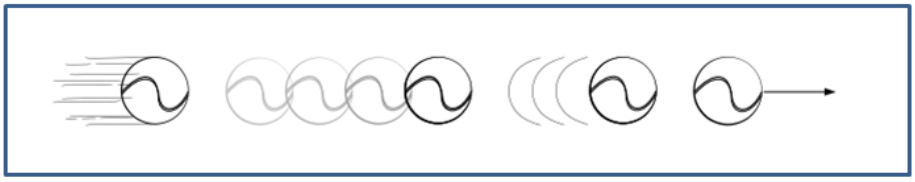


Figure 2.4: Speedlines, ghost images, partial contours, and a motion arrow depicting the movement of a ball in one direction [MSS99].

2.2 Time Depiction in Scientific Visualization

The effectiveness of a time-variant visualization design can be measured by its ability to help a user answer a number of questions about: whether a data element exists at a specific time, whether there is cyclic behavior, the frequency of occurrence of a particular event or data item, the length of the time span from beginning to end of a subset of data, the rate of change, the sequence of events, and whether synchronization exists among data elements.

In general, a visualization of time-dependent data can be either *static* or *dynamic*. Static visual representation does not automatically change over time, and has the advantage that it allows for the conclusion of quantitative statements. While in dynamic representation, the visualization is a function of time. Example techniques for each of these two types, and special attention to their applicability on multivariate data, are surveyed in [MS03].

In a dynamic visualization, an animation is generated from time series data. This can be very useful to understand dynamic behavior, but it relies on the user's memory and cognitive abilities to tie together spatio-temporal relationships, extract relevant features, and track them through time. Whereas in static visualization, techniques aim at reducing the cognitive load by presenting persistent images that represent temporal behavior. In what follows we discuss some of these techniques.

2.2.1 Time Projection

One of the ways to think about time is as an additional dimension in the data. Woodring et al. [WWS03a] provided an alternative to animated visualization, inspired by artistic chronophotographic works like those of Etienne-Jules Marey (Figure 2.2) and Marcel Duchamp (Figure 2.3). Rather than considering space and time as separate components, the data are treated as a four-dimensional data field. Following from the concept of projecting a $3D$ volume onto $2D$ space, they project the $4D$ hyperspace onto a $3D$ hyperplane, allowing for high dimensional direct vol-

ume rendering of the data, as shown in Figure 2.5.

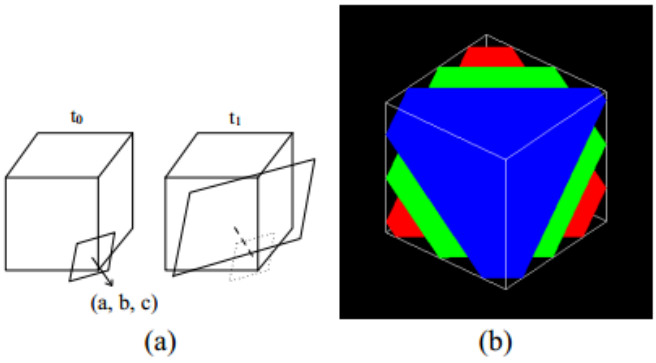


Figure 2.5: Projected $4D$ space onto $3D$ hyperplane [WWS03a].

2.2.2 Time Curves

In addition to sophisticated rendering, scientific visualization of time-variant data is often augmented with information visualization techniques such as curve plots to summarize how the data evolves over time. Fang et al. [FMHC07] addressed the problem of time-varying scientific data visualization from the perspective of medical datasets, assuming stationary anatomy that exhibits dynamic behavior, but is not concerned with more general cases in which shapes can vary non-rigidly in space over time. Their approach replaced the scalar value of each voxel at (x, y, z) location by a time activity curve (TAC). This is done through reconstructing a series of $3D$ images to yield a single time-varying $3D$ image. Similarity measures are evaluated with respect to a template TAC to produce a scalar field, as shown in Figure 2.6

The idea of extracting a template TAC is similar to iso-surfaces in scalar data. It can be determined either by: *(a)* the clinician (as she expects certain temporal behavior from a given tissue), *(b)* finding average of a ROI, *(c)* Principal Component Analysis, or *(d)* nonlinear dimensionality reduction. It is expected that all voxels corresponding to tissues of the same physiology, performing

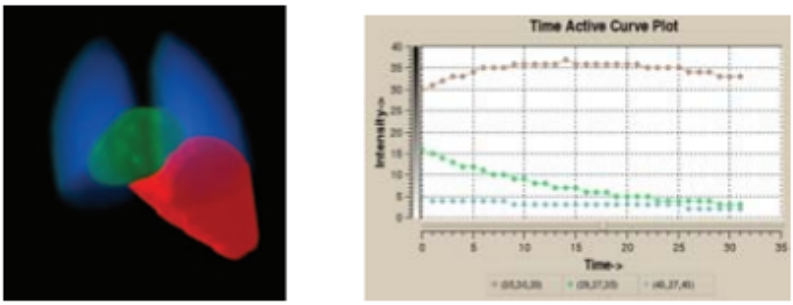


Figure 2.6: Scalar field for time similarity (left) and time activity curves (TAC) for sample voxels (right) [FMHC07].

similar functions, and/or belonging to the same healthy organ will display similar TACs. However temporal behavior can vary across different spatial distributions, which means that close spatial proximity does not imply temporal coherence and vice versa. We note here that Fang et al.'s work does not assume motion. Instead, it deals with intensity changes over time for stationary anatomy tissue.

Wang et al. [WYM08] adopted an information theoretic approach to perform an ***importance-driven*** visual analysis of time-varying data through evaluating importance measures *locally* around a spatial neighborhood (i.e. a data block) in the *joint feature-temporal space*. The feature space is a multidimensional space that consists of data values, local features such as gradient magnitude, and/or domain-specific derivatives or quantities. Entropy and mutual information measures are calculated on a per block level, and then these values are used to determine importance, based on a weighted sum of conditional entropy, for which weights are determined through visualization-specific transfer function and and can change during runtime as the user interacts with the transfer function.

2.2.3 Streamlines and Pathlines

The visualization of vector fields representing flow data is a subset problem of scientific visualization that models dynamics governing the motion of fluid, gas, etc. In addition, features of interest in a time sequence of volumetric data can be mapped across a series of images capturing its evolution. This mapping is achieved through registration of the images, which results in a flow field representing the magnitude and direction of change.

The main objective of flow field visualization has been to convey the correct magnitude and direction in two or three dimensions, but most importantly to convey critical points, or regions of interest (ROIs) in the field. A vector field is as assignment of lines to points in a subset of Euclidean space. Each arrow (vector) has magnitude and direction and can model the speed and direction of a moving fluid through space or the strength and direction of some force, be it magnetic, electric or gravitational force. Figure 2.7 shows an example vector field.

Like volume data, flow data is a type of scientific data that is available through simulation results or collected by sensors in many engineering disciplines. Examples include velocities of wind and ocean currents, results of fluid dynamic simulations, magnetic fields, blood flow, cell migration during embryo development, and components of stress and strain in materials.

The visual representation of flow data has been a central problem in scientific visualization. The main objective is to convey magnitude and direction of flow, while highlighting critical points, or regions of interest (ROIs). In order to fulfill that, four general classes of techniques exist in the literature [PL09]: *(i)* Direct representations [Lar08] place arrow glyphs at every sample point. *(ii)* Texture-based techniques [VW91] [LHD$^+$04] aim to provide a complete, dense representation of the flow field with high spatio-temporal coherency. *(iii)* Geometric, or integration-based, techniques [MLP$^+$10] advect streamlines for steady flow, or pathlines for unsteady flow, using different seeding strategies [MTHG03], for the purpose of occlusion reduction. Ma et al. [MWSJ14] defer

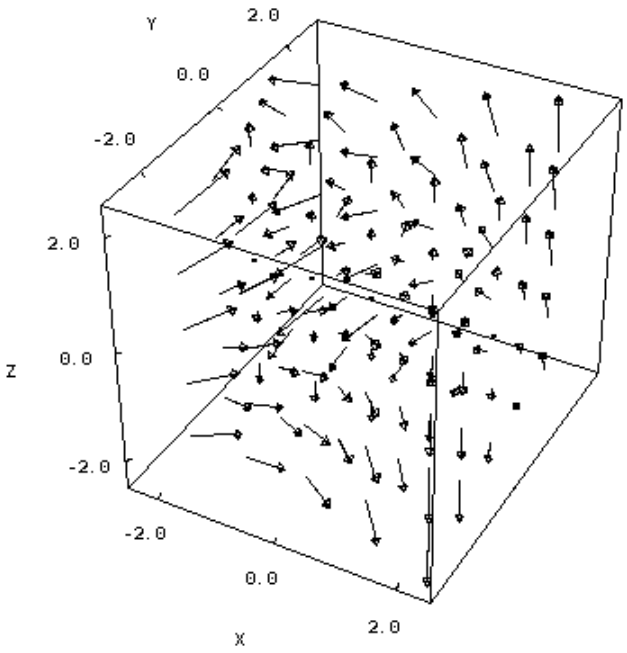


Figure 2.7: Example of a vector field that represents a flow. Arrows are drawn at selected locations where the field is seeded.

the occlusion reduction process until after the streamlines are generated, by creating a graph-based representation. *(iv)* Feature-based [LWS13] methods extract flow features that are deemed interesting by the user in a preprocess. The visualization is then based on the extracted subsets rather than the entire dataset. Particle tracing methods can be used to reconstruct missing flow information for features of interest [COJ15].

In $2D$, texture-based methods were first well perceived by researchers in the field of flow visualization. Wijk et al. [VW91] introduced **spot noise**, a technique using stochastic textures to represent flow data at high density. Cabral et al. [CL93] developed LIC (Line Integral Convolution) which convolves a white noise input texture along streamlines. Their technique results in similar images as those in [VW91]. Early work on two-dimensional vector visualization used streamlines, i.e. integral curves that are everywhere tangent to the vector field. Multi-resolution approaches offer representations ranging from sparse streamlines to texture-like density. The research question

in focus for these approaches was: "Where to *place* seed points to cover the domain densely with streamlines?" The general assumption was that denser streamlines showed greater field strength. Turk et al. [TB96] introduced a stochastic optimization approach in which an initial seeding is assigned on either a regular grid or randomly to pre-compute an initial image. An energy function is computed for this initial image. A streamlet (short streamline) is selected and moved a small amount in a random direction. If the resulting image has a lower energy measure, the change is accepted. The process is repeated until energy reaches a threshold or acceptance of random changes becomes rare. This random optimization approach suffers from severe scalability penalty since with larger datasets, the number of seeding points can be huge and the possibilities for improvement approach infinity.

Jobard et al. [JL97] proposed a single-pass approach that inserts a new seed point at a minimal distance from all already existing streamlines then integrates a new streamline back and forward from the new seed. A multi-resolution approach was later proposed by Jobard et al. in [JL01], in which more lines are inserted on demand for areas of interest where more details about the flow need to be displayed.

Extensions to $3D$ include the work presented by Mattausch et al. [MTHG03] who used the same seeding strategy proposed in [JL01], and represented flow by thick lines illuminated with Phong shading model. They assumed normal vectors that are coplanar with the light vector and the tangent at each sampling point to solve the non-specificity problem of normal vectors for lines. Two transfer functions were designed for the mapping of scalar flow features to optical properties of streamlines on a sample point basis. So for every point, a scalar feature s is evaluated (e.g. velocity, vorticity, pressure, acceleration, curvature). The scalar value is then mapped by either one of the transfer functions to streamline properties (e.g. opacity, color, width, density). The study used fog and halos to provide depth cueing. In addition to the density-based approach they used to guide the seeding process, other 3D techniques used feature-based, and similarity-based measures.

2.3 A Process Model

Contrast to specific visualization techniques, more general frameworks for visualizing time-variant data were introduced in the literature. These theoretical frameworks essentially describe different process models for data flow before, during, and after visual display. In this Section, we explore some of these frameworks, review their components, and guidelines that can inform research in the field.

Samtaney et al. [SSZC94] explicitly specified a task pipeline for the visualization of time-variant data. We summarize the tasks of their proposed pipeline in Figure 2.8.

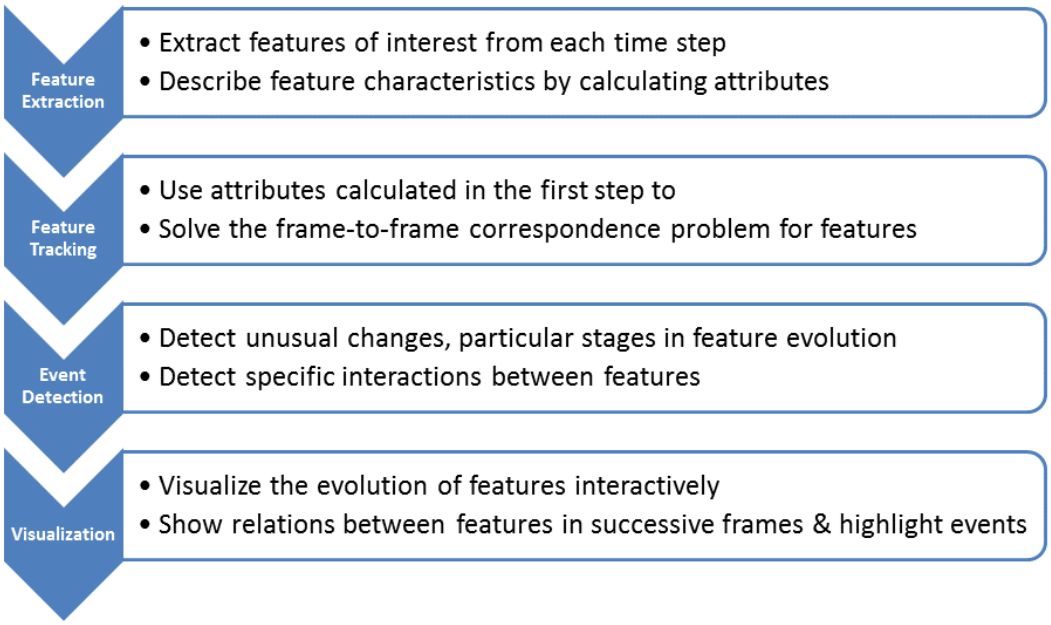


Figure 2.8: Process Summary of tasks involved in visualization of time-varying data as described in [SSZC94].

A related INL (Idaho National Laboratory) report by Rattner et al. [RGJ12] proposed a generalized framework for visualization of time-variant data, as outlined in Figure 2.9. The focus of their work is on the use of illustrative visualization methods to overcome some of the issues and limitations of photo- and physically- realistic techniques. Time-variant data is assumed to be fed to the proposed MarmotViz plugin as individual time steps from ParaView, possibly out of order. Region Of Interest identification, feature matching and tracking and illustrative enhancement techniques are all applied to every time step individually. Their system supports one ROI identification and one feature tracking technique, although it is generalizable to other methods. Multiple illustrative enhancement techniques are supported for unstructured meshes.

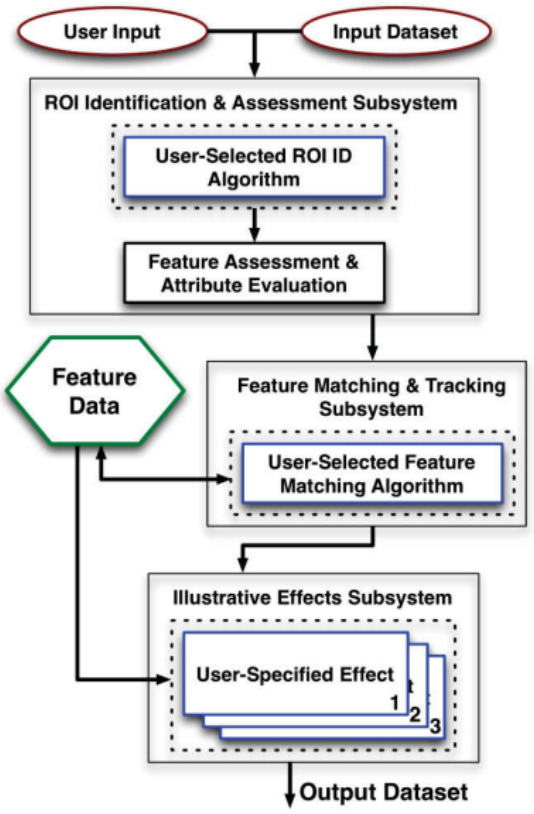


Figure 2.9: Schematic of MarmotViz: a ParaView plug-in for the visualization of time-variant data [RGJ12].

Yu et al. [YLC13] used scale-invariant feature transform for the purpose of the localization of points that contain striking characteristics of an image. Their approach is depicted in Figure 2.10.

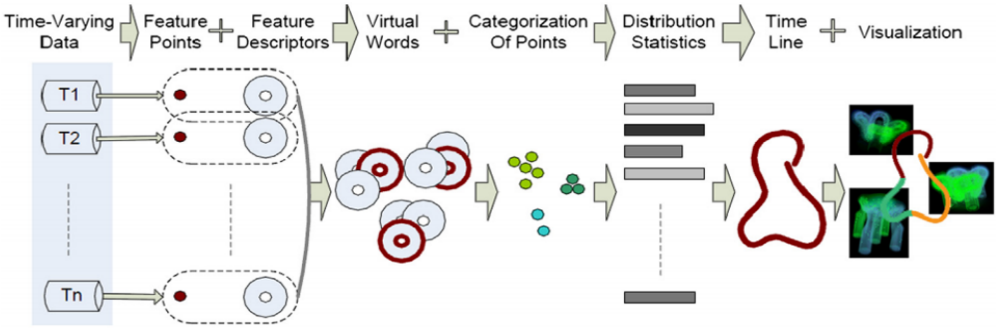


Figure 2.10: Yu et al. [YLC13] Proposed approach for Visualization and Analysis of 3D time-varying simulations with time lines.

We follow a similar process model to the one presented by Samtaney [SSZC94], where the four major components of time-varying data visualization are: *(i) feature extraction*, *(ii) feature tracking*, *(iii) event characterization*, and *(iv) visualization*. The remainder of this Section presents a detailed discussion of each one of the four components and how they have been addressed in the literature.

2.3.1 Feature Extraction

Walsum et al. [VWPSP96a] define a feature as a region in dataset that is of interest for interpretation. They also define, for a given feature, an attribute set as *"A set of characteristic parameter values that represent extracted features"*. The goal of feature extraction then is to obtain an abstract representation of the original data that represent it at a higher level.

In a volumetric regular grid, a feature or a ROI can be a single grid node or a set of nodes from the dataset. A subset of nodes X is created by evaluating some criterion function $H(x)$ over all

grid nodes. A node is included in X if the result of $H(x)$ is true for that node. In which case, we assume that $H(x)$ is a boolean function, possibly constituting of a logical combination of scalar thresholds. However, a criterion function can also include connectivity criteria as in segmentation techniques and connected components.

Figure 2.11 shows a visualization method for isosurface features that was widely used in the 1990s [SSZC94]. The data set is a scalar field with dimensions 256 x 64 x 64 derived from a vorticity vector field resulting from the perturbation of Freon-air interface in five time steps. Despite the low resolution of the dataset, the regions of interest are not clear or easily quantifiable. Further, the visualization only displays five different time steps, each in a separate view. The small number of time steps that can be displayed in multiple views gives very little information about how the feature evolves over time (there are in total 816 time steps in the data set).

As seen in Figure 2.11, thresholded regions classify the dataset into "objects" and "background". Monga et al. [MDR91] classified ROI identification algorithms into two groups: (a) those that identify boundaries around regions of interest, and (b) those that group homogeneous cells. Their study focused on medical imaging , and they argued that it was difficult to find suitable homogeneity conditions for their target application. Instead, they focused on boundary detection methods and developed formulations for first and second order detection methods. It is worth noting here that smoothing is common practice in this type of technique as a preprocessing step before the boundary detection takes place. This is not necessary for simulation data because there are no physical mechanisms involved and therefore, no noise needs to be smoothed.

Reinders et al. [RPS01] developed a feature extraction pipeline based on selective visualization as depicted in Figure 2.12. In this pipeline, data of relevance is first selected by removing clutter and reducing or eliminating information that is irrelevant to features or regions of interest. The extracted data is then integrated through clustering techniques to enable summary statistics and other aggregate information to be calculated, which significantly reduces the size and enables

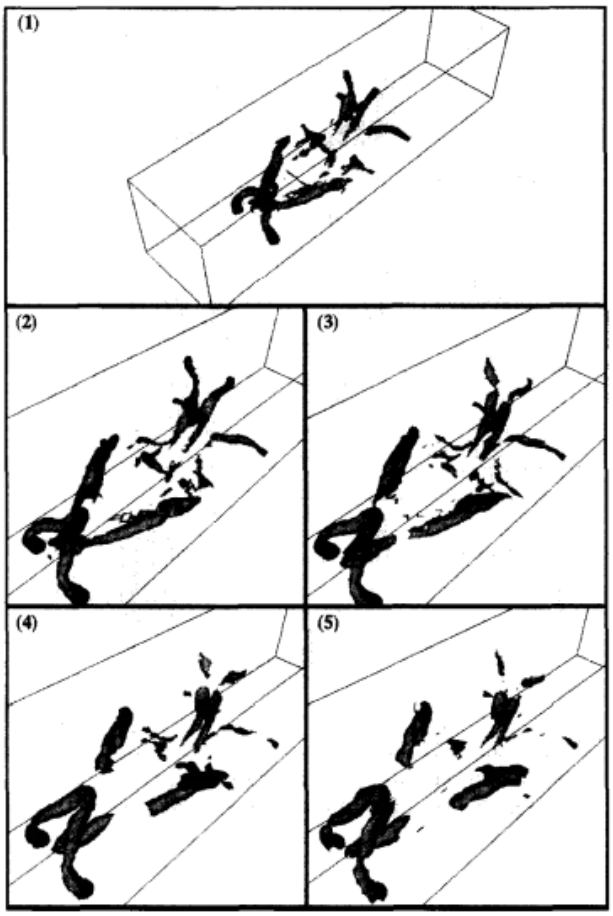


Figure 2.11: Vorticity isosurface for $3D$ shock-interaction, shown in 5 time steps [SSZC94].

interactive visualization.

Rattner et al. [RGJ12] developed a ROI identification and assessment subsystem as part of their framework for time-variant data visualization shown in Figure 2.9. The ROI identification and assessment subsystem is detailed in Figure 2.14.

In their developed ROI identification technique, boundary mesh cells are identified as ones with gradient magnitude above a user-defined threshold. ROIs are then assessed by feature attribute calculation. They suggest the use of attributes like volume, centroid, average velocity, products of inertia, and average angular velocity. ROIs are subdivided into tetrahedra using VTK filter, so attributes can be summed over all tetrahedral volumes composing a particular cell.

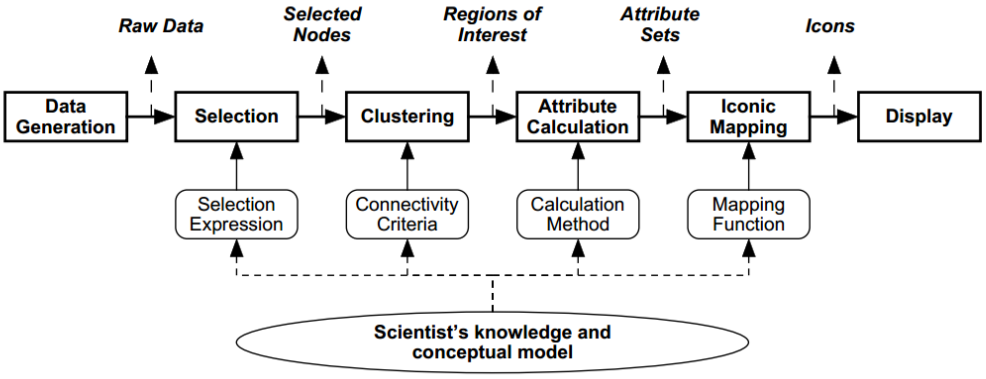


Figure 2.12: Process Summary of tasks involved in feature extraction and visualization in time-varying data as described in [RPS01].

Application-specific feature extraction techniques include techniques that aim at the extraction of critical points [HH91], vortices [BS95], and shock waves [PS93]. More general approaches include ones like region growing [AB94], deformable surfaces [SP97], and selective visualization [VWPSP96b].

2.3.2 Assessing Regions of Interest

Once the individual regions are isolated and stored, attributes describing features for each region can be calculated to quantify those regions. Some common attributes include ([SSZC94]):

1. *Mass*: a weighted sum of all the nodes contained within the extracted region.
2. *Centroid*: the weighted average of all points in the isolated region.
3. *Maximum*: local maxima can be used as representatives of the extracted region
4. *Volume*: the region's volume can be approximated by the number of nodes contained in it
5. *Moment*: the shape and orientation of the region can be characterized by moments of inertia.

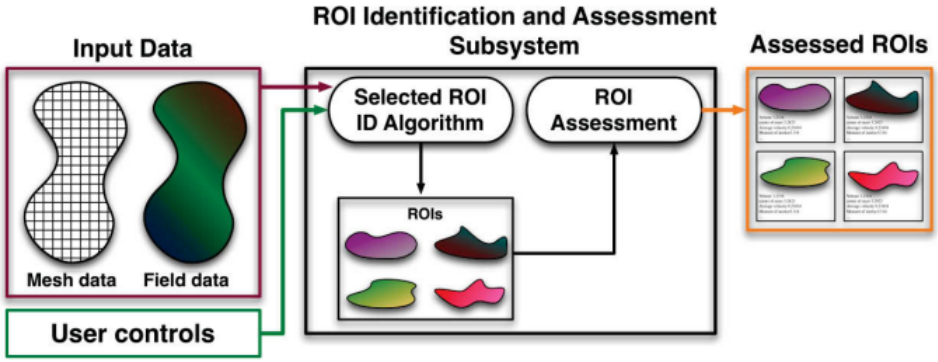


Figure 13. Schematic of the ROI identification and assessment subsystem.

Figure 2.13: ROI identification and assessment subsystem proposed in [RGJ12].

6. *Domain-specific variables*.

7. *Bounding surface*.

2.3.3 Feature Tracking

Feature tracking refers to the process of extracting dynamics, which seeks to identify and describe movement or transport of data. The challenge here is that in the volume data itself, there is no explicit information about motion. The question that feature tracking techniques seek to answer then is: which features are the same object in different time steps? This is a common problem relevant in several fields including Image processing, Computer vision, and Scientific visualization. Each discipline has its own view to the problem but many approaches are analogous. Table 2.1 classifies prior work in the literature in feature matching and tracking. Such techniques can be broadly categorized as:

- Pixel-Based Tracking:

  These techniques seek to correlate among a sequence of frames representing different time steps. Feature correspondence is established through a similarity measure taken at a pixel or

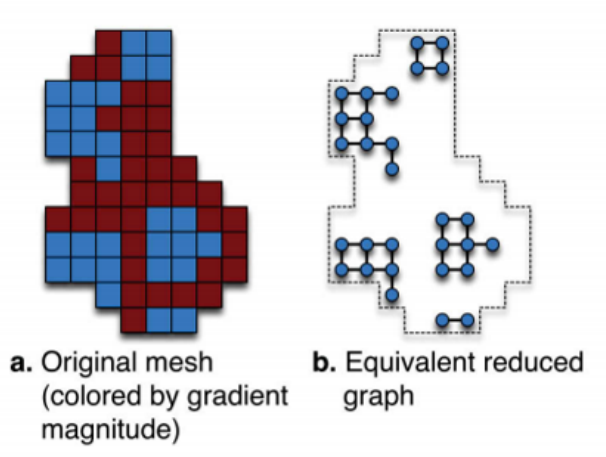


Figure 2.14: ROI identification technique transforming from mesh with (a) high-gradient magnitude cells in red to (b) graph with connections between low-gradient magnitude cell face-neighbors [RGJ12].

Table 2.1: Related work in feature tracking

Pixel-Based Tracking	**Feature-Based Tracking**	
Cross correlation [RdE90]	Region Correspondence	Attribute Correspondence
Optical Flow [Adi]	Match segmented regions based on overlap [SW]	Tracking of markers [SPY94]
Spatio-temporal gradients in image sequence Region Matching e.g. VCM [PPH$^+$08]	Match segmented regions based on minimization of an affine transformation matrix [KS91]	Event detection [SSZC94]

block level. They have the drawback of being computationally expensive especially in 3D fields, and of requiring small time steps for accurate matching.

- Feature-Based Tracking:

  Techniques that seek to reduce the amount of computation required by calculating similarity measures at a more abstract level, that of the extracted features. This has the advantage of reducing the amount of work required by only establishing correspondence among ROIs. These techniques are most useful when the spatial resolution of the data is sparse.

For $2D$ datasets, tracking temporal patterns in a series of images (i.e. motion tracking) has been addressed by research in Computer Vision, providing techniques to track how the boundaries of $2D$ regions evolve over time. Matching an object in one frame to one in another is called the *correspondence problem*. Objects are generally matched in $2D$ using a range of attributes such as pixel values, edges, moments, and geometry. Other techniques for tracking $1D$ and $2D$ feature evolution are reviewed in Sections 4.2 and 4.3 of this dissertation. However, we focus here on $3D$ and higher dimensional feature spaces, for which the problem is complicated due to several reasons:

1. $3D$ features may not be well defined
2. The massive nature of the data poses a set of challenges and limitations
3. Limitations of the human cognitive system which perceives the world as $3D$ makes higher dimensional visualizations ambiguous.

Indeed, the problem of tracking feature evolution in N-D space can be viewed as if time was an additional $(N+1)^{st}$ dimension in the dataset. Adequate tracking can greatly ease the process of invesitgating large datasets, and efficiently automate searching for events of interest. However, automatically tracking features is difficult because the features are continually evolving and interacting.

2.3.4 Event Characterization and Visualization

Event characterization aims at the detection of particular events of interest and deducing quantitative or qualitative descriptions that characterize those events, in order to support expert's understanding og their dynamic behavior. Examples include characterizing the change in size and texture of a tumor, ejected substances from volcanic eruptions in the atmosphere, etc. The visualization system can highlight such events by using illustrative rendering techniques or importance-driven methods like the one presented in [GRT13].

The addition of visual aids and glyphs in the rendering is a common way to direct the experts' attention to detected events. Such detections can be performed automatically or semi-automatically through careful design and implementation of machine learning techniques. However, in many application domains, real-world problems are not easily amenable to such automated detection. Instead, human involvement is crucial to the detection and characterization of events of interest. The ideal system would allow the user to interactively extract and track features, then map the tracked features to a multitude of visual encodings. This effectively empowers the expert to select the best possible visual representation that accurately characterizes the event(s) under study.

2.4 Discussion

Dynamic data present a big challenge to visualization researchers. Not only do we need to allow the user to discriminate between the different features of a volume, but also to identify changes that different features undergo over time. For example, a brain tumor may change in size, shape, or texture. Timely identification and classification of such changes can save lives. Meterologic data collected or simulated over different time periods can reveal important patterns and aid in hourly and daily weather forecasts, and in understanding long-term climate changes. Modern simulation software can generate massive amounts of data. Important practical implications of the results can be explained, given adequate analysis techniques. The ability to identify changes over time or across simulation runs can be of vital importance in these and several other application domains.

A related set of problems arises in areas where there exists a need for simultaneous analysis of multiple correlated datasets or several different regions of interest within a single volume. This is a sophisticated problem with special focus on both interactivity and adequate visual representation.

If a particular region of activity is of interest, scientists attempt to identify and quantify it. However, addressing these questions in dynamic $3D$ datasets is a daunting task given the overwhelming

sizes of the generated datasets. A typical $3D$ unsteady simulation (time dependent) in computational fluid dynamics, for example, may involve hundreds of time steps and thousands of data points.

Traditionally, scientists could not routinely store this information locally or access it interactively for visualization and analysis. Recent hardware advances like larger memory sizes, and increased CPU and GPU speeds, are allowing more and more data to be computed and visualized but at the cost of burdening the cognitive system of the human user with clutter. Displaying too few data elements can miss out important features, while too much data represented leads to clutter which also hinders the user's ability to identify important features. Therefore, a compromise needs to be found in order to design data selection methods that extract and display the "right" amount of information. This can be thought of as an optimization problem. In order to guide this optimization, a notion of *importance* can be defined for the dataset in question. However, in many real-life situations importance information is not available a priori. This is where interactive exploratory analysis becomes crucial to understand the structure of the data and identify the parts that constitute important information.

This chapter covered issues that arise in the visualization of time-variant and/or multiple streams datasets. In Chapter 3, we survey proposed solutions in the literature for 1-dimensional, 2-dimensional, and 3-dimensional interactive visualizations. Every temporal dataset can be described as a function in time of a number of variables. How many such variables need to be considered to identify a feature or a family of features forms the basis of our classification. We categorize research in the field by the dimensionality of the data model's source function, and we survey related work for each. In Part II of this dissertation, we will describe our proposed visualization approach which combines ideas from geometric and feature-based visualization to reconstruct flow information within an extracted ROI and then use the reconstructed model for efficient real-time tracking of the extracted feature.

Chapter 3

Volumetric Data

"Light issues in all directions opposite a body that is illuminated with any light."

Ibn Al-Haytham from the *Book of Optics*, Cairo, Egypt. 11^{th} *Century*

Volume visualization is concerned with the generation of visual representations of volumetric data [PB07]. The process of volume visualization is a complex one. The key challenge is to convey the values within a volume clearly and accurately, while projecting the volume on a $2D$ viewing screen. This projection complicates the view because simply showing each sampled position in the volume with full opacity and clarity is impossible without necessarily obscuring relevant details.

Furthermore, understanding the different media of a volumetric dataset requires computing light interaction with sampled locations in the volume. The locations at which the volume is sampled as well as the rules that govern light interaction have been heavily studied in the literature. We begin this chapter with background on these calculations. Then we describe the state-of-the-art in research in volumetric data visualization, and discuss implications on our solution approach.

3.1 Historical Background

Volume visualization is the process of projecting a multidimensional dataset (usually $3D$) onto a $2D$ image plane, in order to gain understanding of the information contained within the volume data. Historically, volume visualization methods were originally developed for medical imaging visualization, but were also found useful in geoscience, astrophysics, chemistry, microscopy, mechanical engineering, and other areas. The most challenging characteristics that emerged from these application areas are both the size of the data sets involved (order of terabytes) and the multidimensional nature of the data. For example, atmospheric scientists may want to record tens of parameters at each point in the $3D$ space and the visualization should provide a way to display as many of such parameters as possible.

The concepts and consequences of volume sampling can be explained by a fundamental theorem known as the *Nyquist Sampling Theorem* [Nyq24]. In this Section, we describe the details of the theorem, its influence on volume reconstruction and variants proposed in the literature.

The influence of the sampling theorem is basically twofold, and is best described by breaking down the theorem into two fundamental parts:

Nyquist Sampling Theorem (Part 1): "The ideal samples of a continuous function contain all the information in the original function if and only if the continuous function is sampled at a frequency greater than twice the highest frequency in the function." [AS11]. This part addresses sampling errors by restricting our sampling to functions that are band-limited in the frequency domain in a window of width less than the sampling frequency, centered at the origin. For a three-dimensional volume, the sampling frequencies are determined by the spacing of a three-dimensional grid with spacings $1/h_x$, $1/h_y$, and $1/h_z$ in the x, y, and z directions, respectively. The sampling locations along a ray traversing the volume have to be picked at a rate that is twice as high as the respective grid spacing to avoid major artifacts. The theorem assumes ideal sampling

where an infinite number of samples is collected to compute the exact value at each grid point. In practice, however, the finite number of samples that we can store is limited by the resolution of our buffer. Consequently, we cannot produce a truly band-limited function in the frequency domain. This leads to imperfections in the reconstruction process known as "aliasing" artifacts. Often, we can minimize aliasing by prefiltering before a volume is scanned or by controlling the area of the data that the scanner uses to measure a sample.

The second part of the theorem governs our ability to go back from the digital image to a continuous representation without incurring additional error, a process known as **reconstruction** and widely referred to as **interpolation**. The reconstruction of the original continuous function from the samples is based on part 2 of the Nyquist sampling theorem.

Nyquist Sampling Theorem (Part 2): We can reconstruct a 1-D continuous function $f(x)$ from its samples f_i by the formula

$$f(x) = \sum_{i=-\infty}^{\infty} f_i sinc(x - x_i) \tag{3.1}$$

According to the sampling theory, the ideal reconstruction of the exact signal requires the convolution of the sample points with the $sinc$ function in the spatial domain. The $sinc$ function is defined as

$$sinc(x) = \frac{sin(\pi x)}{\pi x} \tag{3.2}$$

The three-dimensional version of this function is simply obtained by tensor-product. This function has infinite extent. Thus, for an exact reconstruction of the original signal at any arbitrary position, all the sampling points must be considered, not only those in a local neighborhood. This is computationally intractable in practice. In practice, the ideal $sinc$ function is approximated by either a box filter, or a tent filter. Since the actual samples are defined only on grid locations, the choice of the $sinc$ function approximation filter determines the interpolation method that reconstructs values between grid locations. For example, the box filter calculates nearest-neighbor

interpolation, resulting in discontinuities between neighboring cells and a rather blocky appearance. A first-order trilinear interpolation, achieved by convolution with a $3D$ tent filter, is a good compromise between computation time and smoothness of the output signal, taking into account the immediate voxel neighborhood for the sample value. A larger neighborhood of the sample is considered in third-order tricubic interpolation, which provides a lowpass filter of the sampled region.

3.2 Optical Models for Volume Rendering

Rendering, in general, is the interaction of light, objects, and the medium in between. The fundamental volume rendering algorithms are of two types: **surface fitting (SF)** and **direct volume rendering (DVR)**. Traditional $3D$ modeling creates objects using surface representations in what is called indirect volume rendering techniques. The word "indirect" emerges from the idea that information is extracted from the $3D$ volume dataset by rendering only a subset $2D$ representation of it. The original data is not directly mapped to its visual representation, but rather an intermediate geometric representation of the data is. These intermediate geometries are then efficiently rendered.

Surface Fitting, also called iso-surfacing, or $3D$ contouring, comprises of a set of techniques to fit planar polygons or surface patches to constant-value contour surfaces, effectively separating a volume dataset into an *inside* and an *outside* subsets. Visual properties are evaluated by light transport only at points on the surface. These methods usually lack the ability to account for light interaction that takes place in the interior of an object. The most popular surface fitting technique is the marching cubes algorithm [LC87], in which a polygonal representation of an isosurface is constructed and rendered. We refer the interested reader to [PB07] for more information on these techniques.

Direct Volume Rendering (DVR) is best fitted for point and particle clouds such as fluids and gases. DVR techniques are originally motivated by scientific visualization as they map $3D$ scalar elements directly to screen space, providing information about the inside of complex volumetric structures rather than the outside surface alone [EHK$^+$04].

Theoretical models for direct volume rendering techniques specify how the display of voxel data is performed by evaluating an optical model, which describes how $3D$ objects interact with light and the surrounding medium, typically by emitting, scattering, reflecting, absorbing, and occluding light particles. These models have their physical foundation in the "transport theory of light" [Kru90a], which is evaluated in its steady state version by the linear Boltzmann equation that describes increase or decrease of the intensity of particles in space [Kru90b].

A survey paper by Nelson Max [Max95] describes the most important optical models for direct volume rendering. Due to the complexity of the original physical models, a number of simplifications can be made by ignoring the influence of changing medium, dropping consideration to different wavelengths, and skipping the scattering component which takes into account light interaction between voxels.

These simplifications bring our model down to the traditional rendering equation introduced by [Kaj86]. If we further discard reflection and refraction, we reach the "density emitter model" [Sab88], a standard physical model for volume rendering, that essentially models every particle in our volume as a tiny light emitting source whose intensity is attenuated while traveling through the dataset. A survey of volume visualization techniques is given in [Elv92].

3.3 The Volume Rendering Pipeline

The volume rendering pipeline [EHK+06] specifies the order in which individual operations are performed to evaluate the discrete volume rendering equation. As Figure 3.1 shows, volume rendering techniques fall in one of two main categories: (1) Indirect volume rendering seeks to identify isosurfaces of the data in a preprocessing step, then renders polygonal representations of them. (2) Direct volume rendering, on the other hand, renders the original data rather than a polygonal representation of it.

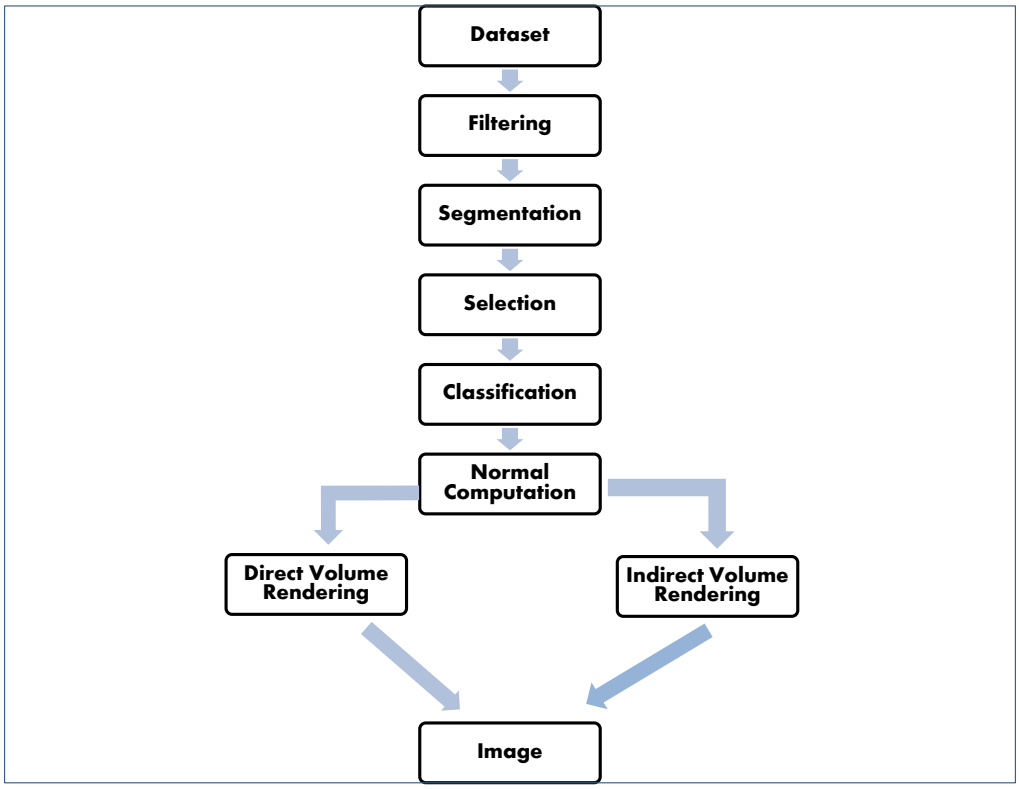


Figure 3.1: Volume Visualization Pipeline

Figure 3.2 shows two variants of the direct volume rendering pipeline The variants shown here differ mainly in the order of execution for the three main modules:

1. Sampling
2. Classification and illumination
3. Compositing

Each one of these modules encapsulates a series of pipeline stages cooperating to achieve its goal. We note here that this generic volume rendering pipeline targets mainly volumes represented as regular three-dimensional grids, in which a transfer function is designed to map voxel values to visual encodings to classify and extract regions of interest. Let us briefly introduce these three generic modules, what they imply in the literature, and how we interpret and generalize them before we delve into the details of specific stages in our proposed framework.

Sampling. For the purpose of evaluating the volume rendering equation in a regular $3D$ volumetric grid, a ray has to traverse the volume dataset at specific locations. The sampling stage of the volume rendering pipeline specifies those locations at which the volume is traversed. Generally speaking, sampling is the process of converting a continuous representation of a function into a discrete approximation of the entity described by that function. In the case of volume datasets, the continuous function is a three-dimensional signal which is sampled at infinitesimally small points called "voxels". An ideal sampler then maps each possible value in the continuous domain to a corresponding value f_{ijk} in the discrete buffer as

$$f_{ijk} = f(x_0 + ih_x, y_0 + jhy, z_0 + kh_z) \tag{3.3}$$

Where h_x, h_y and h_z are the distances between grid points in the x-, y- and z- directions, respectively. Unfortunately, no real sampler is capable of making such a precise measurement.

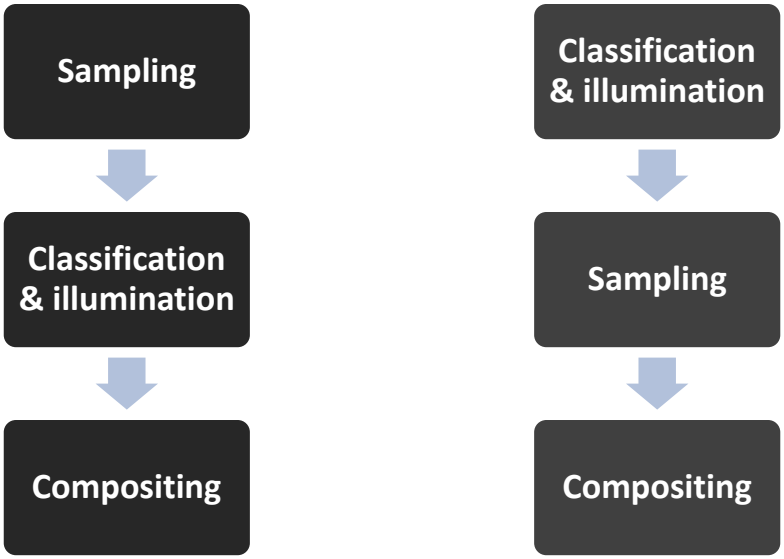


Figure 3.2: Variants of volume rendering pipeline: Post-classified (left) and pre-classified (right).

Classification and Illumination. Once the location and value of a sample point are determined, the contribution of this sample to the volume rendering equation must be computed. This computation is performed in the classification step of the volume rendering pipeline. In this step, the voxel values of a volume dataset are mapped to optical properties by one or more *transfer functions*. A transfer function is a function in a mathematical sense that specifies the mapping from a domain that is constituent of the space specified by Cartesian product of voxel values V_i into the range specified by the product of optical properties O_i. The dimensionality of the transfer function is determined by the number of scalar values attributed to voxels in the domain dataset. If the mapping is from a single scalar value representing voxel intensity to a set of optical properties, the

function is one-dimensional (i.e. with $1D$ domain). More complex transfer function designs take higher domain dimensionality into account. In all cases, the choice of transfer function is a crucial design decision for every volume rendering strategy. An opacity transfer function (OTF) is one that determines the attenuation factor t_i of the rendering equation, based on the sampling value. Opacity is further discussed in Section 3.4.

Compositing. Once sample values are computed, classified, and illuminated, they must be accumulated along the ray according to the physical model, as specified by the rendering equation. This accumulation is done through a numerical approximation of the volume rendering integral and is called compositing, or alpha blending. Order-independent transparency techniques have been proposed in the literature, e.g.[ZWM13b], to enhance the visibility of features of interest in a volumetric dataset, while enhancing users' depth ordering perception.

3.4 Transparency

Visualizing complex datasets usually requires that we render certain parts of the volume semi-transparently, in order to better convey the inner structure of the volume or to provide context. Therefore, voxel transparency as determined by the transfer function, is crucial to the correct interpretation of the visualization. However, transparent or semi-transparent objects present a challenge to volume rendering methods, due to introducing ambiguity to depth order perception.

In the medical field, for pre-surgery planning, physicians need to correctly infer the spatial relations between structures, such as the relation between tumors and their surrounding vessels. Failing to understand this type of spatial relation, possibly due to depth ambiguity, can potentially lead to serious surgical mistakes. The problem is aggravated by the fact that anatomical structures are usually displayed semi-transparently to expose internal structures and additional information. User interaction can alleviate the problem to a certain extent, but it relies heavily on the experi-

ence of the user and her ability to choose the right interactions to reveal important information. Furthermore, if the volume is only available as a static image, with no interaction, the only way to perceive depth is through correct transparency calculation and presentation.

Transparency is a key visual cue in the perception of depth ordering for semi-transparent structures. This is due to the overlap of multiple layers and features in three dimensions which necessitates the semi-transparent rendering of some features to reveal underlying structures of interest. The opacity assigned to each structure needs to vary depending on the importance of the features that the visualization ideally reveals about the structure. This becomes an optimization process that seeks to calculate the optimal opacity for each geometric primitive in our dataset.

Optimal opacity is highly dependent on features of interest. A compromise needs to be made between having maximal conveyed information and having minimal occlusion for features of interest. This process can benefit from importance measures to be calculated, possibly during a preprocessing step, and is commonly referred to as ***importance-driven*** opacity optimization.

A perception-based strategy for opacity optimization was presented by Zheng et al. [ZWM13a] for which a flow diagram is shown in Figure 3.3. Their work is based on the fact that research in visual psychology has revealed that depth order perception of semi-transparent structures is highly dependent on transparency. The X-junction model and its complement the Transmittance Anchoring Model (TAP) suggest that depth perception can be enhanced simply by adjusting the opacity and luminance of the volume. X-Junctions [AA90] can be diagnostic of the nature of transparent interaction, and the depth ordering of layers. Although Zheng's work makes no use of GPU parallelism, and lacks interaction, they perform an optimization for the purpose of statically visualizing three relatively low resolution volumetric datasets. The main contribution of their work is the presentation of a static image of the volume that shows the complex structure of volume features, while maintaining context information. Perceptual transparency is used as a visual cue to convey the correct depth ordering of overlapping objects. This static presentation is able to deliver

the cognitive model of the volume data to the user. Processing times reported in their work lie in the range from 5 to 20 seconds per image (image resolution is 512∗512). In the case of high resolution time-variant dynamic datasets, for example obtained by a medical imaging modality like fMRI, interactive real-time performance becomes an issue.

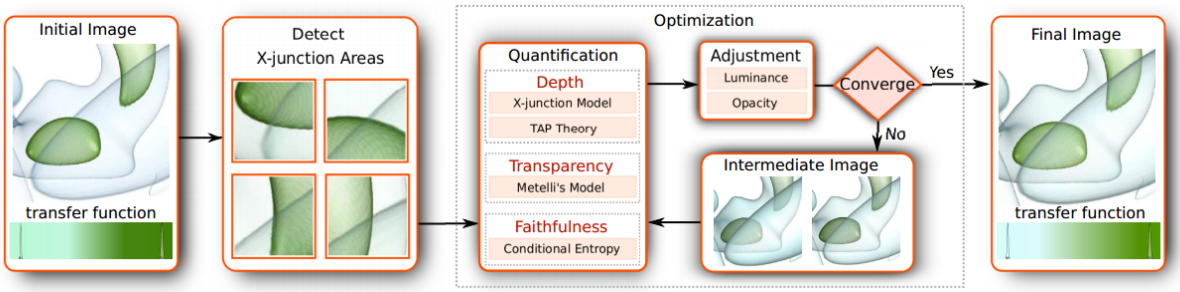


Figure 3.3: Perceptually-based depth ordering enhancement system presented in [ZWM13b].

In addition to transparency-based depth perception, Graphics researchers have long considered providing depth cues to render along with volume data. Interested readers are referred to [War00] and [CBC10] for examples of such depth cues. A general categorization of depth cues classifies them as *pictorial*, *oculomotor*, *binocular*, and *motion-related*. However, some depth cues are not suitable for $3D$ models. Further, different sources of depth cues need to be unified into a single knowledge; but there is no consensus on how they should be combined to convey that knowledge. There is no single accepted cue combination model but the mostly-accepted models of cue interaction are generally variations of the following categories [HR09]:

- *Cue averaging* in which weights are assigned to each cue to determine its reliability. The individual depth cues are then multiplied by their weights and summed to calculate the overall perception.

- *Cue dominance* is a model proposed to consider cue conflicts, in which if two cues provide conflicting information, one of them can be suppressed and perception is based on the other

cue.

- *Cue specialization*
- *Range extension*
- *Probabilistic models*

Cipiloglu et al. [CBC10] describe a system designed to enhance depth perception as one that should consider aspects like the nature of tasks, spatial scene layout, and computational costs of depth cue rendering. They present a framework that uses a fuzzy logic based algorithm for the automatic selection of the appropriate depth enhancement method for a given scene and task.

3.5 Discussion

Volume data is acquired by measurement or numerical simulation of natural phenomena. The most popular examples of volumetric datasets are medical data representing structures and/or functions of the interior of the human body. Different modalities are used for the acquisition of such data like Computed Tomography (CT), magnetic resonance imaging (MRI), electroencephalography (EEG) and magnetoencephalography (MEG). Other examples of volumetric data include computational fluid dynamics (CFD), geological and seismic data, abstract mathematical data such as $3D$ probability distributions of pseudo random variables, as well as synthetically generated volume data in visual arts and computer games where fuzzy objects such as fluids, gases, clouds, fog, and fire need to be visualized.

In contrast to surface rendering methods, direct volume rendering (DVR) techniques generate images directly from three-dimensional scalar datasets without constructing surface-based meta-representations. However, volume data are more difficult to visualize than surfaces. This is due to the fact that volume rendering methods display data by evaluating an optical model at each

individual voxel describing how the volume emits, reflects, scatters, absorbs, and occludes light at every sampled $3D$ location. A transfer function performs a virtual mapping of 3D scalar values to physical quantities that model light interaction at the respective point in $3D$ space. The visual properties of $3D$ structures are inferred from those physical quantities for the actual rendering and display.

The problem of volume rendering is complicated by the type of data present in the different application domains and the fact that data is obtained from multiple sources. In such volumes, multiple overlapping structures introduce clutter and the problem becomes view-dependent. This requires semi-transparent rendering of overlapping structures and the ability to interact with the data in real-time to reveal underlying details. Semi-transparency introduces another challenge. Because of lacking appropriate opacity configuration for each structure, ambiguity is introduced in depth order perception and can lead to wrong assumptions about the spatial correlation among different structures within the view. Special care must be given to the problem of rendering multiple volumetric datasets. An optimal representation is one that assigns higher opacity to regions of interest (ROIs) while rendering the less important structures with relatively higher transparency. However, the correct identification of such ROIs is highly application-dependent and is yet another challenge that we address in this dissertation.

In a general statement, despite difficulties and challenges, it is both worthwhile and rewarding to develop techniques that render volumetric data as truly three-dimensional entities without falling back to 2D subset representations. We generalize the concepts from the volume visualization pipeline (Figure 3.1) to the more generic case of multiple-set $3D$ data visualization without regarding the underlying representation of the data, whether it is structured or unstructured volumetric grids. We argue that the low level concepts of volume rendering can be utilized at a higher level of abstraction to encapsulate the processing stages required to solve the more general problem of $3D$ rendering. Figure 3.4 outlines and categorizes the stages of our proposed multi-volume rendering pipeline, in light of the more generic volume pipeline.

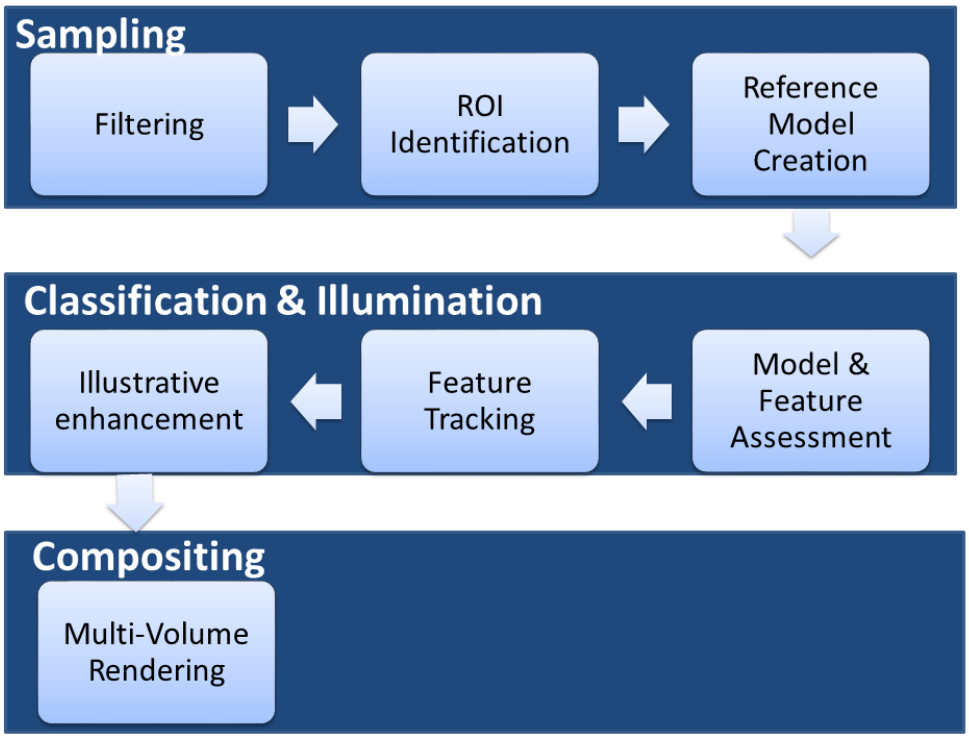


Figure 3.4: Proposed multi-volume rendering Pipeline.

In a multi-volume visualization, several datasets are displayed and the problem of identifying the optimal opacity transfer function to convey a correct spatial organization of the scene, can be regarded as an optimization problem that seeks to minimize the error between the depth information perceived by users and the actual depth order in the dataset. In our proposed approach, instead of performing opacity optimization as a preprocessing step on the CPU, we leave the optimization process to the user through interaction. Opacity calculations are ported to the GPU to achieve real-time rendering at high resolutions, and the decision of what makes an adequate opacity mapping for a given dataset is subject to expert judgment. Furthermore, different visual encoding strategies are required by each volume in order to provide the ability to discriminate among different data sources to the user. In order to achieve that, we introduced a new pipeline stage to serve data classification by selecting a representative data set, which is then used as a guiding model for the visual presentation of other volumes. To accommodate for time variance, a feature matching and

tracking stage is added here before visually encoding the different features to ensure consistency of individual feature appearances across time. Given the large size and data diversity, it is crucial to leverage GPU capabilities at this stage to perform these calculations in real-time.

For the sake of visualizing data obtained from multiple sources, while taking into account the time component, we expand upon the concept of volume sampling. In our proposed pipeline (see Figure 3.4), the sampling phase encapsulates a suite of pipeline stages that aim at efficiently representing the individual volumes of our dataset in both space and time. Different volumes have different sampling requirements and are therefore treated separately at this stage. For example, one volume can be supersampled for higher resolution, while others can be subsampled for size reduction. The decision of which sampling strategy to adopt for a particular volume highly depends on the application domain and the amount of available data in every modality. Therefore, such decision requires a human in the loop to take advantage of users' expertise and their knowledge of the different data acquisition modalities. Preprocessing tasks are implemented to prepare the data for interactive exploration, which is crucial to guide the users' decisions for the sampling phase. In some cases, a volume dataset can be subsampled to support interactive exploration, then once regions of interest (ROIs) have been identified by the user, the data can be supersampled again for effective visual presentation.

Chapter 4

Multidimensional and Multimodal Data

"In the newly evolving multi-dimensional world, we are all one and we are one with our environments."

Elaine Seiler, Multi-Dimensional You: Exploring Energetic Evolution.

Data dimensionality has long been a challenging issue in information and scientific data visualization. Application domains that require the integration of data obtained from multiple sources present the issue of data heterogeneity. Parameters obtained from different measuring modalities (e.g., satellites) at a variety of scales and resolutions need to be combined by the visualization into a unified view that supports understanding of the individual parameters and their interaction. This integration of heterogeneous information is best achieved when a higher level of abstraction is obtained for the data.

This Chapter reviews the data model theory to lay a foundation of data abstraction from existing work in the literature. We begin with a definition and a historical overview of of the data model.

Next, Sections 4.2, 4.3, and 4.4 cover functions of time-varying variables in $1D$, $2D$, and $3D$. In this discussion, function dimensionality is based on the number of independent variables in the source function, apart from the *time* variable. For example, a $1D$ function $f_i(x,t)$, also called a *function graph* [KMG+06], is a function in one independent variable x and time t, while a $2D$ function assigns a set of time-dependent values to each (x_i, y_i) combination of independent variables, and a $3D$ function does the same for each (x_i, y_i, z_i) tuple.

In Section 4.5, we will present a historical overview of the data fusion problem, and discuss related work in the fusion literature.

4.1 Historical Background

In the general sense, a data model consists of a data definition and a manipulation language.

4.1.1 Data Definition

Data resulting from engineering simulations, or collected via real-world sensors share some common characteristics in their data definitions. Each data set contains values for m independent variables and n dependent variables. The subset I of the dataset consists of the independent variables **x**=[x_1,..., x_m] and their values. Every data point $\mathbf{x}_i$ in $I \subset R^m$ represents a specific combination of independent variable values. For each $\mathbf{x}_i$, a corresponding set of values for dependent variables is determined.

4.1.2 Data Manipulation

In the 19th century, mathematicians have established models for the connection between many standard $3D$ geometric concepts and their higher dimensional counterparts [Ban90]. Methods

for describing, transforming, interacting with, and displaying such higher dimensional geometries geared research toward the use of quaternions as $4D$ geometric objects, which laid the foundation for the manipulation and display of time-dependent $3D$ frames.

In a SIGGRAPH 1998 course, Hanson [Han98] explained an *N-simplex* as a "set of $(N + 1)$ points that together specify the simplest non vanishing N-dimensional volume element." This definition focuses on the geometric representation of N-dimensional data. For example, two points $(\vec{x_0}, \vec{x_1})$ define a line in $1D$, this is called a 1-$simplex$. Similarly, three noncolinear points $(\vec{x_0}, \vec{x_1}, \vec{x_2})$ uniquely specify a $2D$ plane, the triangle delineated by these points is a 2-$simplex$. Extending the definition to $3D$, four noncoplanar points $(\vec{x_0}, \vec{x_1}, \vec{x_2}, \vec{x_3})$ form a 3-$simplex$ which is a solid tetrahedron formed by a set of four triangular faces, and so on.

The table shown in Figure 4.1 lists the different types of a simplex used in different space dimensionality. For example, a $1D$ data space can use a 0-$simplex$ (point), in which case there are $N + 1$ possible tessellations of points describing the dataset. We can also use 1-$simplex$ (lines) to describe the dataset in which case the number of possible component structures (tessellations) increases. No higher-dimensional simplex can be used in a $1D$ space. The same logic applies to higher dimensional datasets. Once data is tied to its geometric components, graphics operations like rotation, perspective projection, clipping, shading, and rendering are defined and can be generalized.

The number of dependent variables (Equation 4.1) that are needed in visual analytics tasks determines the dimensionality of the visualization tool.

	Dimension of Space					
Type of Simplex	$N=1$	$N=2$	$N=3$	$N=4$	...	N
Points (0D)	2	3	4	5	...	$\binom{N+1}{1} = N+1$
Edges (1D simplex)	1	3	6	10	...	$\binom{N+1}{2}$
Faces (2D simplex)	0	1	4	10	...	$\binom{N+1}{3}$
Volumes (3D simplex)		0	1	5	...	$\binom{N+1}{4}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$(N-2)$D simplex					...	$\binom{N+1}{N-1}$
$(N-1)$D simplex					...	$\binom{N+1}{N} = N+1$
ND simplex				1	...	$\binom{N+1}{N+1} = 1$

Figure 4.1: Numbers of component structures making up an N-$simplex$ [Han98].

4.2 1-Dimensional Visualization

A dependent variable can be described as one of two types: *(i) regular* variables have a singular value for each $\mathbf{x}_i$, while *(ii) function graph* variables use time as an additional independent variable, providing a set of values for each $\mathbf{x}_i$. Konyha et al. [KMG+06] introduced the concept of *families of function graphs*. That is a collection of curves describing how a single dependent function graph variable changes over time, when different input parameters are presented. They used multiple views along with brushing and linking interaction techniques to answer a number of questions about a real-world road traffic measurements data set, and a diesel injection system simulation dataset. In their approach, no prior assumptions were made about the visualization; and the user is free to display any of the dependent or independent variables in any of the available views.

Visualizing a function graph is intuitively done using a $2D$ plot that shows how this particular dependent variable changes over time, for a specific combination $\mathbf{x}_i$ of independent variable values. A set of values of dependent variables can therefore be defined, at a particular set of independent values $\mathbf{x}_i$ and a fixed time instance t_j, as:

$$\mathbf{d} = [r_1(\mathbf{x}_i), ..., r_{n_r}(\mathbf{x}_i), f_1(\mathbf{x}_i, t_j), ..., f_{n_f}(\mathbf{x}_i, t_j)] \tag{4.1}$$

Where n_r is the number of regular dependent variables, and n_f is the number of function graph dependent variables, which we would like to express in the form of optical characteristics in the visualization. Note that $n = n_r + n_f$ is the total number of dependent variables in the dataset. A family of function graphs $\{f_j(\mathbf{x}_i, t)|\forall \mathbf{x}_i \in I\}$ is defined for each given function graph variable $f_j(\mathbf{x}, t)$ as the set of function graphs for each possible value of $\mathbf{x}$ in the space of independent variables I.

Users in several application areas seek to extract patterns of certain phenomena from simulation or collected data. In $1D$ visualization, only one dependent variable from the set $\mathbf{d}$ in Equation 4.1 is visible at any given view in the visualization tool. This approach works well for visual analytic tasks that do not require the correlation of more than one independent variable across time. That is, for a typical data analysis query, we need to iteratively examine trends, tendencies, and outliers in a small number of dependent variables, for which a separate view is assigned to each individual family of function graphs. Further, how different input combinations map to patterns in only a single output variable's behavior across time needs to be explored through the visualization.

Neuronal spike streams in sensor readings of brain signals are an example of $1D$ datasets that vary over time. The variable x maps different events. Fast solutions are required that are capable of mining a large set of neuronal activity recordings for patterns of interest. Cao et al. [CPP$^+$10] used GPU parallelism to study neuronal function in event streams aquired through multi-electrode arrays (MEAs), with the aims of identifying cascades of firing neurons, determining their characteristic delays, and reconstructing the functional connectivity of neuronal circuits. Their work models

a spike train dataset as a discrete stream of symbols, where each symbol denotes an event type corresponding to a specific neuron or clump of neurons. Occurrence times of these events are encoded in the dataset after a number of pre-processing steps.

4.3 2-Dimensional Visualization

Consideration of data with respect to pairs of independent dimensions is needed in application areas that are similar to the ones discussed in Section 4.2. It is sometimes the case that experts need to identify trends, tendencies, and outliers in the interaction of two independent variables across time. In this case the data model becomes a two-level structure as described in [MGKH09]. At the top level, the dataset constitutes of a collection of tuples (data points):

$$D = \{\mathbf{x}^1, ..., \mathbf{x}^i, ..., \mathbf{x}^m\} \tag{4.2}$$

where m is the size of the dataset. We might also say that m is the number of data points or *rows*, if we are to express each tuple as a row in a table. Each data point is a d-tuple $\mathbf{x}^i = (x^i_1, ..., x^i_j, ..., x^i_d)$, where d is the dimensionality of the dataset, i.e. the number of attributes. An attibute x^i_j can be either an independent or dependent variable. It can be categorical, numerical, or a data series. At the sub-level, for each data series attribute x^i_j of a tuple $\mathbf{x}^i$, we have a set of sub-tuples with its own cardinality and dimension, defined as:

$$D_{ij} = \{\mathbf{y}^1, ..., \mathbf{y}^k, ..., \mathbf{y}^{n_{ij}}\} \tag{4.3}$$

where n_{ij} is the number of sub-tuples, or sub-attributes, whose dimensionality is d_{ij}. A sub-tuple can be defined as $\mathbf{y}^k = (y^k_1, ..., y^k_{d_{ij}})$. The dimensionality of the lower level data series need not overlap with those of the top level dataset tuples. Considering the cases where $d_{ij} \leq 3$, data series can be a sequence of numbers ($1D$) like in the cases discussed in Section 4.2, where time or sequence position is used as an independent variable (function domain) on the horizontal axis and the measured or simulated sequence is the dependent variable on the vertical axis. In case of sequences of pairs of numbers (y_1, y_2), families of function graphs (curves) can be displayed

where one number is used as independent and the other as dependent variable; whereas in case of sequences of three-tuples (y_1, y_2, y_3), one of the data series dimensions can be selected as the dependent variable and the other two as independent variables, which can be visualized as a surface.

To illustrate, we present the meteorological dataset example from Matković et al. [MGKH09]. A multi-run climate simulation is computed while varying external parameters, to investigate scenarios around several melt water outbreak events of pro-glacial lake Agassiz, located in the center of North America. Three independent variables are defined ($I \in R^3$): *horizontal diffusivity*, *vertical diffusivity*, and *time*. The diffusivity parameters are varied across a number of simulation runs (10 variations each), resulting in 10 x 10 = 100 runs. Each run spans 500 years of annual data.

Independent variables in the data model are 35 aggregated values from simulation results, including CO_2 concentration, global surface air temperature, surface air temperatures for both hemispheres, etc. For each simulation run, these values are available for 500 time steps. If data corresponding to these 500 time steps are grouped into one row of time-series per independent variable combination, we will have 100 rows of data in the table. Therefore, $\mathbf{D} = \{\mathbf{x}^1, ..., \mathbf{x}^{100}\}$, where $\mathbf{x}^i = (x^i_1, ..., x^i_{35})$. Each x^i_j is made up of a time-series of measured values at each given time step (i.e. 500 values each). $D_{ij} = \{\mathbf{y}^1, ..., \mathbf{y}^{500}\}$, where each $\mathbf{y}^k$ is a scalar value. Despite the fact that the stored sub-tuples of this dataset are one-dimensional scalar values per time step, some of the visual analytic tasks involved call for an understanding of how this value varies over time with the correlation of the two independent variables ($diff_v$, $diff_h$).

We argue here that the dimensionality of the visualization depends not only on that of the dependent data sub-tuples but rather on the number of independent variables that are needed to answer queries about the data. The number of variables, in turn, depends on a number of standard analysis steps, each requiring data presentation at increasing level of details. Three levels are typically used. Consider the example dataset described above. Suppose the analyst would like to look at all

data surfaces representing one climate descriptor across different simulation runs. This is a *family of data surfaces*. Each surface describes how the climate descriptor changes when varying the two diffusivity parameters.

At the topmost level, the analyst seeks to observe overall trends and high-level correlations among the data surfaces. Scalar aggregates are used to represent each surface. Some standard aggregates include the maximum, minimum, median, mean, and span of a dependent variable. Visualization techniques like parallel coordinates can then be used to display those aggregates (one axis each) for a group of surfaces corresponding to one or more climate descriptor(s). At the second level, the concept of *aggregated profiles* is used to introduce cross-family relations or to explore how each dependent variable behaves with respect to each of the independent variables. To do so, we select one of the two independent parameters and create a cutting plane parallel to its axis. The intersection of this cutting plane with the surface creates a curve. Where the plane cuts the surface is determined based on our selection of standard profile aggregate. Again, it may be the maximum, minimum, mean, median, or all curves profile. In the case of a maximum profile, a surface $f(x_1, x_2)$ is replaced by a curve $f(x_1) = max_{x_2} f(x_1, x_2)$.

4.4 3-Dimensional Visualization

As noted in Chapter 3, rendering is the interaction of light, objects, and the medium in between. The fundamental volume rendering algorithms are of two types: **surface fitting (SF)** and **direct volume rendering (DVR)**. Traditional $3D$ modeling creates objects using surface representations in what is called indirect volume rendering techniques. The original data is not directly mapped to its visual representation, but rather an intermediate geometric representation of the data is. These intermediate geometries are then efficiently rendered. Surface Fitting, also called iso-surfacing, and $3D$ contouring, comprises of a set of techniques to fit planar polygons or surface patches to constant-value contour surfaces, effectively separating a volume dataset into an *inside* and an *outside* subsets. Visual properties are evaluated by light transport only at points on the surface.

These methods usually lack the ability to account for light interaction that takes place in the interior of an object. The most popular surface fitting technique is the marching cubes algorithm [LC87], in which a polygonal representation of an isosurface is constructed and rendered. We refer the interested reader to [PB07] for more information on these techniques.

Direct Volume Rendering (DVR) is best fitted for point and particle clouds such as fluids and gases. A popular approach to volume rendering in this case is ray casting [Lev88]. DVR techniques are originally motivated by scientific visualization as they map $3D$ scalar elements directly to screen space, providing information about the inside of complex volumetric structures rather than the outside surface alone [EHK$^+$04]. In this dissertation, we refer to a three-dimensional dataset as a "volume" regardless of whether it is a structured array of voxels or an unstructured set of points in $3D$ space. This generality of terminology is motivated by the fact that we focus on the integration of heterogeneous datasets, therefore, we make no specific assumptions on their format or $3D$ structure.

Data from most medical imaging modalities are either 2D or 3D, with voxel coordinates *(x, y)* or *(x, y, z)* respectively. In static data, these voxels are assigned scalar values, while in dynamic time-variant data, each voxel is associated with a series of sensor readings, or simulation measures. The time-series of each voxel can be represented by a temporal function, typically referred to as a Time-Activity Curve (TAC). Temporal behavior can be attributed to interaction of tissue with a radioactive tracer (e.g. in SPECT (Single Photon Emission Computed Tomography) and PET (Positron Emission Tomography), or to the functional behavior of voxels in the dataset (e.g. fMRI).

Problems that require unsteady modeling of 3D geometries or consider unsteady 3D physics, e.g. in functional medical imaging or computational fluid dynamics, necessitate considerations of multiple physical and temporal scales.

4.5 Data Fusion

The process of merging information from heterogeneous sources with differing conceptual, contextual and typographical representations is called information integration. Techniques are used in data mining and consolidation of data from unstructured or semi-structured resources. A related term, "information fusion" involves the combination of information into a new set with the aim of reducing uncertainty. Over the years, the scope of applications that use data and information fusion techniques has increased tremendously from strictly military-related to a broad variety of domains, especially in the commercial and industrial sectors. This Section covers related research from the data fusion literature and summarizes concepts that are relevant to our work. Interested readers are referred to [FN13] for a more detailed overview of the field.

Data fusion is the process of integrating multiple data and knowledge sources representing the same real-world object into a consistent, accurate, and useful representation.

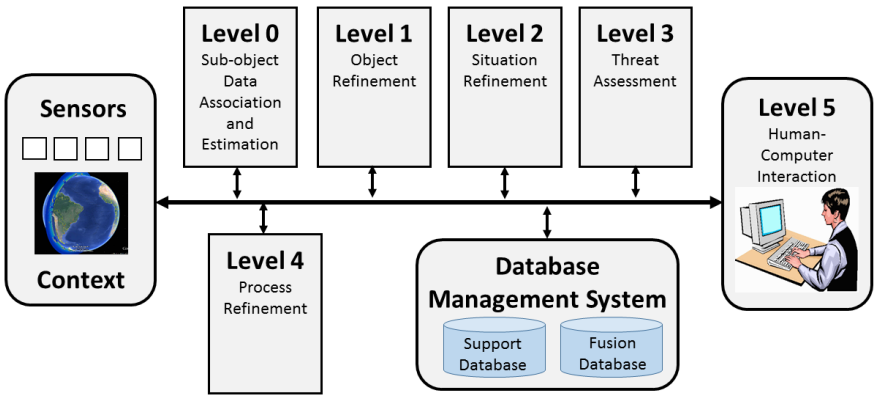


Figure 4.2: JDL Data Fusion Model with Level 5 [HM04].

Data fusion processes are often categorized as low, intermediate or high, according to the processing stage at which the fusion process takes place [Kle04]. Low level data fusion methods combine several sources of raw data to produce new data. This combination usually happens at the sensor or data collection level. Fusion at intermediate to high level involves some form of data collection and processing prior to when fusion takes place. The higher the level, the more human involvement becomes crucial to the fusion process. In all cases, fused data is expected to be more informative and synthetic than the original inputs. After years of research mainly focused at low level fusion, the current trend is shifting towards more high level techniques (Figure 1.2) which poses theoretical and practical challenges that are yet to be addressed.

The trend toward high level fusion began in the mid-1980s, when the Joint Directors of Laboratories formed the Data Fusion Subpanel (which later became known as the Data Fusion Group). With the advent of the World Wide Web, data fusion became more and more relevant and included data, sensor, and information fusion. The JDL introduced a model of data fusion that divided the various processes, as was shown in Figure 4.2.

In the year 2000, Hall et al. [HHT00] proposed the inclusion of the Human Computer Interface as a fifth level of the Joint Directors of Laboratory (JDL) Data Fusion Process Model, shown in Figure 4.2. They argued that advances in HCI, visualization tools, cognitive aids, and collaboration tools should be explicitly considered for their effect on the traditional levels of fusion processing. However, since that time, little research has been done to determine the effect of human-in-the-loop on the process of multi-sourced data integration, though the need for such research has been highlighted by the recent shift in IEEE Scientific Visualization Contest towards datasets constituting of multiple modalities and the merge between simulation and measurement data (see Table 1.1).

4.6 High Level Fusion

High-level data fusion is the study of relationships among objects and events of interest within a dynamic environment. Objects in a dynamic environment are something physical that can be perceived, while events are observed physical reality represented in space and time involving objects. In such dynamic environment, the relationships among objects and events change due to both natural/internal events and external events by players (also called actions). High-level fusion imposes challenging demands for several complicated tasks such as large dataset manipulation, dealing with time-aggregate data, and the capability of incorporating depth information.

Generally speaking, the data fusion process has been commonly characterized as a hierarchy with three general levels of abstraction:

- Data level
- Feature level
- Decision level

Dasarathy [Das94] [Das97] pointed out that fusion may occur within or across these levels .

The Visual Data Fusion (VDF) Model was proposed in the 1990's by Karakowski [Kar98] [BRW07] as an extension to the JDL data fusion model. Figure 5.2 depicts that model.

In this model, the human is a central participant in the information fusion process, and the whole process becomes a creative problem-solving task. Visualization is primarily used to help the user gain fuller perception and decide on possible approaches towards solving the problem. The basic VDF model is used as a building block for visual situation awareness systems. It is the closest of alternative models to our developed framework (see [FN13] for a more complete list of alternatives).

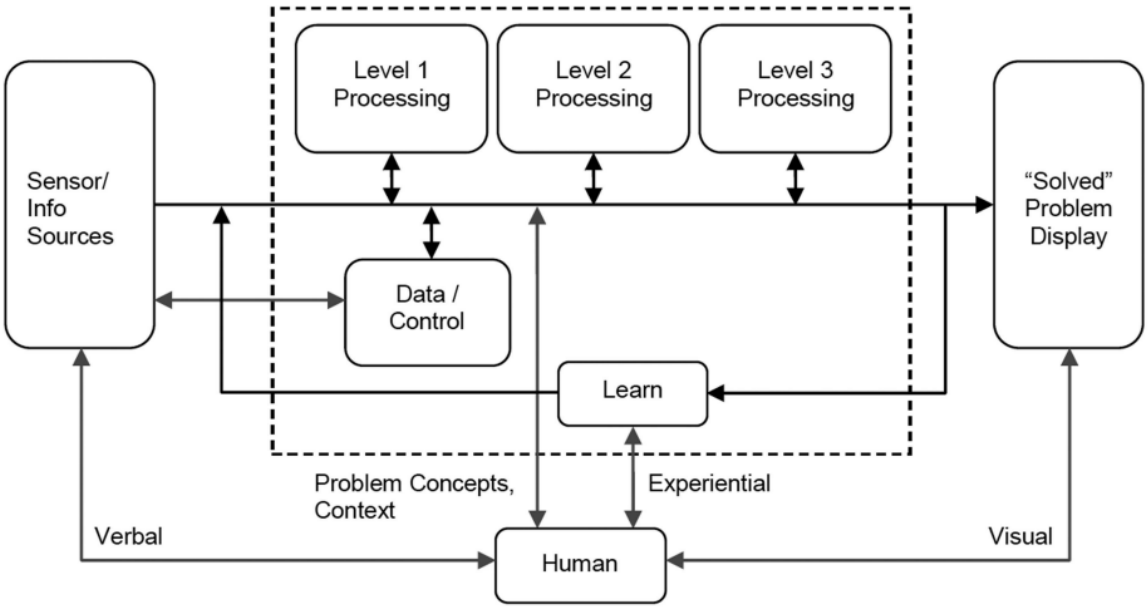


Figure 4.3: Visual Data Fusion Model as proposed in [Kar98].

Different representations of multi-volume data acquired over time, or across different volume data sets, may use different transfer functions for rendering each frame or require viewing the data from different viewpoints simultaneously. Examples of such situations arise when two different data acquisition techniques (e.g. CT and MRI), or data from different time intervals need to be compared. Relying on human eye alone to identify shape differences among distinct static images can be misleading. Volume rendering techniques that focus on the interactive analysis of a single volume dataset have become an established field of research. However, the simultaneous analysis of multiple correlated datasets or several different regions of interest within a single complex volume is a more sophisticated problem. One that not only focuses on interactivity but rather mainly on adequate visual representation.

This calls for research in advanced visualization and rendering techniques, in which a set of challenges is present due to issues like data sparsity, model reconstruction, interactivity, memory management, depth perception, view-dependent rendering, and cognitive models. This PhD re-

search aims to address some of these issues and develop a framework for the interactive enhanced rendering of dynamic three-dimensional multi-source datasets.

Nested or intersecting surfaces are proven techniques for this purpose in 3D static objects. However, the resulting overlap presents a number of challenges due to the semi-transparent nature of some of the complex volume features. In this case, depth ordering perception requires special attention in order to convey the correct feature structure while maintaining context information.

Busking et al. [BBF+11] extend the use of nesting and intersection to dynamic scenarios, in which surfaces can be manipulated or even deformed interactively. Through the use of layered rendering, the authors could enhance the intersecting surfaces visualization.

Further, multiple view displays are also an active area of research in comparative visualization. However, few studies exist in the literature to explore their effectiveness in visualizing volumetric data. Lewis et al. [LHM06] compared different layouts for multiple views in terms of the time required and user preference while completing a number of exploration tasks on both time-invariant and time-variant datasets.

4.7 Discussion

This study aims to develop a framework for the effective visualization of multi-sourced time-variant datasets. We seek to integrate the powers of computer graphics and interactive visualization techniques to effectively display and track features of interest available in multi-modal data sets acquired over time.

Our proposed framework does not assume structure either in space or time. Instead, we tackle the more generic case in which a stream of time-stamped data points are available from simulation

and/or a measuring modality. This level of data abstraction makes visual exploration of heterogeneous sets possible within a unified view. In this view, independent variables can be time, $3D$ position, data modality, or a set of data attributes. Whereas dependent variables can be either measured parameters, or visual encodings of independent variables sub-tuples. Contrast to Rattner's framework which focuses on static illustration, our proposed framework advocates *interactivity* as the main focus at every stage, including the mapping between independent variables and visual encodings.

Isenberg et al. [IEGC08] promoted interactivity in two-dimensional vector field visualization to allow users to probe the data locally, place multiple glyphs to show larger scale properties, and to place glyph sources which aid the process of exploring directional properties of the data. Furthermore, in the literature, a huge body of research advocates interactivity as of having a strong impact on the users' learning process when exploring large datasets; one that bridges the knowledge gap preventing them from accomplishing the tasks. Therefore, we prioritize the matter of user interaction in our overall system design; For this purpose, we focus on techniques that leverage modern GPU capabilities at every stage of our proposed framework.

Our approach builds on motivation for the Visual Data Fusion Model and targets similar goals of keeping the human-in-the-loop at different levels of operation. We present a solution that focuses more on the Graphics side including GPU-based techniques to support interaction at most levels of processing, while enhancing the final visual display of the solved problem.

Part II

Part II: Problem Definition and Proposed Approach

In Part II, we review the definition and characterization of the problem of visualizing multimodal time-variant three-dimensional data from different theoretical perspectives, which contribute to our design rationale. Design requirements are derived from this rationale and are mapped to the components of our proposed framework. Chapter 5 describes this problem characterization and overviews the proposed framework. Chapters 6, 7, and 8 then elaborate on the components of the framework. Specifically, our work focuses on the Exploration subsystem (Chapter 6) and the Reconstruction subsystem (Chapter 7). Whereas the proposed illustration methods are geared toward providing a high degree of flexibility to the user to select from a wide variety of visual encodings, in order to facilitate the cognitive model construction process. Illustration issues are, therefore, covered in the text throughout the different chapters and are subsequently summarized in Chapter 8.

Chapter 5

Rationale and Framework Overview

"Lady Farncombe,
The sponsor of my expedition made only one condition– that I find artists to help compile a pictorial journal recording everything we see."

Egyptology. *Emily Sands. Cairo, Egypt. 1926.*

Applications with complex dynamic datasets need careful management of computational resources, and human-computer interaction (HCI) paradigms. Our work on visualizing multimodal time-variant datasets classifies in the cognitive HCI paradigm described by Harrison et al. [HTS07], which stems from cognitive science. In it, the emphasis is not only on the computational aspect of interaction but also, simultaneously, on the human aspect. Inspired by this concept and the cyclic process we described in Chapter 1, we propose a visualization framework based on the metaphor that *"mind and computer are symmetric coupled information processors"*. The proposed interactive visualizations seek to provide an efficient mapping between the processes in both. Consequently, cognitive psychology — as a *basic* science that seeks to model how humans process input,

how they represent information internally, etc. — plays a fundamental role in informing the design of our interactive visualization framework.

Cognitive theories differ in style and content. However, they share several concepts which outline key activities that contribute to understanding and solving problems. Such activities can guide the design of problem-driven visualization and interaction techniques, and play a key role in validating the resulting system.

In designing a framework and implementing a system for the interactive visualization of multimodal scientific data, we start by characterizing the problem that the framework seeks to solve, from a cognitive point of view. In doing so, we follow the proposition made by Polya [PKR+57] which established that a person would go through four phases when solving a problem:

(a) Understanding (comprehending) the problem;

(b) Generating one or more hypotheses;

(c) Testing hypotheses; and

(d) Checking results.

This part of the dissertation presents a mapping between these problem solving steps and our visualization approach. We begin with a discussion of the rationale for our framework design from a cognitive psychology perspective (Section 5.1.1), a data fusion perspective (Section 5.1.2), a Human-Computer Interaction perspective (Section 5.1.3), and a Computer Graphics perspective (Section 5.1.4). We derive design principles from all of these rationale perspectives and propose a visualization framework that combines them to serve visualization goals that were described in Section 1.3.2. Next, we present an overview of the proposed framework in Section 5.2. Chapter 6 will then set the focus on the first two phases of Polya's problem solving process. Namely, we will describe how our proposed exploratory visual analysis system is designed to support interactivity

and aid the comprehension and hypothesis generation phases. Later, in Chapter 7, we will describe new techniques for the interactive data reconstruction, for the sake of hypothesis testing and results checking.

5.1 Design Rationale

Visualization research can be broadly categorized into **technique-driven** (see the feedback loop from the *"transitional"* space into itself in Figure 1.1), or **problem-driven** (see the arrow from *"applied"* to *"transitional"* space in Figure 1.1). In technique-driven visualization, the goal is to develop new and improved visualization techniques without necessarily addressing specific problems faced by real domain experts. General-purpose techniques for clutter reduction, depth perception enhancement, and illumination are examples of technique-driven research; in which the main goal is to optimize performance to improve support for user interaction and to provide better visual presentations. On the other end of the spectrum, problem-driven visualization research aims primarily at solving real world problems with real data, while addressing the challenges faced in the target domain. Therefore, collaboration with domain experts is crucial for the success of the design and implementation of problem-driven visualization systems.

In some cases, a visualization tool can serve a specific purpose in the problem domain by improving upon existing visualization techniques. The StreamProbe technique, which we present in Section 6.2.2, is an example of this hybrid category. Namely, the technique was designed to serve the exploration purposes of domain experts in the atmospheric science domain, but it also presents a novel generic clutter reduction tool for dynamic flow fields, that builds upon previous interactive exploration methods. The rest of our proposed framework falls in the problem-driven research spectrum; in which domain experts play a fundamental role at different stages of the design, implementation, and evaluation of the system.

This problem-driven approach, guided by the cognitive-based paradigm of HCI research, drives the mapping between information that exists in the human brain and that which is stored in computer memory. For this purpose, visualization acts as a channel for information transport that is optimized to maximize communication between the two.

In this section, we derive design rationale from different perspectives, and identify guidelines and implications to our proposed system in each. We then combine these design principles into a unified framework that we discuss in Section 5.2.

5.1.1 A Cognitive Psychology Perspective:

The ability to understand the meaning of a concept is called *comprehension*. Problem comprehension is the first and foremost important step toward finding a solution. In order to propose a visualization framework that aids in solving the problem of multimodal time-variant data analysis, the first step needs to assist domain experts to build up *comprehension* of the problem at hand, the available datasets, and the challenges they pose to a solution. In cognitive psychology, the comprehension process is modeled as a composition of *concepts*. Wang et al. [WG10] define a *concept* as a *"basic cognitive unit to identify and/or model a real-world concrete entity and a perceived-world abstract subject"*. This composition of real-world and perceived-world components involves the construction of an internal representation of concepts based on knowledge previously gained in the brain [Mat05].

The construction of a cognitive model for the comprehension process is summarized by Wang et al. [WG10] in the following steps (also depicted in Figure 5.1):

1. **Search for relations:** the knowledge manipulation engine of the brain matches an identified object from real entities to existing *objects* and *attributes* in memory. The brain looks for relations at the image layer and then the abstract layer of memory (boxes 3 and 4 in Fig-

ure 5.1). An *object*, in this context, is an abstract model of a physical entity or an abstract artifact; and an *attribute* is a sub-object that describes detailed properties of an object.

2. **Construction of Object-Attribute-Relation (OAR) model for internal knowledge representation:** the OAR model [Wan07] is a logical formulation of the cognitive model of long-term memory (LTM), which explains its structure at neurological and physiological levels. The process of knowledge acquisition can be formulated as the composition between existing OAR in the brain and the newly created sub-OAR, that is the result of the search from the previous step. The quality of this sub-OAR model depends on the adequacy of relations that have been found (box 5). Ideally, if enough familiarity with the problem domain exists in the brain, a full OAR model can be created which can lead immediately to comprehension. In practice, however, comprehension levels vary between 0% and 100% and only a sub-OAR model is built. At this point, additional information is sought (box 10) from external resources (e.g. the user can look up a dictionary, an atlas map, etc.) and the brain checks again if the findings are good enough to achieve comprehension (box 11) .

3. **Wrapping up the sub-OAR:** if adequate findings are available, the brain classifies them and connects them to appropriate clusters of the entire OAR that exists in long-term memory (box 7), at which point comprehension is considered to have been achieved. Otherwise, incomprehension is identified (box 12), and a new ID is created for the sub-OAR before it is stored in long-term memory for future comprehension. This can lead to formulation of new questions or exploration tasks that can improve comprehension.

4. **Memorization of the new OAR model and its connections:** the results of the completed comprehension process are stored in long-term memory for future reference (box 9).

The search for relations phase of comprehension requires that the visualization system offers selective exploration capabilities that give the expert control over how much of every dataset is being displayed (e.g. through filtering and clutter reduction), and how it is visually encoded (e.g.

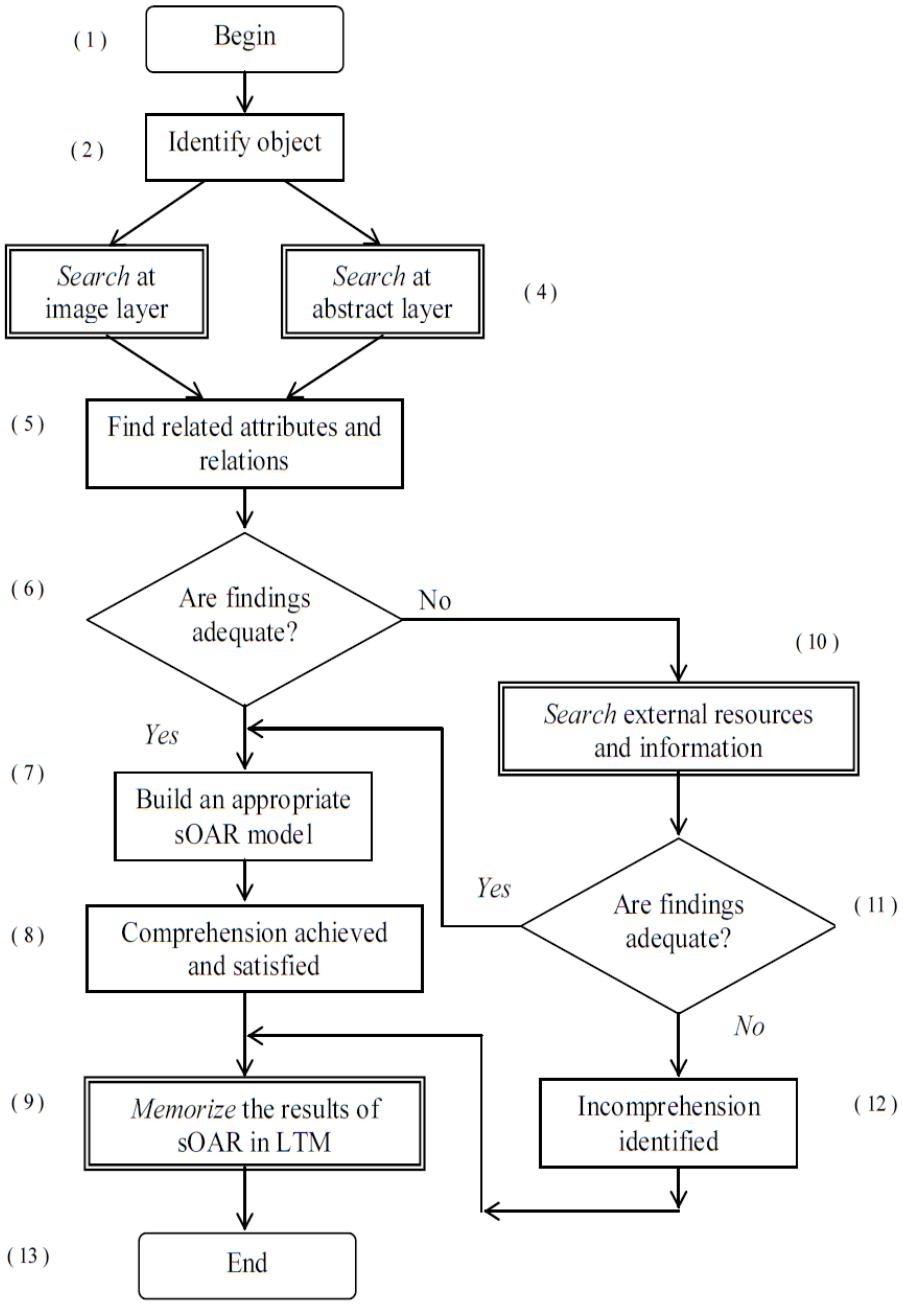


Figure 5.1: The cognitive model of the comprehension process [WG10].

by manipulating illustration techniques). This interactive exploration makes it possible to reach an image that relates to existing knowledge, if such knowledge exists. We identify the first Design Requirement (DR) as:

Definition *(1)*

DR1: Provide interactive clutter reduction and filtering tools to support free form exploration.

The sub-OAR model is constructed as a result of the search for relations step when an object is identified during exploration. For example, a physician may identify the existence of a brain lesion (Object) with a certain size (Attribute) and decide that it can be attributed to a trauma (Relation). At this point, if these findings adequately explain the patient's symptoms, then comprehension is achieved and the physician moves on to generating hypotheses (e.g. treatment methods) and later testing them. If not, then the expert looks up external sources that are outside the scope of the visualization system (e.g. a medical atlas) to improve the sub-OAR model. Alternatively, the user may wish to visually combine multiple data sources to achieve this improved sub-OAR. This leads us to the second Design Requirement (DR):

Definition *(2)*

DR2: Provide user control over *what* combination of datasets can be rendered together, and *how* they are visually encoded.

In conclusion, cognitive models embody the process of learning or adding to the human user's existing knowledge. The construction of a reliable cognitive model from visual exploration tasks is crucial to comprehension and consequently to the solution of the data reconstruction and visualization problem. Interaction is key, and user control over how much of the data is being displayed and how it is visually encoded optimizes the search for relations to accelerate the cognitive model construction process.

5.1.2 A Data Fusion Perspective:

The process of merging information from heterogeneous sources with differing conceptual, contextual and typographical representations is called information integration. Techniques are used in data mining and consolidation of data from unstructured or semi-structured resources. The related term, "data fusion" involves the combination of datasets into a new set of data with the aim of reducing uncertainty and increasing information content. Over the years, the scope of applications that use data and information fusion techniques has increased tremendously from strictly military-related to a broad variety of domains, especially in the commercial and industrial sectors.

The Visual Data Fusion (VDF) Model, depicted in Figure 5.2, was proposed in the 1990's by Karakowski [Kar98] and revised in [BRW07] as an extension to the JDL data fusion model (Figure 4.2). In this model, the human is a central participant in the information fusion process, and

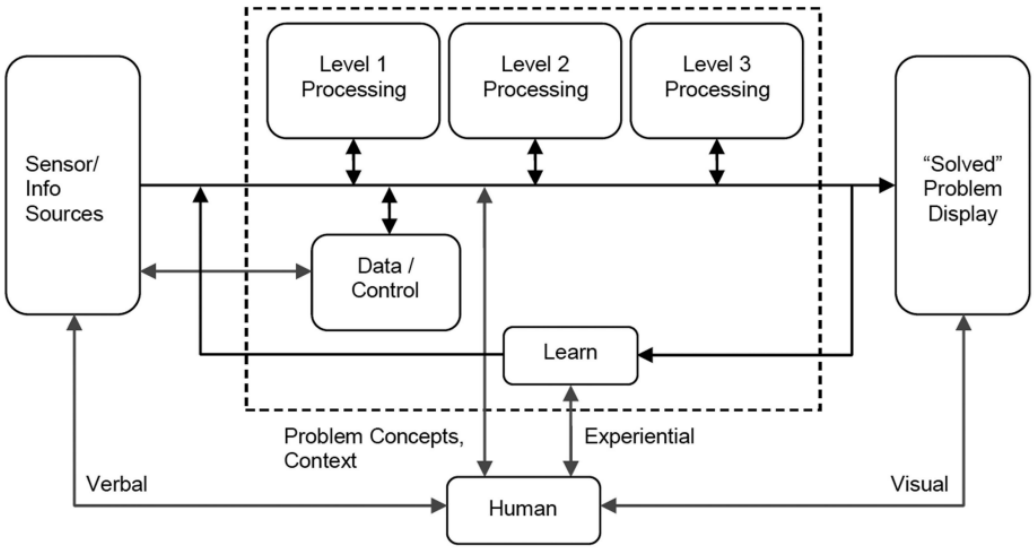


Figure 5.2: Visual Data Fusion Model as proposed in [Kar98].

the whole process becomes a creative problem-solving task. Visualization is primarily used to help the user gain better perception and decide on possible approaches towards solving the problem.

To fulfill this purpose, imagery acts as a channel for perceptual transport that seeks to maximize the information acquired by the user's brain and reduce the amount of information that needs to be visualized by the system. Ideally, this knowledge maximization process starts with little to no prior information required. Through a set of exploration tools, the user begins to comprehend the data and the phenomena it represents, then starts to build hypotheses based on this understanding. To test these hypotheses, the user has the ability to filter out irrelevant data, effectively reducing clutter, and controlling the size of data of interest; which serves for better interactivity of the system.

The basic VDF model is used as a building block for visual situation awareness systems. It is the closest of alternative models to our proposed framework (see [FN13] for a more complete list of alternatives). Our approach builds on motivation for the Visual Data Fusion Model and targets similar goals. We designed a problem-driven system that relates directly to users' needs and perception, and satisfies the following visual data fusion goals:

1. Make Data fusion an active problem solving experience; during which, interaction is key.
2. Focus on Human as a central participant at all stages of operation.
3. Maximize relevant information content while reducing display of information.
4. Tailor the fusion system capabilities for use by all skill levels of users.
5. Provide increasingly sophisticated queries to support hypothesis generation and testing.

The first three goals are mostly addressed by design requirements $DR1$ and $DR2$. Specifically, keeping the human-in-the-loop as a central participant at different levels of operation is ensured by interactive visual exploration tasks. These tasks seek to increase the amount of information communicated *"forwardly"* from the visualization system to the human user. Clutter reduction and interactive filtering enable the maximization of information content in the user's cognitive model while reducing the amount of data in display.

In order to address the fourth goal of visual fusion design, we set the generic requirement of keeping the Graphical User Interface (GUI) simple.

Definition ***(3)***

DR3: Tailor the GUI to different skill levels.

This aims to reduce the cognitive load of the user when trying to figure out how to perform a certain interaction task. More specific GUI simplification guidelines are derived from informal testing, by asking domain experts to use the system to perform certain tasks or obtain insights. For example, one of the domain experts we collaborated with found that time sliders are simpler to use than textual time and date input. Another expert suggested the need to switch between orthographic and perspective views to simplify ROI identification.

High-level data fusion aims at the study of relationships among *objects* and *events* of interest within a dynamic environment. *Objects* in a dynamic environment are something physical that can be perceived, while *events* are observed reality represented in space and time involving objects. Both environment objects and events are represented in the brain as *objects* in the OAR model described in Section 5.1.1. In this dynamic environment, the relationships among objects and events change due to both natural/internal events and external events by players (also called actions). Observing the dynamic behavior of such objects and events is a central topic of multimodal time-variant data analysis. However, such observation and analysis may not be readily possible from existing datasets because of their heterogeneous nature. Visualizing any one of the available datasets alone usually gives partial information about the space-time domain; while naïvely displaying multiple datasets in one view can potentially promote clutter and reduce information content of the display. Interactive selection and clutter reduction tools avoid this pitfall by offering the flexibility of controlling what combination of data items are on display and how relationships among the different components are manifested in the visualization.

As we noted in Section 5.1.1, interactive exploration results in the construction of a cognitive model that relates visualized datasets to user's existing knowledge. Along with information learned

from potential external sources or from combining multiple datasets in one view, the human expert's cognitive model at the post-exploration phase contains maximal information content. This cognitive model creation process is the focus of Chapter 6. The next design requirement then seeks to capture this constructed cognitive model from the human brain back to the visualization system to provide a more complete visual display of information and enable the formation of increasingly sophisticated queries, thus, satisfying the fifth goal of visual data fusion.

Definition *(4)*

DR4: Enable feedback communication from the user's cognitive model back into the visualization.

5.1.3 A Human-Computer Interaction Perspective

The design goals we have described in Section 5.1.2 provide a rationale for our design decisions from the perspective of the problem domain of multi-sensor data fusion. From a Human-Computer Interaction (HCI) perspective, a problem-driven visualization system must address these identified goals of the application domain while also satisfying ones that serve the progress toward problem solution from the domain expert's perspective. In order to do so, a number of visualization design guidelines have been adopted in the literature. In this section, we describe the guidelines that are most relevant to our design decisions.

Sedlmair et al. [SMM12] identified three main contributions of design studies that characterize problem-driven visualization research. In what follows, we describe these contributions and explain their implications on our design strategy.

i. Problem characterization and abstraction

In problem characterization, the visualization researcher seeks to reach a common terminology with domain experts to be able to explain the problem in terms that are meaningful to the end user. The next step is then to abstract this characterization into a mode generic description that can appeal to a broader audience and enable the communication and dissemination of information from the visualization system to the research community or the public. One of the most effective ways for this characterization in scientific visualization is through interviews and observation sessions with domain experts [SBM92].

Problem abstraction can be further broken down into two specific abstraction steps:

(i) **Data abstraction:** the system must not make assumptions about how to abstract and visually encode the available data. Common intuition from a visualization design perspective may not necessarily serve the cognitive model that domain experts seek to construct. Instead, the system should provide a variety of abstractions and enable the user to switch from one abstraction to another to make decisions about what constitutes a meaningful visual encoding, that connects to their existing knowledge in the domain. This decision typically involves data transformation, derivation, and even creation of new datasets. Ideally, the transformed, derived or generated data can ultimately lead to automated solutions at later stages of the analysis that serve in hypothesis testing.

(ii) **Task abstraction:** Tamara Munzner summarized the terminology overload for the word *"task"* in the literature by classifying tasks on the basis of abstraction level and granularity [Mun09]. A high-level domain task denotes a problem that needs to be solved such as curing a disease, or providing good user experience. In a low-level domain task, the problem is one of a more concrete nature with a specific sequence of steps to solve it. For example, developing a treatment protocol to cure the disease, or analyzing session logs to generate hypotheses about user satisfaction. Abstract tasks are more generic operations that can be

beneficial to a variety of application domains and range from low-level operations like comparison, correlation, querying, etc. to high-level ones like testing hypotheses, identifying regions of interest, deriving system parameters, and exposing uncertainty.

Task and data abstraction, together, play a fundamental role in visualization design. In an ideal scenario, high level information exists on the computer and crisp low-level tasks are fully specified which requires no user intervention to solve the problem and makes full automation the most effective solution. However, this is not the case in the majority of real-world application domains. In a typical analysis scenario, domain tasks are ill-defined and exploratory in nature; and the data is only partially available, which makes human induction a necessity. Visualization systems seek to provide cognitive aids to this induction process. Therefore, a *complete* visualization system gives the user the flexibility to switch between different levels of abstraction.

Definition *(5)*

DR5: Provide different levels of data and task abstraction, and enable derivation of new data.

---

ii. Validated visualization design

The outcome of problem characterization and abstraction is a set of visualization and interaction techniques that, when combined together, form a framework which guides the development of a visualization system. The system design needs to be validated with evidence that it does in fact help solve the target domain problem and that end users can truly benefit from it.

Munzner [Mun09] noted that validation aims to avoid pitfalls at the problem characterization level, the abstraction level, the visualization/interaction technique design level, and the algorithm level. Design Requirements $DR1$ through $DR5$ that we have specified for our framework avoid such pitfalls by ensuring user involvement in every step of the exploration and problem solving process. Specifically, at the problem characterization and abstraction levels, $DR1$ and $DR5$ ensure

that user-guided characterization is supported through interactive exploration and that a diversity of abstraction levels are offered for the user to select from. At the technique and algorithm levels, $DR2$ ensures user control over the visualization technique and $DR4$ provides user-specified parameters to be fed back into the system to customize the underlying interpolation and rendering algorithms.

According to Sedlmair et al. [SMM12], validation crosscuts a nine-stage framework for design study methodology, in which three phases can be outlined for validation as shown in Figure 5.3. In the precondition phase, the visualization researcher performs validation at a personal and collaboration level to ensure she has a good understanding of the problem and adequate communication with domain experts and collaborators. At this phase, actions for validation include interviewing and observing domain experts and the way they interact with data to evaluate the current problem characterization and abstraction.

In the core phase, validation is inward-facing, meaning that it emphasizes the verification and justification of important visualization/interaction design decisions. Actions that serve this phase include matching these design decisions to established perceptual and cognitive principles, soliciting expert review through qualitative discussion or in the form of a design study paper to ensure that no established guidelines are being left out or violated by the design. Furthermore, informal testing with users in the target domain can help pinpoint usability issues and collect anecdotal evidence that the system meets its design goals. Last but not least, outward-facing validation is the highlight of the analysis phase; with the aim to justify visualization results to the outside world.

In order to support validation at these phases, the proposed framework aims to maintain expert feedback at all stages. Anecdotal evidence is collected through preliminary testing to uncover potential pitfalls at the earlier stages of design. User feedback on usability is incorporated to better fit users' analysis goals and support their comprehension. This leads to our sixth design requirement:

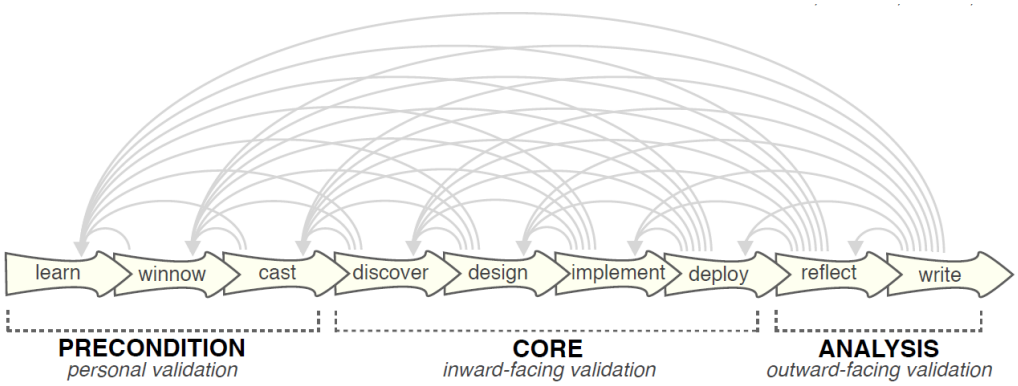


Figure 5.3: Nine-stage framework for design study as suggested by [SMM12] with three encompassing validation phases.

Definition ***(6)***

DC6: Incorporate user's feedback on system usability at different stages of design.

---

iii. Reflection on design

The main purpose of reflection is to reiterate on lessons learned from the design, development, and evaluation of a new visualization system. Such experiences can help improve current guidelines that exist in the literature, propose novel theoretical frameworks, and present opportunities for paradigm shifts. Reflections from our initial system design and early informal testing led us to some requirements that are in accordance with Ben Shneiderman's early description of direct manipulation [Shn93]. For instance, we have observed that domain experts favored selection by pointing rather than typing in all exploration tasks. More than one expert has reported that moving sliders, probes and other GUI components was less burdening and more appealing than typing in specific values.

From this observation, we adopt the following classic design principle from the literature:

Definition ***(7)***

DP7: Provide selection by pointing, not typing, and maintain continuous visual display of status.

5.1.4 A Computer Graphics Perspective

Volume rendering methods can present depth information of partially occluded structures by rendering the overlapping entities semi-transparently. Without appropriate enhancement, however, these methods fail to provide the correct spatial correlation if the images presented are visualizing complex data with many layers and fine features. User interaction can help alleviate the problem through rotation and free navigation in the scene visualizing complex data. Other interaction techniques such as ones defined by Shneiderman's mantra [Shn96] of overview first, zoom and filter, then details on demand, can provide adequate insight.

Generally speaking, interaction in scientific visualization techniques aims at incorporating user expertise in the analysis process and aims to maintain an active channel of information communication between the human user and the computer. Therefore, in an ideal visualization, the initial view of the world should give an overview of the context while simplifying interactive exploration tasks to guide user comprehension. Rendering adequate representation of complex time-variant $3D$ datasets at interactive rates is one of the major challenges we address in this work. We seek to keep calculations for filtering, data reconstruction, and rendering on the same graphics hardware to minimize data communication with the CPU and maintain interactive rates despite the large size of the data.

Definition ***(8)***

DC8: Minimize data transfers between CPU and GPU to support interactivity.

5.2 Overview of the Proposed Framework

We propose a framework for the visual fusion of multimodal time-variant three-dimensional datasets. The ultimate goal of this visualization framework is to enable users to answer questions about the spatio-temporal behavior of physical phenomena that the collected and simulated data represent. Answering these questions normally would require the availability of sensor-measured data at all times and all spatial locations at which a phenomenon of interest occurred. Since this is not the case for any given modality, we propose a framework that incorporates expert knowledge with data obtained from different sources to fill the gaps in the data and provide a coherent story to the user. Our solution accomplishes this through three visualization and interaction phases (Figure 5.4):

1. *Exploratory Analysis:* During this phase, the system enables the integration of multiple data sets in a common reference frame while providing filtering and clutter reduction capabilities. The goal is to build up and improve experts' understanding of the real world phenomena being represented, as well as of the strengths and shortcomings of different modalities in contributing to this representation. In order to fulfill that, user interaction is crucial in all exploration tasks, including filtering, clutter reduction, etc. The output of this exploratory analysis is a cognitive (sub-OAR) model that captures users' comprehension and supports hypothesis generation about which modalities can be most informative to include in a "reference model", and ways to combine them.

2. *Data Tracking and Reconstruction:* The cognitive model constructed in the previous phase is communicated back to the visualization system through a set of identified and extracted Regions Of Interest (ROIs) and interpolation parameters. The ROIs can be combined together using nearest neighbor calculations to generate a new reference dataset; or they can only guide the derivation of the new reference dataset. That is, the reference set may be fully generated within a space-time ROI identified by the user. An example would be for the user to decide to generate a number of points within a spatial region at a certain temporal

resolution. This generation process can span a temporal range of interest during which the phenomenon under study is believed to have persisted. One such example is described in detail in Chapter 7.

Either way, the generation of this reference dataset requires more user intervention to specify interpolation parameters and decide on the desired spatio-temporal resolution. The resulting reference model captures expected behavior of the studied phenomena, based on expert knowledge and the available data sets, and acts as a prediction for the tracking of ground truth measurements. Given this dataset, the positions of ground truth measurements can be tracked by mapping them to their neighborhood in the reference set. This effectively fills the gaps in the dynamic evolution of the data and provides a more complete understanding of how the detected points must have evolved over time.

3. *Illustrative Visual Feedback:* Both raw data and tracked results are fed to the illustrative rendering subsystem; in order to provide immediate visual feedback for exploration tasks and tracking calculations. By providing a diversity of potential visual encodings for every dataset, the system enables the user to generate different pictorial representations. This enhances the search for relations in the comprehension phase of the analysis by increasing the chance to find a match in user's existing knowledge. Informed by exploration tasks, illustrative enhancement techniques that control visual encodings of the data being displayed also further improve the cognitive model built from both exploration and tracking.

In all three stages, interactivity is crucial to capture human expertise in the process and to enable visual analytics tasks that guide adjustments to fusion parameters, in order to yield a visualization that tells an as accurate as possible story of what happened in the real world.

Figure 5.4 outlines the framework and groups together stages that require a unified user interaction scheme to achieve a specific goal (e.g. model creation, feature tracking, or model rendering).

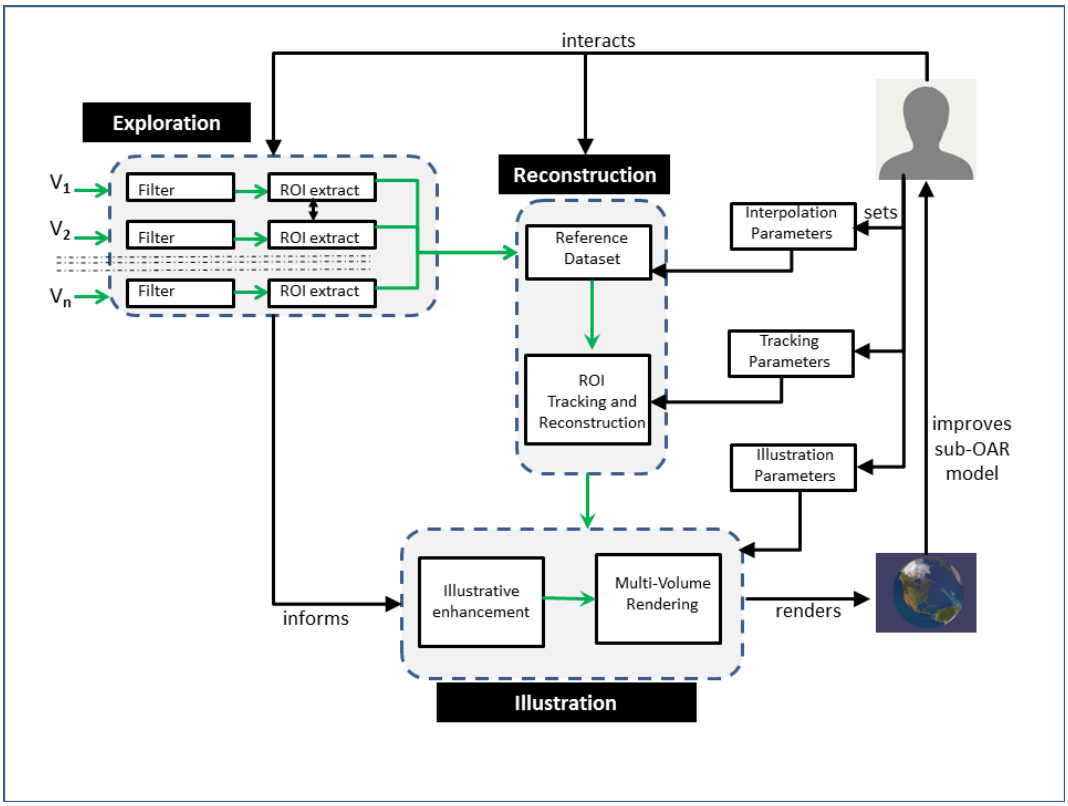


Figure 5.4: Data visualization framework. Green arrows mark data flow.

Each identified goal (refer to Figure 1.7) is mapped to a visualization subsystem to guide the implementation process for further investigation and evaluation of our proposed framework. The first goal is addressed by an **Exploration Subsystem (ES)** which acts as the entry point for visual analytics tasks involving large datasets. An effective exploratory analysis subsystem seeks to help the user answer the question of: *What constitutes a representative model of what happened in the data?* Chapter 6 describes the ES in detail and demonstrates how it answers this question through serving the first two steps of problem solving, namely : *comprehension* and *hypothesis generation*.

Once a reference model is extracted, a **Tracking Subsystem (TS)** uses this model as a guide to track data points from other datasets and to fill in the spatio-temporal gaps. An important question that should be answered by the Tracking Subsystem then is: *For a feature/point of interest detected at time t_0, what was the location of this same feature/point at time t_d ?* Chapter 7 will demonstrate the TS and how it answers this question in light of a case study that we describe in Section 5.3. Answering this type of question effectively enables the expert to test hypotheses, check results based on the constructed model, and possibly iterate back to improve the model.

Finally, the **Illustration Subsystem (IS)** focuses on adequate visual representation of the reconstructed data representing ROIs. It seeks to answer the question of: *What is the optimal visual representation of the reconstructed $3D$ data from all sources?* Since this subsystem serves both the ES and the TS, we will not dedicate a separate chapter for illustrative techniques. Instead, we will describe those techniques and outline their potential in improving the user's cognitive model during exploration (Chapter 6) and in tracking (Chapter 7).

5.3 Case Study: Atmospheric Data

We address the problem of missing data reconstruction in the context of an atmospheric science problem. The datasets represent ash detections in the atmosphere following the 2011 eruption

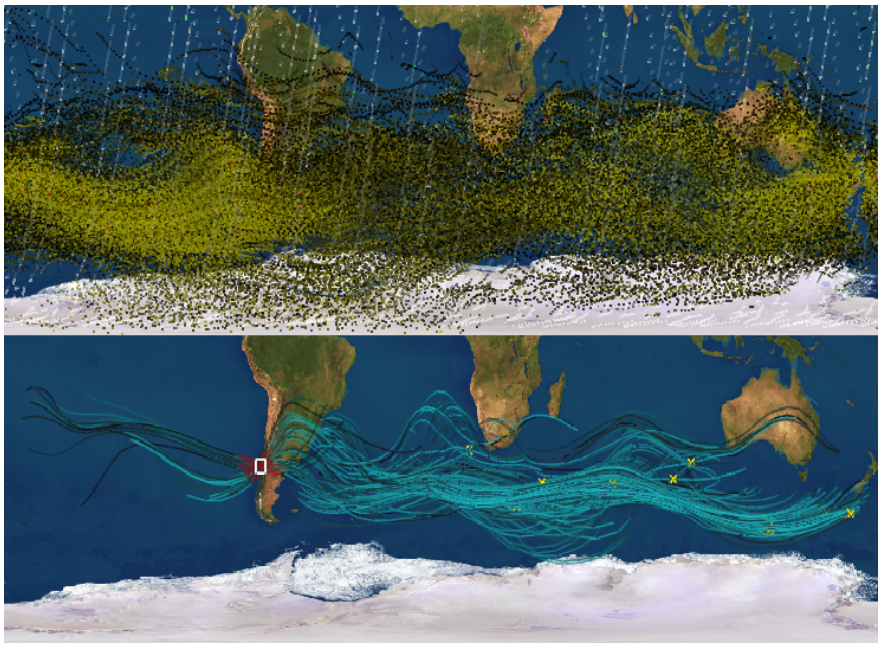


Figure 5.5: CLaMS trajectories seeded at volcanic ash detections from 0 S to 90 S: **Top:** Simulation points (shades of green), and MIPAS measurements (white) filtered by time between June 12 at 13:00 and June 15 at 16:00 **Bottom:** Simulation pathlines filtered by StreamProbe technique [EGG+14] around the eruption location with trajectory seeds at MIPAS detections marked by yellow X's.

of Puyehue Cordón Caulle volcano in Chile. Visualization aims at revealing the ash plume that resulted from the volcanic eruption and were injected into the atmosphere. Interest in visualizing such a plume and how it evolved over time is motivated by the fact that ash particles pose severe danger to aircrafts. Experts seek to understand the behavior of volcanic ash in the atmosphere to make decisions and find means to reduce the adverse effect that ash has on Air Traffic Management. To this end, our visualization aims to provide a coherent story about the ash plume ejected into the atmosphere, while enabling the experts to interact with it, view it at any user-defined time instant, and predict its future evolution. We start by describing the datasets that were used, and the challenges posed by the data.

5.3.1 Datasets and Challenges

The datasets made available through the IEEE Scientific Visualization Contest of 2014 [IEE14] contain both a simulation set and satellite measurements from June and July of 2011 following the eruption. Simulation data are obtained from The Chemical Lagrangian Model of the Stratosphere (CLaMS) [MKG$^+$02]. It contains trajectories simulating individual air parcels including vertical transport. The trajectories are calculated using a fourth order Runge-Kutta scheme and are seeded at volcanic ash detections captured by The Michelson Interferometer for Passive Atmospheric Sounding (MIPAS) aboard the European Space Agency's Envisat satellite [GHSR14].

Figure 5.5 displays two subsets of CLaMS trajectories. The top view is filtered by time only while the bottom view is further filtered by an intersection with a region of interest around the eruption location and time. The seed points are marked in yellow but are hard to spot in the top view due to clutter. However, clutter is not the only challenging aspect of the CLaMS dataset. Like many simulation datasets, the trajectories are irregular in space and time which makes this dataset a poor candidate for interactive fusion. Correlating these trajectories to regions of interest in other datasets is computationally expensive.

The Atmospheric Infrared Sounder (AIRS) [HGM14] is one of the instruments aboard NASA's Aqua satellite. Figure 5.6 shows sample readings on two different 12-hour periods from AIRS data. The detected ash plume is shown in blue. Like the majority of satellite datasets, this set in its original form is of limited value for the analysis of the temporal behavior of the plume. This is primarily due to data discontinuity. In the particular case of the Puyehue Cordón Caulle eruption, the direction in which the satellite revolves around the Earth is opposite to the direction in which the plume is evolving. Therefore, detections found by AIRS at a certain location can only be captured again 12 hours later. A lot of change may have occurred during these 12 hours. Given AIRS data alone, there is no way for the expert to analyze such change. This challenge is clearly depicted by the discontinuity of the plume near Australia shown in the lower right corner

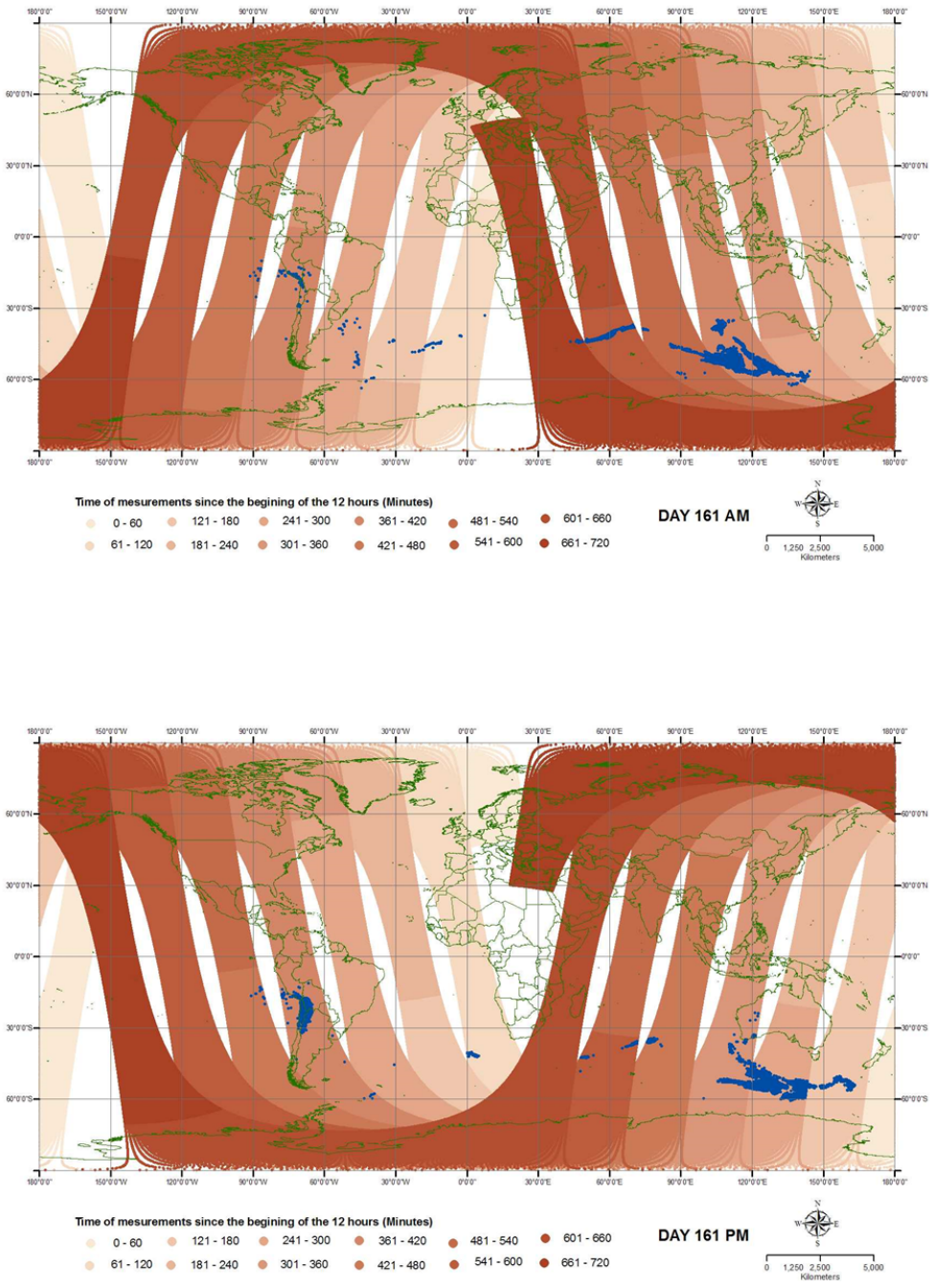


Figure 5.6: Sample 12 hour AIRS coverage on June 10th 2011. Morning measurements are displayed on the left while afternoon readings are on the right. Shades of brown encode time while the ash plume is shown in blue.

in Figure 5.6. To address this problem and reconstruct the ash plume, Günther et al. [GSFT14] (the winning entry of the Contest) applied spatio-temporal interpolation of AIRS data in a pre-process. Since their interpolation is based on AIRS data alone, it does not readily provide altitude information. In a later step, they used CLaMS and MIPAS data to create a probability distribution of ash concentration on the z-dimension. Unlike their approach, our work focuses on speed and interactivity in the missing data reconstruction process to support timely decision making. We perform both spatio-temporal interpolation and altitude information reconstruction on the GPU by projecting raw AIRS data points on a synthetic plume model.

Chapter 6 describes the Exploration Subsystem through a task-oriented approach. By splitting the tasks into three main groups, we describe how our system meets the Shneiderman's mantra of overview first, zoom and filter, then provide details on demand. Next, the Feature Matching and Tracking Subsystem is covered in Chapter 7, and Chapter 8 summarizes our system support for visual enhancement and Illustration tasks.

Chapter 6

Exploration Subsystem

"Despite the curator's doubts, I am not going to give up my search for the lost tomb just yet."

Egyptology. *Emily Sands. Cairo, Egypt. 1926.*

In this Chapter, we present the first component subsystem of our proposed framework in detail while explaining its features with an example. Using the case study we described in Section 5.3, we illustrate the system's capabilities by applying it to this real-world problem that has been presented to the VIS community as a challenge in the 2014 IEEE Scientific Visualization Contest. A detailed data description was given in Section 5.3. We use the processes outlined in our proposed framework to solve the problem through visualization and interaction techniques that we developed and implemented at each stage. Our visualization system design follows Shneiderman's Visual Information Seeking Mantra [Shn96] of overview first, zoom and filter, then details on demand. Further, we take into account the fact that data from multiple sources are displayed together in one view which necessitates the use of different visual encoding strategies for each.

The main goal of the visual exploration subsystem is to aid users' *understanding* of the available datasets through interaction. Consequently, this understanding contributes to the construction of the user's cognitive model, which in turn enables the construction of a reference model, effectively fulfilling the first goal of our visualization system (G1 as described in Section 1.3.2). Visualization as a cognitive aid is used to support this cognitive model construction phase through empowering the user to identify the shortcomings of existing datasets in answering initial questions, formulate new questions, apply logic, look up other sources of information, and reflect on their understanding so far.

Shneiderman's information seeking mantra [Shn96] offers a foundation for the successful design of a visualization system that serves cognitive objectives of comprehensible, predictive, and controllable human-computer interfaces. It presents design principles that start with *Overview first*, effectively providing a general context for understanding the dataset(s), their major components, and their relationships to one another; then proceed to enabling *zoom and filter* capabilities to eliminate extraneous information and decrease cognitive load; and finally underline the importance of obtaining *details on demand* to answer specific analysis questions.

The main goals of these principles and their underlying techniques are to aid exploratory analysis tasks, bridge the knowledge gap for the user to support better understanding and create mental models of trends in the data. User's understanding is crucial for the output of our Exploration Subsystem, which is the process of representative (or *reference*) model formulation. We note here that the representative model can be derived from one of the original datasets and fed to the Tracking Subsystem or it can combine data from multiple sources to create an as complete as possible reference. An inaccurate cognitive model can mislead the user into selecting or creating a representative model of poor quality which can subsequently hinder the accuracy of the tracking process and result in an unsuccessful data reconstruction and visualization.

For overview and data integration, GUI components are designed and implemented to pre-filter a dataset, load and unload entire datasets or parts of them, slide through time using unified sliders, change attributes of individual datasets to support visually matching patterns among them, and switching from one view to another. We describe these capabilities in Section 6.1. As for the "zoom and filter", a set of real-time data filtering and ROI identification techniques are provided and elaborated upon in Section 6.2. Next the "details on demand" principle is covered in Section 6.3 along with the techniques we propose to fulfill this purpose.

6.1 Overview and Data Integration

Our proposed framework sets a great deal of attention on data integration and exploration tasks. The Exploration Subsystem, which acts as the main entry point, integrates multiple three-dimensional datasets in a single view while providing a unified reference frame for visualization and interaction. Figure 6.1 shows the Graphical User Interface components relevant to the concepts presented in this Chapter. These components can be described as follows:

1. **Dataset Browser:** a side pane that is populated with dataset names and event names. The user is given the freedom to either load an entire dataset of a certain modality, by highlighting this modality in the tree widget then pressing the "Load" button, or to pre-filter the data according to a particular event, by selecting a child node of the event that corresponds to the modality in question. Pre-filtering the data can be useful in some cases but for data overview purposes, the user is more likely interested in loading entire datasets at this phase. We note here that the knowledge of filtering criteria for data points corresponding to a given event may not be available a priori for some datasets. An example is the AIRS dataset. In which case, we enable the user to either load data within a specific time frame (e.g. load AIRS files covering from 6/6/2011 to 6/14/2011), or to select an event and the system automatically loads data that was captured within a week from the time of that event.

2. **Data Renderer:** an OpenGL widget that initializes a separate renderer thread to take the

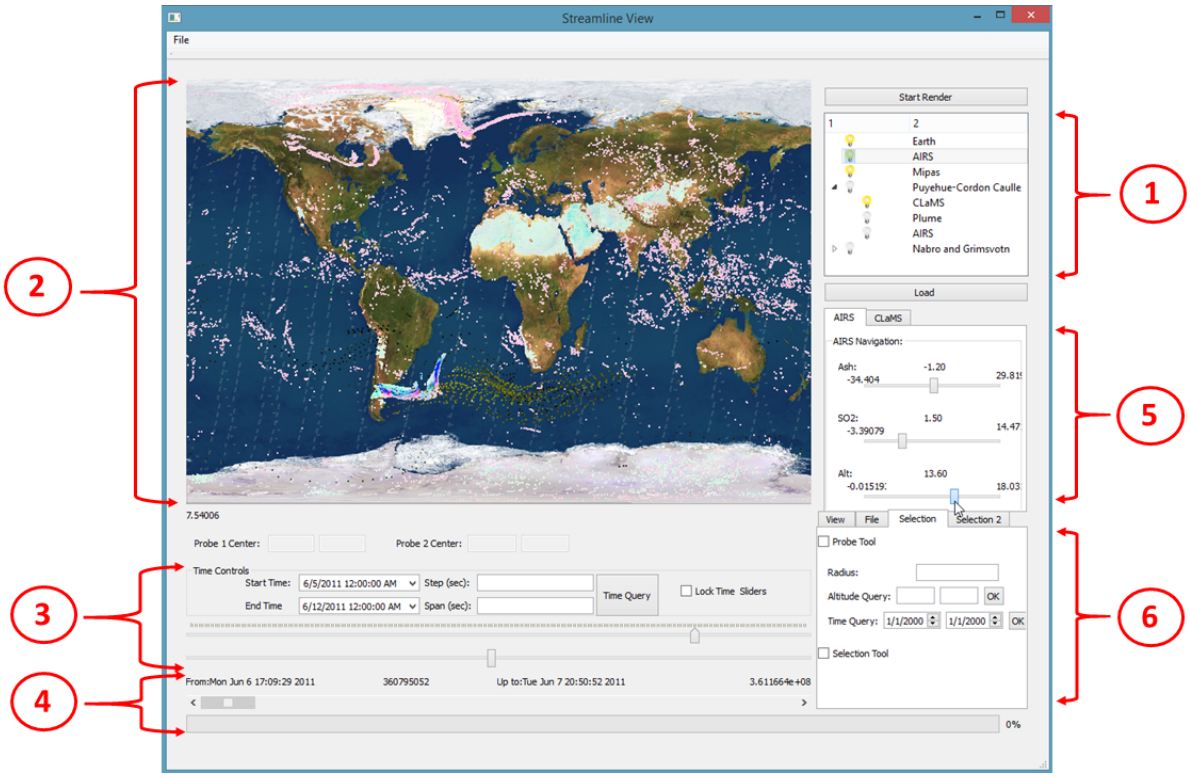


Figure 6.1: System Overview: *(1)* Data Browser, *(2)* Renderer, *(3)* Time Controls, *(3)* System Feedback, *(5)* Attribute Sliders, *(6)* View and Selection Tools.

rendering workload away from the main thread. This frees the GUI system from whatever the renderer is doing and makes it highly responsive to user commands. This separation makes our system more expandable in terms of providing the ability to add more views, in cases where the need for multiple data views arises. The render thread uses a shader program to execute the OpenGL's rendering pipeline (Figure B.3 in Appendix B). All data processing and rendering tasks are performed in GPU shaders to speedup performance.

3. **Time Controls:** Time navigation is crucial for exploration tasks in any time-variant data. We implement two time-sliders, for which the user can set range parameters including specifying start/end date and time to control the time span of both sliders and the size of time-step for each. Moving the top slider shows only data with timestamps less than the current slider position, which defines an "end" time threshold, while the bottom slider defines a "start" threshold, against which earlier data points are filtered out. The user can move these sliders independently to define a start and end time for the data in display or lock the sliders together so they define a sliding time window, smoothly animating data points. This is implemented by passing slider values as read-only shader invariants, and letting the vertex or geometry shader perform the culling based on those thresholds.

4. **System Feedback:** Visualization systems are among software applications in which system responsiveness is crucial to gain users' trust and maintain human interest. Therefore, the system feedback section includes a progress bar (bottom part) to provide visual feedback on lengthy processes such as data loading/file streaming. More display widgets provide both textual and numerical representations of the start and end date and time for the time frame under display as specified by time sliders. Further feedback is provided through other parts of the system like in the data view, for example, through cursor location in the datasets' world coordinates.

5. **Attribute sliders:** Similar to time sliders, these sliders are used to cull away geometry corresponding to a value above or below a certain threshold. However, the main difference here is that data included in the display is mapped to color and opacity values that are proportional

to how far an attribute is from the threshold. As the name implies, this thresholding is not based on time stamps but on attribute values. More on this in Section 6.2.

6. **View and Selection Tools:** This section is where major interactive filtering gets setup. Multiple tabs allow for different ways to create and specify criteria for interactive filters (probes), or use a selection tool that enables the selection of specific data points and have the trajectory for their temporal evolution exclusively displayed. The "View" tab enables switching between different projection methods (orthographic or perspective) and different views (flat map or spherical Earth).

Overview tasks can be further assisted with InfoVis tools like ComVis [MFGH08] to display aggregate information in multiple views and support multi-dimensional information visualization. This complementary relationship between information and scientific visualization has been proven successful in [MWSJ13] through brushing and linking between a graph view and a 3D render of the flow field. In our framework, we advocate a loose interaction between the two systems through a file exporter embedded in our system that is capable of writing .comvis files. Feeding data back from ComVis to our framework is also possible through exporting selections from ComVis and loading them in our $3D$ renderer.

6.2 Zoom and Filter

Zooming and filtering primarily aim at enhancing users' understanding of information regarding a region of interest (ROI), what types of events occur in it, when, for how long, and how they behave within the ROI versus outside of it. Similarly, the user may be interested in studying the events themselves regardless of any specific ROIs. In that case, questions that are sought after with zoom and filter exploration are: whether an event exists in a certain space at a certain time, how long it persists, and what is the sequence of events. Whatever the target, zoom and filter capabilities have a strong potential in enabling ROI identification tasks by reducing clutter and focusing user's

attention on data items of interest.

Multiple zooming and filtering methods are proposed in our system. We describe the proposed methods here in light of their ability to aid user's comprehension and cognitive model construction process by answering questions about *datasets*, *events*, and *Regions of Interest (ROIs)*. We then study the ability of visual fusion of multiple datasets to combine these answers into forming hypotheses about all three types of questions.

6.2.1 Data-Specific Questions:

Every dataset has its own coverage patterns of the space-time domain. Identifying the strengths and shortcomings of every available modality separate from all others comes as one of the initial exploration goals for multimodal data analysis. This information may not be readily available through displaying entire datasets either due to clutter (Figure 6.2) or lack of adequate visual encoding (Figure 6.3).

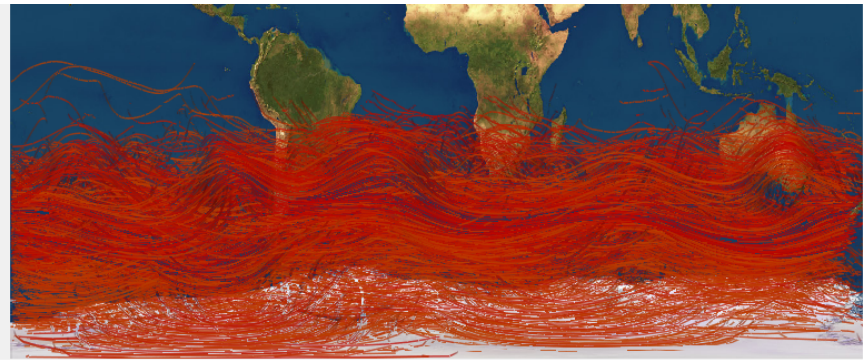


Figure 6.2: Clutter when displaying all trajectories for CLaMS simulation in the southern hemisphere.

Simulation usually results in rather dense flow fields. Different seeding strategies were introduced in the literature to reduce the clutter, as was discussed in Chapter 2. Figure 6.2 gives an

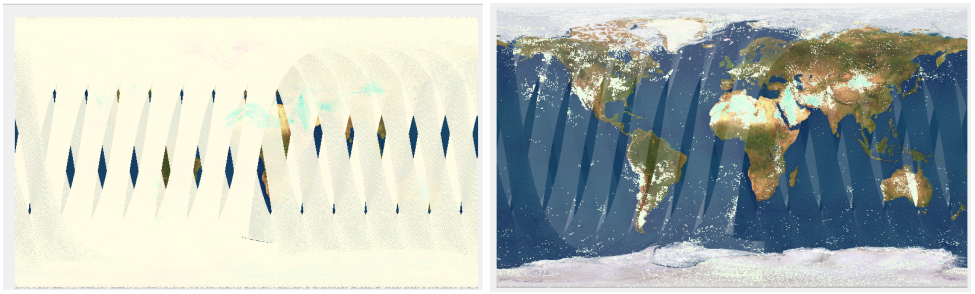


Figure 6.3: AIRS detections captured in one day with naïve visual encoding (left), and with visual encoding using color and opacity mapping to ash and sulfate aerosol ratio thresholds (right).

example of a typical set of pathlines that result from placing seed points and tracking their motion in a dynamic flow. Every pathline/trajectory is considered a locus of motion for its corresponding seed. From this view, it is almost impossible to answer questions like:

1. Where does a specific pathline start and end?
2. How long does the seed exist in the simulated flow?
3. Do all pathlines have similar lengths?
4. How are pathlines organized in file?
5. How does a group of consecutive seed pathlines evolve over time?

To answer dataset-specific questions like these, every pathline in the dataset is given a unique identifying number. A preliminary exploratory selection tool aids in selecting pathlines based on their identifiers. Figure 6.4 shows a sample of user-selected pathlines (numbered 1, 150, and 1050), that address questions *(1)*, *(2)*, and *(3)* from above. While in Figure 6.5, pathlines with a range of IDs (numbered 2 to 12) are shown in two different views to gain some perspective on their organization in file, and their evolution over time and space (questions *(4)* and *(5)* above).

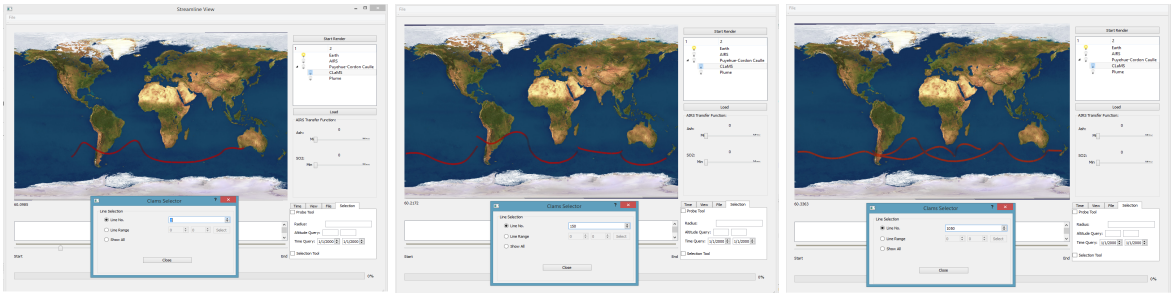


Figure 6.4: User-selected pathline No. 1 (left), No. 150 (middle) and No. 1050 (right). User can answer questions (1), (2), and (3) using this selection tool.

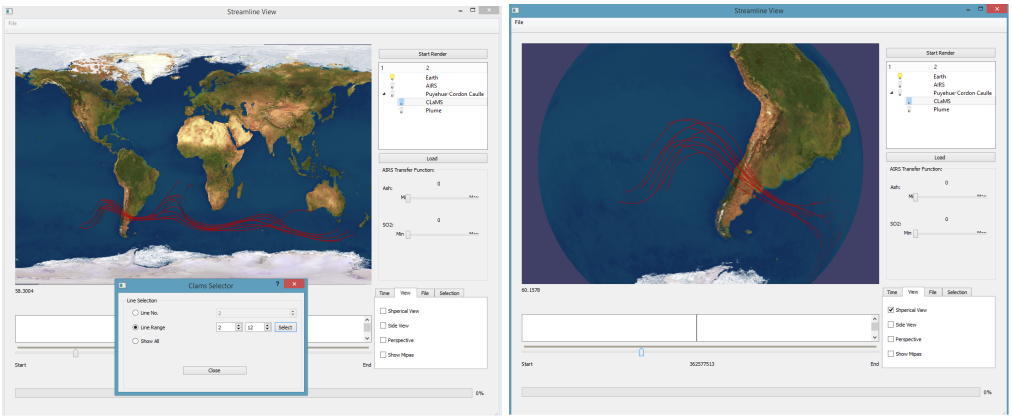


Figure 6.5: User-selected pathline Nos. 2–12 shown in flat map view with orthographic projection (left), and in spherical view using perspective projection (right).

More data-specific questions arise from the view of AIRS satellite data displayed in Figure 6.3. Examples are:

1. What constitutes a set of data points of interest?
2. How does data coverage evolve over time?
3. What are the shortcomings of this dataset?

To answer the first question, one way to filter the data is through attribute threshold sliders as we noted in the previous section. Attribute sliders are implemented by passing their values to the vertex shader, which performs a test on a per vertex level, effectively testing attribute values for different vertices in parallel. For any given vertex, the shader program tests its attribute value against the location of the corresponding slider, and calculates this point's color and opacity values based on its distance from the current threshold. This visual encoding is depicted in the right image of Figure 6.3. As the user moves the sliders for ash index and sulfate aerosol index, the color and opacity values of every point are calculated in real-time. This offers a variety of pictorial representations that can be linked against the user's existing knowledge. One domain expert, for example, noted during informal testing that the values selected by the sliders gave an adequate visual representation of the ash cloud above the Sahara Desert in the Northern part of Africa. This can help the user understand what range of values in the dataset are of interest to a specific phenomenon.

More sophisticated transfer functions can be adopted in this phase of the analysis to detect different features of interest. Interactively tuning such transfer functions is crucial to the search for relations step of the cognitive model construction (refer to Figure 5.1).

To answer the second question about how data evolves over time, time filtering enables the user to scroll two sliders and only view data points that were captured between the start and end times

specified by those sliders. The "up to" time slider is of particular interest to this question as it creates a smooth animation of new data arriving in display as the user moves the slider. Figure 6.6 shows a sequence of images captured as user scrolls the "up to" time slider.

For the last question in this category, the shortcomings of every dataset become evident as the clutter is reduced and more focus is set on data points or regions of interest. For example, Figure 6.4 reveals that CLaMS trajectories have a diverse range of trajectory lengths and time spans (some trajectories only travel from West to East, others go around the globe and return to the same point). Spatio-temporal data structures that are used to organize this data and accelerate access to it should take this fact into account and make no assumptions about temporal uniformity. Another example can be seen in the bottom row of Figure 6.6, where it becomes clear that ash clouds detected by AIRS (see the ones above Sahara Desert) should not be considered with the same shape they appear within this 12 hours display. Rather different strips of these clouds were captured at different times. Changes between the times of acquisition are missing from AIRS data and need to be reconstructed by tracking those detections.

6.2.2 Event-Specific Questions

A limitation in simulation data is that it contains trajectories that may or may not relate to a specific event of interest. In many cases, those trajectories are seeded at points of interest but to attribute these points to specific events or phenomena is a difficult task. In the case of CLaMS data, for example, simulation trajectories are derived from a flow field that simulates the motion of air parcels in the atmosphere. The specific trajectories are obtained by seeding the field at locations where an ash detection was captured by the MIPAS satellite. This seeding then tracks detected points in the flow back to a certain point in time. The detection can either be traced back to a specific event or not.

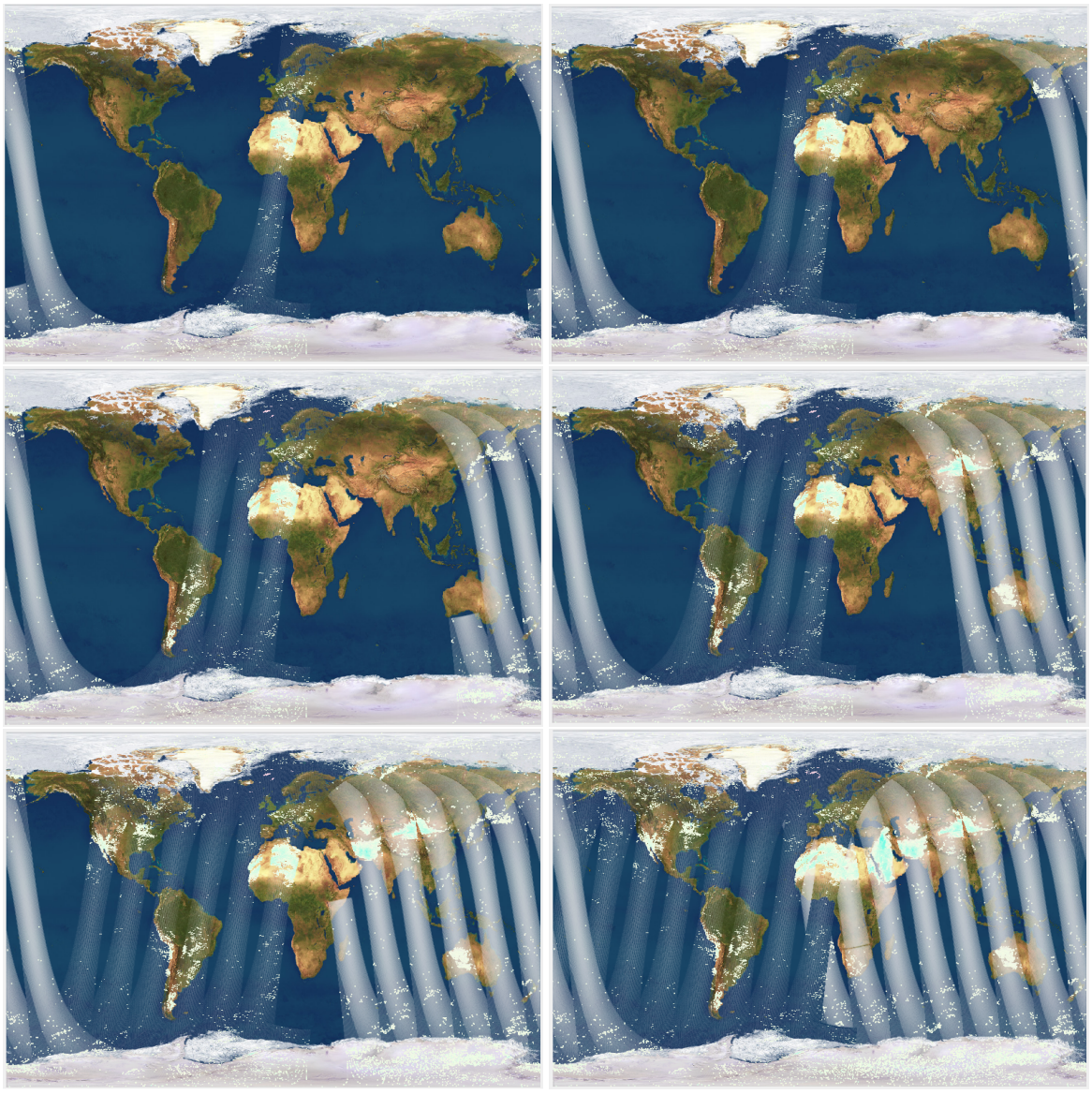


Figure 6.6: Time scrolling trough AIRS detections captured in 12 hours on June 1^{st}, 2011.

Ash detections can be present in the atmosphere as a result from different events that happen over the course of months. Ash stays in the atmosphere and doesn't get dissolved in water. Consequently, the task of attributing a set of trajectories to a specific event that took place at a certain location and time can be a complicated task, especially if the exact location of the event is not known a priori. Our system offers the ability for the user to probe localized parts of the flow to examine those specific regions and investigate which parts of the flow can be attributed to them. Previous work in the literature has addressed this regional flow field exploration problem either by offline enhancement [GRT13], or through organizing field pathlines in a hierarchical data structure which facilitates interactive selection [MWSJ13]. Both methods require long pre-processing times (order of minutes) and cannot handle well large datasets with tens of thousands of lines. We propose "StreamProbe" [EGG+14] to address these issues.

"StreamProbe" a GPU-Based Technique for Flow Field Exploration and Clutter Reduction

Time-variant data can be obtained in the form of a flow field that represents spatial displacement of individual data points across time. However, obtaining geometries to describe motion on a per-point level results in a dense visual representation which hinders user's ability for sense making and cognitive model construction. For these cases, we seek visualization and interaction techniques that aid user exploration and clutter reduction within the dense representation.

Flow visualization aims primarily at the occlusion-free and controllable visual exploration of dense flow fields. Flow fields are usually visualized using streamlines (steady-state fields) or pathlines (time-dependent flows). The challenge is how to highlight crucial features while reducing clutter to convey relevant trends to the expert user in a timely manner.

Dense flow field visualization has been an important research question in scientific visualization for several decades. Effective visual representation is crucial, due to the massive amounts of vector field data available through scientific simulation and measurement modalities in many application

domains. Examples include velocities of wind and ocean currents, results of fluid dynamics simulations, magnetic fields, blood flow, cell migration during embryo development, and components of stress and strain in materials. However, existing visualization techniques have fallen short in addressing the full complexity of the flow field visualization problem, particularly the issue of clutter reduction both in 2- and 3-dimensional fields.

Traditionally, vector fields have been visualized using texture-based methods. Examples include spot noise [VW91] and LIC (Line Integral Convolution) [CL93]. Another popular visualization of vector fields is through advecting streamlines, i.e. integral curves that are everywhere tangent to the vector field. The user places seed points to cover the domain densely with streamlines. However, using denser streamlines to show greater field strength introduced clutter. More sophisticated 2D seeding strategies were introduced to reduce this clutter [JL01] and extended to 3D [MTHG03].

A recent technique by Günther et al. [GRT13] proposed intelligent opacity optimization based on CPU-computed importance values on a per line segment basis. Despite the high quality of images in their results, the optimization process serializes heavily and the question of whether it is possible to achieve interactive rates with importance-driven opacity optimization remains an open one. The calculation of importance values constitutes an extra preprocessing step that hinders the ability to run the algorithm in real-time.

Ma et al. [MWSJ13] proposed the use of information visualization tools to aid the visualization of scientific 3D flow fields. Their graph-based approach provides the user with a high level view of relationships among pathline clusters and spatio-temporal regions, both represented as graph nodes. The main disadvantage is the poor scalability of their approach both in terms of computation and storage space due to the fact that their selection process relies on the construction of two hierarchical data structures in which field data is organized.

We present a novel querying and selection technique for the ad-hoc exploration of dense flow fields. Our goal is to allow the user to create a geometric shape that acts as a "probe" for the field, displaying only the information that flows through a region covered by the probe. The main contribution of our technique is that it does not require the construction of expensive hierarchical data structures for pathlines representing the flow. Unlike other techniques, ours operates directly on pathlines' vertices and requires no knowledge of line features. We implemented our technique using GPU-parallelism and new features in OpenGL 4.4 shaders to achieve interactive frame rates.

Our work is inspired by the work proposed by Ma et al. [MWSJ13] which led us to realize that interactive navigation and filtering through a pathline set can be of great benefit to the exploration and ROI identification process. The main challenge was to do so interactively without much pre-processing of the data, while keeping memory usage manageable for a GPU to operate on. We present "StreamProbe", our proposed GPU-parallel algorithm that makes use of advances in recent GPU hardware and Graphics libraries to avoid the need for data preprocessing and expensive data structure construction.

The proposed novel streamline/pathline visualization and interaction technique makes use of the geometry shader stage of OpenGL's shading language, in order to address the problem of filtering and clutter reduction as one of collision detection. The user can select from a variety of geometric shapes (e.g. a circle, a cylinder, or a box) to define a probe for the flow field. As the user drags the probe around, every trajectory the probe intersects is visible while all others are culled out.

Figure 6.7 shows sample pathline selections given two different probes centered at the location of the 2011 Puyehue-Cordón Caulle volcanic eruption [IEE14]. The circular probe on the left has a radius of three degrees in geographic coordinates. The box probe on the right allows the user to specify an altitude range for the polylines of interest, ultimately providing a 3D selection tool.

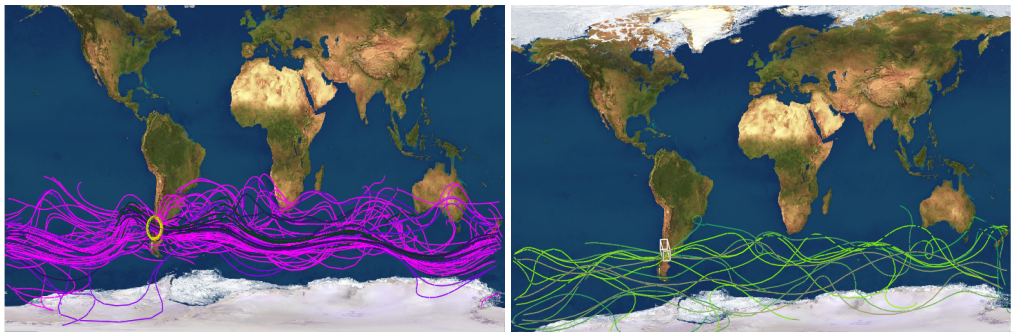


Figure 6.7: Circular probe (left) and box probe (right) used to filter subsets of The Chemical Lagrangian Model of the Stratosphere (CLaMS).

The intersection algorithm can be implemented by a simple intersection test in the geometry shader, much like tests used for collision detection in game development. However, the geometry shader only has access to a single geometric primitive (a line segment in the case of line strips) or a maximum of two primitives (in the case of line adjacency), so the above approach will only make visible line segments under the probe. We use new features in OpenGL 4.x that allow us to create a Shader Storage Buffer Object (SSBO) in GPU memory that can be used for general read and write access in shaders; at which visibility flags can be set for the complete line strip geometry.

Figure 6.8 shows how the created SSBO is accessed by both the application and geometry shader sides. Synchronization points implemented as memory barriers are inserted on both sides to ensure that correct data is being accessed by the different shader invocations. The flags array is sized according to the number of pathlines in a given dataset and contains a set of boolean switches, one for every polyline. The algorithm is designed to perform intersection tests in parallel by every geometry shader invocation on each line segment independently and display complete polylines for which any line segment passes the test. Once a line segment intersects the probe, it will set the flag for the corresponding polyline in the flags array to “ON” which means that this pathline should be visible in the final render. A synchronization point is executed to ensure that all threads have finished their intersection tests and the switches for the visible pathlines have been set.

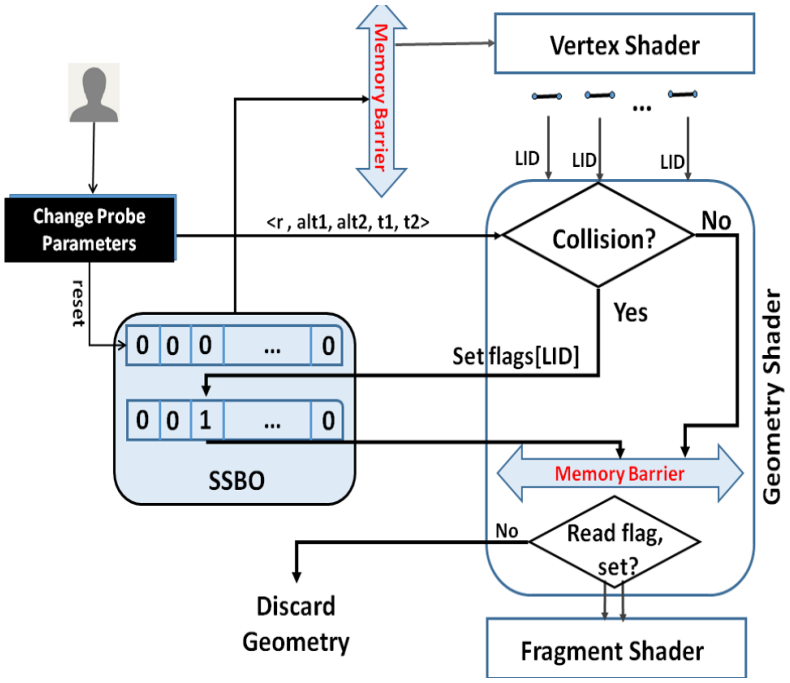


Figure 6.8: Workflow of StreamProbe algorithm.

Note that there is no race condition here as geometric primitives that fail the test will not write anything to the shared flags array. After synchronization, each primitive will go ahead and read the flag value for its corresponding polyline, if the value is ON then geometry for this thread's vertices are emitted by the geometry shader, otherwise the primitive is culled. The use of a memory barrier in our geometry shader is well justified. Due to the highly parallel nature of OpenGL, several rendering iterations can overlap in time so some geometry shader invocations can be working on iteration i for example while others would be processing primitives from iteration $i-1$. This initially resulted in a flicker effect in our render. To overcome this problem, we let the application reset all flags whenever probe parameters are changed by the user, after which a second synchronization point was inserted on the application side.

The usability of StreamProbe in the exploration phase has proven useful for domain experts to reduce clutter, understand patterns, and gain more insight into the dynamics of the flow. Example event-specific questions that domain experts were interested in answering are:

1. Can we single out all detected points that can be attributed to an event?

2. How are data points emerging from an eruption event expected to behave?

Figure 6.9 shows how the system enables the correlation of different data points to a particular event. This correlation is made possible by placing the probe at the eruption event and adding a time query to the probe to only select trajectories that pass through the probe during a time frame that is specified by the domain expert based on their knowledge of the duration of eruption from external news sources. Only MIPAS detections that map to selected trajectories are displayed over time to observe how the resulting plume is expected to behave, hence, answering the second question.

6.2.3 ROI-Specific Questions

Instead of focusing on an event, domain users may be interested in a specific region of interest. For example, interest may be on studying how a particular country or region is affected by the ash plume that results from an eruption event. Questions that need to be answered are like:

1. Is the ROI affected by that eruption event?

2. If yes, when does the plume reach that ROI?

3. For how long is the ROI affected?

4. How is the flow expected to behave, over time, inside and outside of the ROI?

To answer this type of questions, we introduce the idea of combining multiple probes. In the case of volcanic eruptions, we need only two probes: one that selects the time and location surrounding the eruption and one that covers the ROI that we would like to study.

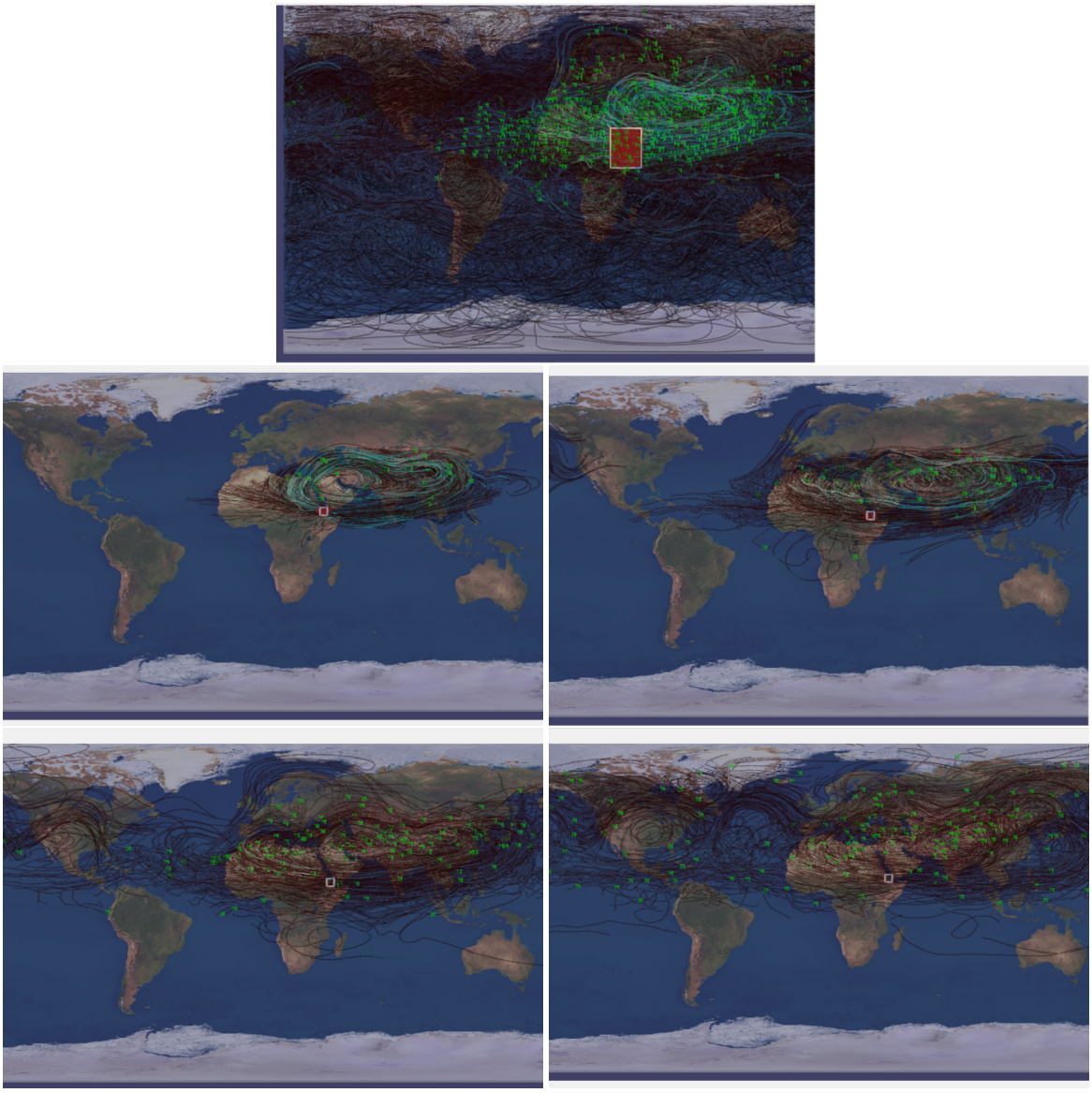


Figure 6.9: **Top:** CLaMS simulation trajectories (black) and MIPAS ash detections (light green) in an area around the Nabro volcanic eruption. **Middle and Bottom:** data that can be attributed to the Nabro eruption event are selected and visualized over time.

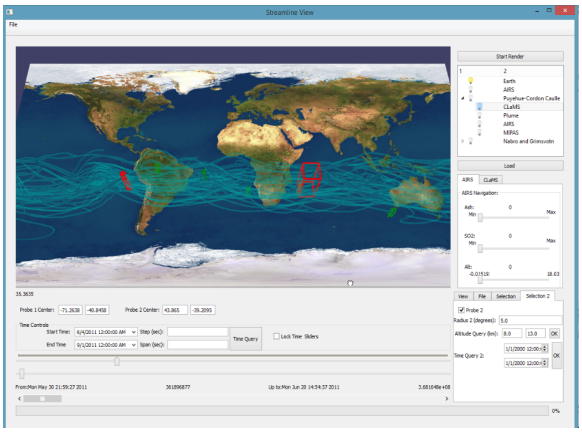


Figure 6.10: ROI query using 2 probes: one that creates a query for the time and location of a source, and the other probe just selects a random ROI to see how it is expected to be affected by trajectories flowing from the source.

Another way of filtering data and zooming in on locations of interest is through projecting one dataset to a $2D$ plane (annulling its Z-dimension if it is available) and using this dataset's planar image as a culling plane for another dataset. The advantage of using this technique, in addition to clutter reduction is that it helps correlate the parts of a dataset matching regions of interest from another. Figures 6.11 and 6.12 depict an example of this Z-culling technique based on AIRS dataset. AIRS data is only available in $2D$. It has higher spatial resolution in the XY plane. Therefore, sliding AIRS images through a volumetric dataset with lower spatial resolution can give the visual effect of filling in the spatial gaps in regions of interest in the three-dimensional set; while a higher temporal resolution datset can be used to bridge the temporal gaps in AIRS data, as Figure 6.11 depicts (see flow in the Southern Hemisphere). Augmenting one $3D$ dataset with a $2D$ culling plane representing another in this manner enables both sets to complement one another and provides a visualization that reduces the shortcomings of both.

In addition to the filtering and selection techniques we described so far, camera interactions including rotation, translation, and zoom help the user gain perspective in the $3D$ model, while reducing some of the clutter caused by the large amount of data available. Figure 6.13 shows a

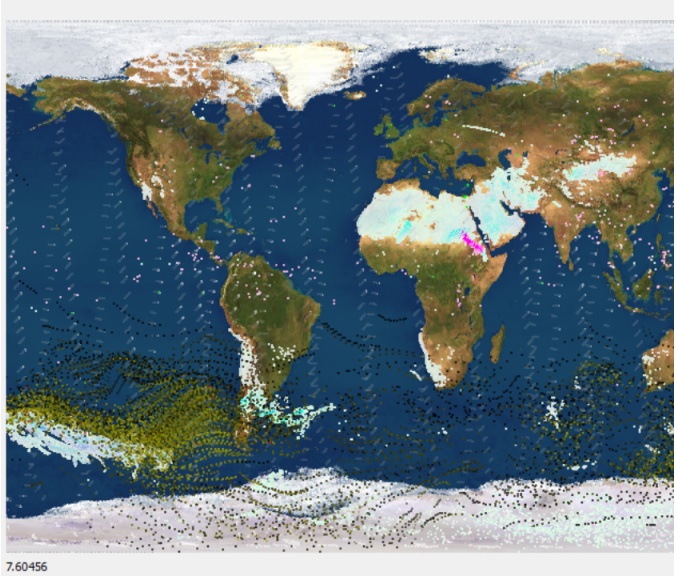


Figure 6.11: View includes three datasets (AIRS, CLaMS, and MIPAS). AIRS data (cyan and white) are used as culling plane for CLAMS simulation trajectory points (dark green). In the Southern Hemisphere, simulation data helps identify the altitude range in which AIRS detections should exist and how they are expected to flow over time from one region to the next. AIRS data gives a better understanding of the plume shape.

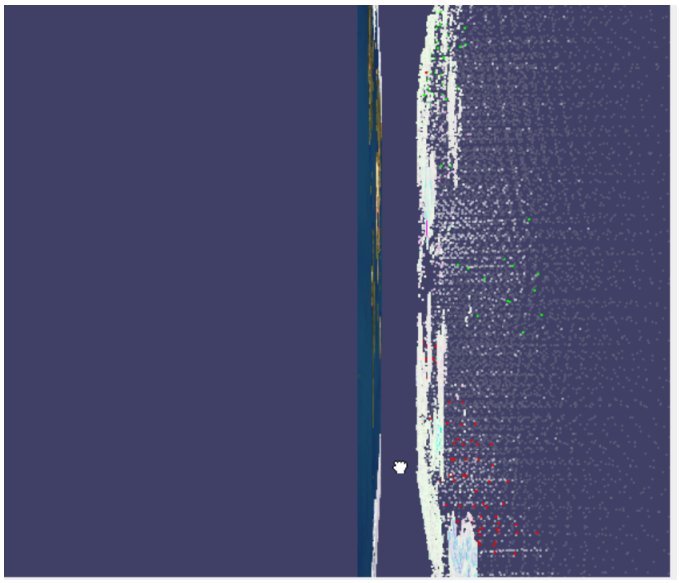


Figure 6.12: Side View: AIRS data used as a culling plane for MIPAS data. Again, MiPAS detections can help identify altitude range while AIRS data give a better understanding of plume shape.

slightly rotated data model using perspective projection, displaying two interleaved plumes, one from CLaMS simulation trajectories (blue) along with a synthetic plume (white) generated by interpolating a subset of the simulation trajectories at regular time intervals.

6.3 Details on Demand

Upon user request, details on a specific region or point of interest should be available. Cursor location in world coordinates can optionally be displayed, to accommodate for the need to obtain more detailed information about a specific dataset location as seen in Figure 6.14. Cursor location is readily obtainable in viewport coordinates from standard mouse events. To display it in world coordinates, we adopt the method of feeding cursor location in viewport coordinates back to the OpenGL's rendering pipeline, only letting them traverse the pipeline in reverse order. Through applying inverse transformations, we calculate the position to which the cursor points to in terms

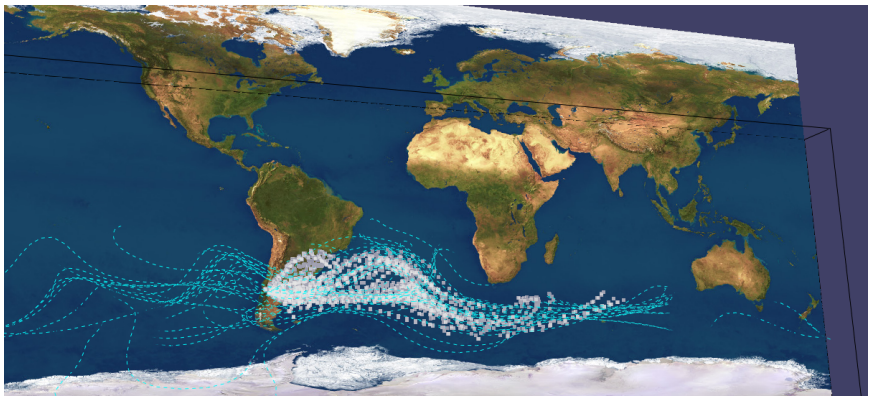


Figure 6.13: Rotated Model to gain perspective.

of data coordinates, which enables the selection of specific points in the data space.

Our system enables a cross-dataset selection method that allows the user to select one point detection obtained from the MiPAS instrument, and using geometry shader processing, the system highlights the entire pathline from a simulation dataset (CLaMS) representing the trajectory of that detection through time. This process uses a Shader Storage Buffer Object (SSBO) similar to the one used in StreamProbe technique for flag setting. Example questions that can be answered for details on demand are:

1. Is a specific detection point caused by or emerging from a certain eruption event?
2. What is the expected location of a particle of interest at a specific time?

The concept of brushing and linking is prevalent in mainstream visualization tools, especially ones with multiple coordinated views. The user selects a data point or subset of points in one view and the selection is highlighted in all other views. In those scenarios, a single dataset is visualized in different views and so the linking process becomes a matter of fetching data records corresponding to a selection and giving them a different color in all available views. We advocate

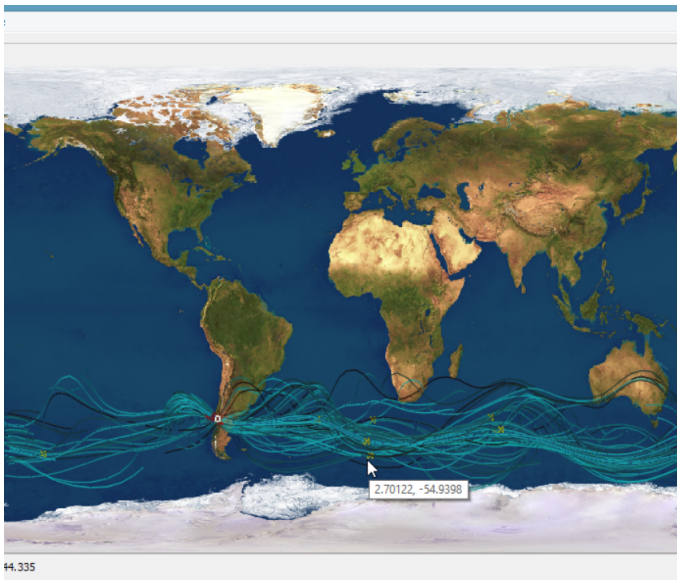


Figure 6.14: Cursor location calculated and displayed in world coordinates by traversing OpenGL's transform pipeline in reverse.

another concept here. Particularly, we advocate single view visualization for multiple datasets (opposite to single set multiple views).

To account for "Details on demand" visualization tasks in our exploratory visualization system, we developed a selection technique that allows the user to use one dataset to select data from another. More specifically, to provide the ability to track a specific point detection back to its origin and visualize its progression through time, we shoot a ray from cursor location through the data world. Points in the MiPAS dataset that have corresponding trajectories in ClaMS data are tested, in parallel, for proximity. For data points found within a certain proximity around the ray, the geometry shader sets a distance measure in shader's global memory. Once a point with minimum distance is found, the shader sets a flag value, again in shader global memory, for that point's corresponding trajectory and a memory barrier is executed. After which every point in the trajectory set retrieves the flag to test whether or not it belongs to the selected trajectory and updates its color and opacity accordingly. Figure 6.16 shows a screenshot of the selection process.

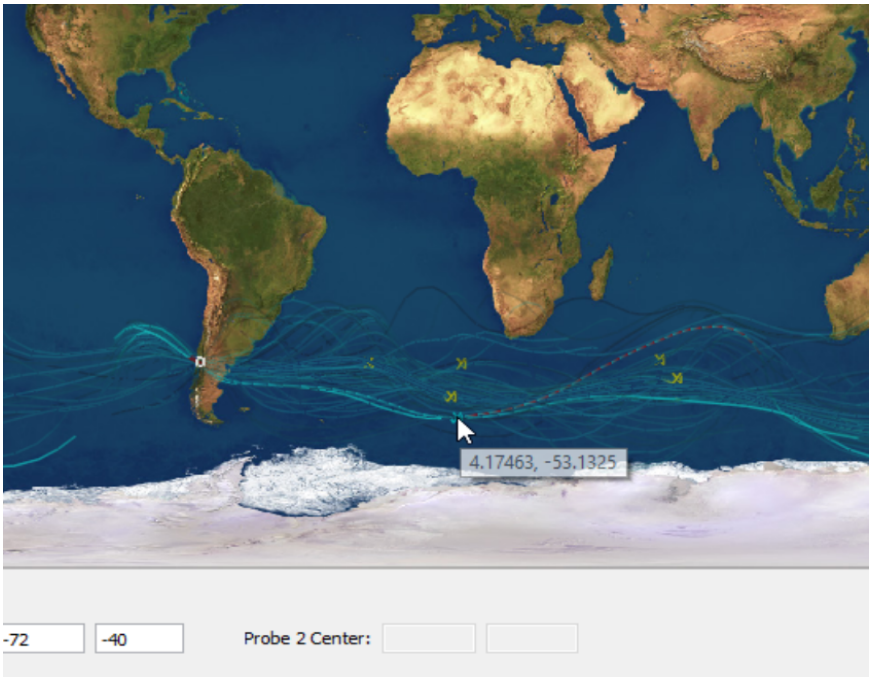


Figure 6.15: Attributing a detection point to an eruption event.

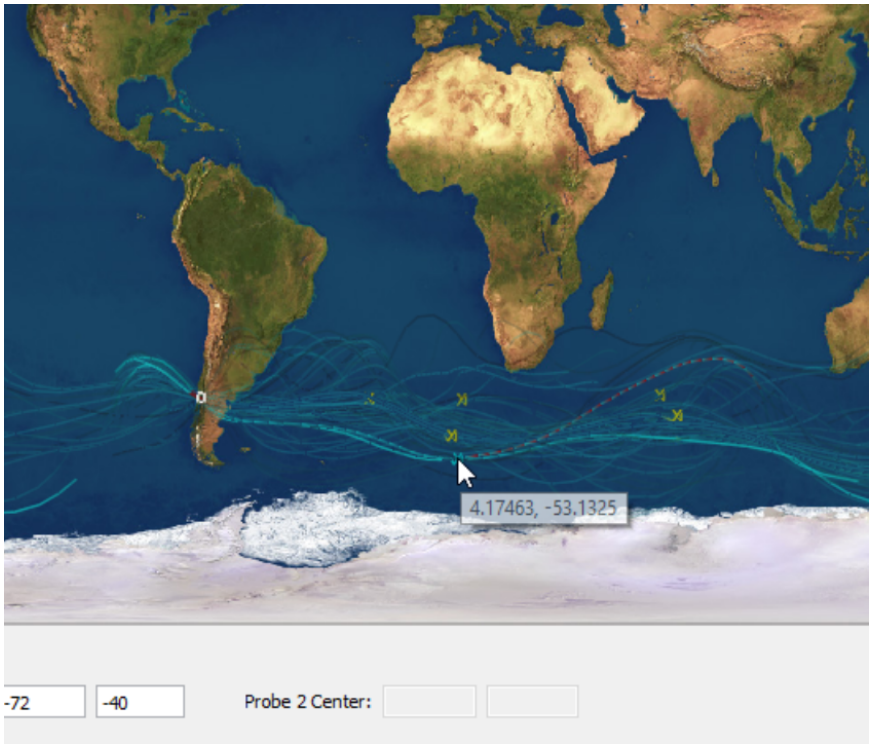


Figure 6.16: Point Selection technique: Pathline highlighted in red and aqua pattern represents the trajectory of the selected detection through time. Nearby trajectories are displayed in full opacity while other trajectories are given lower opacity to provide context.

Chapter 7

Data Tracking Subsystem

"In my bones I feel we are closing in on our target, and so I am studying more about Ancient Egyptian customs for burial of the dead."

Egyptology. *Emily Sands. Aswan, Egypt. 1926.*

We addressed the real-time tracking problem with two different solution approaches. Both approaches were applied to the problem of tracking ash satellite detections, following volcanic eruptions, by advecting the detections in simulated flow interactively to correct the positions of the detected particles and obtain an accurate representation of their spatio-temporal behavior. The goal is to allow the expert to scroll through time using a time slider and obtain a full instantaneous image of all satellite detections that were captured within a time frame around the selected time, and be able to visualize these detections at their corrected locations. To enable such interactive results, GPU-based fusion and tracking techniques were implemented in OpenGL's vertex shader, to which the detection points were fed as vertex arrays for massively parallel tracking. Our experiments have shown that fairly accurate tracking results are obtainable using the proposed

spatio-temporal interpolation method, and that interactive tracking is possible with frame rates varying in direct proportion to the number of particles that are being tracked in parallel.

In the first approach, we proposed a reference model creation process [EGG+15] that samples the space-time domain at known locations, given human expertise and understanding of the origins of the studied phenomena. The generated sample points are advected through space and time by interpolating simulation data to create a synthetic plume, describing the expected behavior of the samples and covering their predicted motion paths as densely as possible. Visualizing this reference model reveals that it does not exactly mimic the ground truth but rather provides a dense enough prediction of motion pathlines. Figure 7.1 displays the constructed synthetic plume (yellow), along with raw satellite detections (red). The constructed plume interpolates between the patterns detected by the satellite and provides a denser coverage of the ROI within 12 hour intervals on June 7th (left) and June 8th (right) of 2011. Both the captured and constructed patterns are overlaid on top of readings from a geostationary satellite that we used for comparison with our tracking results. The details of this comparison will be described in Chapter 10.

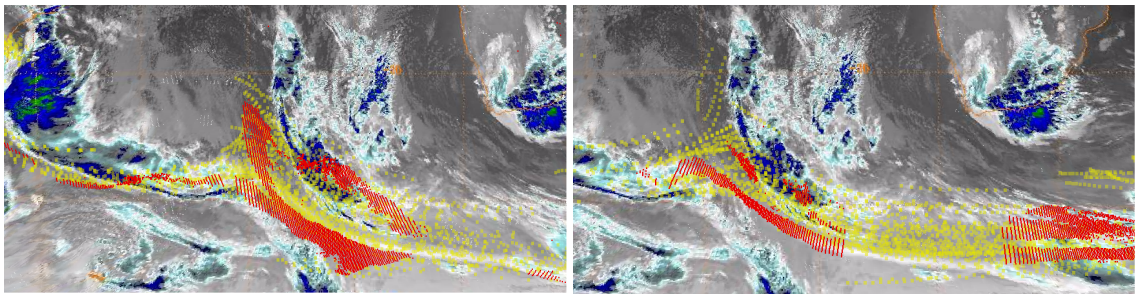


Figure 7.1: Reference model plume (yellow) fills the spatio-temporal gaps for the satellite readings captured within 12 hours on June 7th, 2011 (left) and June 8th, 2011 (right).

So how do we create this reference model? As described in [EGG+15], the problem of reconstructing missing data from sensor measurements can be theoretically insoluble, just like the problem of obtaining a 3D scene from a 2D image. However, in the case of a 2D image we use

certain assumptions about, for example, the size of people, the shades, etc. to interpret the scene. These assumptions are inferred from the viewer's prior knowledge of what the world is like in real life. We apply the same concept to scientific data.

The main assumption we make is that expert knowledge can be combined with the available datasets to create a reference model that roughly describes the expected data behavior. We adopt a reference set construction technique that samples the space at regular time steps within an extracted region of interest (ROI). At each time step, a fixed number of sample points are generated and advected through time by using an interpolation scheme based on simulation trajectories. The technique consists of two modules, the along trajectory adjustment (ADJUSTT) and the spatio-temporal interpolation (SPATIOT). For every generated sample, motion information is obtained from the nearest neighboring points in the simulation dataset by following two steps:

1. The neighboring points are adjusted in time and space to match the time of sample generation,

2. Spatial interpolation is used to obtain translation information to the un-gauged location under consideration based on its relative location to the nearest simulation points.

The technique starts by pre-calculating coefficients for a polynomial regression of a second degree at each simulation point, using the least squares method [LH74]. Regression uses time difference as predictor to calculate the spatial shift as a response. ADJUSTT uses the regression coefficients to calculate the location for each of the neighboring points constituting the nearest simulated neighborhood N_{X_i} to a generated sample point X_i at time t_i (Figure 7.2*a*).

SPATIOT determines the motion vector during a certain period at any un-gauged location (i.e., not available in the simulation data) by interpolating the corresponding shifts of the neighborhood N_{X_i} to the un-gauged location. The spatial interpolation implemented is based on the inverse

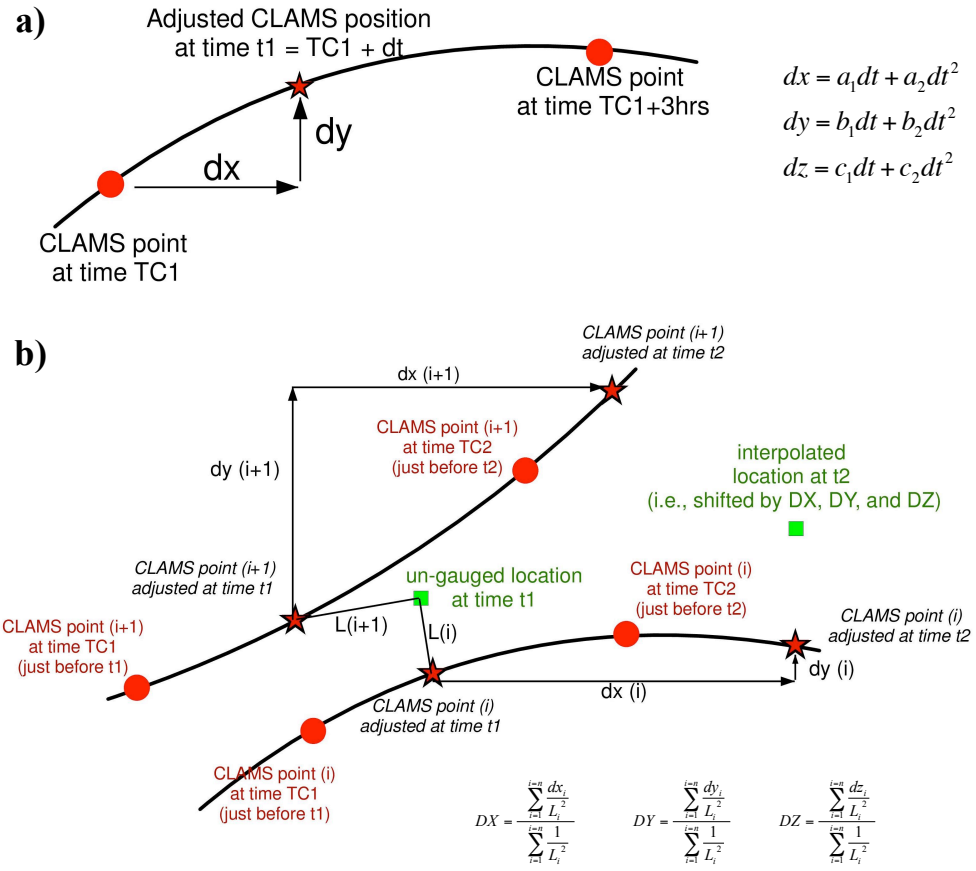


Figure 7.2: Data processing: CPU-based spatio-temporal interpolation technique.

weighted principle [She68] in which closer points are given higher weights (Figure 7.2*b*). Neighborhood size affects the accuracy of the prediction. Considering more neighboring points yields more accurate interpolation results. However, no relevant gain in accuracy was observed with neighborhood sizes of more than 8 points. We found a neighborhood size of 4–16 points to be locally descriptive of the flow field. Full evaluation of different methods of interpolation with varying neighborhood sizes is an interesting topic for future research.

The resulting dataset is a customizable time-regular model grid that can be accessed efficiently on GPU for nearest neighbor search and interactive measurement tracking. The developed technique accepts user input to determine sample generation locations (e.g., based on a known ROI),

time frequency of sample generation (e.g., every hour, day, etc.) and the number of points generated at each time step.

We then proposed a GPU-based approach that uses massive parallelism to project real-world measurements onto the constructed model pathlines, effectively enabling interactive tracking that gives the user the ability to navigate through time while inspecting every detection point's location at any given time instant. This supports visual analytics tasks that help experts decide whether adjustments need to be made to the simulation to enhance their model based on real world measured phenomena.

The first proposed technique created the reference model in a pre-process using the above mentioned temporal adjustment module and a spatial interpolation module. The generation of this reference model took a long time on the CPU (order of minutes) to create a dense enough representation of the spatio-temporal region of interest. This hinders our ability to use it in a real-time application setting. Further, the accuracy of the proposed tracking relied on that of the interpolated model, which was of slightly lower quality, in terms of accurate tracking, than the original simulation pathlines. However, the generated model offered the advantage of regularity in time and that the number of pathline particles in each time step was predetermined based on the sampling strategy defined by the user. To enhance the accuracy and enable real-time tracking, we proposed a second GPU-based method that makes use of the same modules (time adjustment and spatio-temporal interpolation) but performs these calculations directly on the GPU. The main challenge was to accelerate the neighborhood extraction process on the GPU while also enabling efficient access to particle positions at arbitrary time steps.

The proposed novel GPU-parallel spatio-temporal interpolation technique aims to reconstruct motion field information at any un-gauged location (i.e., not given in the original trajectories obtained from simulation) in real-time. A seeding set is fed to the vertex shader in a vertex array. All particles in this set are tracked in parallel and rendered at their updated positions after being ad-

vected through the flow to a destination time; one that is interactively specified by the user through a time slider. Algorithm 1 summarizes the tracking process as executed by the vertex shader on the GPU.

Algorithm 1 Vertex shader algorithm for point tracking.

INPUT: $\mathbf{v} = (v_x, v_y, v_z, T_s)$ query vertex

INPUT: T_d user selected destination time

OUTPUT: $\mathbf{ipv}$ interpolated position vector

T_1 = round_to_nearest_simulation_step(T_s);

T_2 = round_to_nearest_simulation_step(T_d);

TIME_FRAME = define_tracking_time_frame(T_d);

if $T_s \in TIME\_FRAME$ **then**

NN_{source} = traverse_kd_tree(T_1);

NN_{dest} = fetch(NN_{source}, T_2);

for $\mathbf{p}_s \in NN_{source} \mid \exists_{=1}\mathbf{p}_d \in NN_{dest}$ **do**

$\mathbf{p\prime}_s$ = ADJUSTT($\mathbf{p}_s, T_1, T_s$);

$\mathbf{p\prime}_d$ = ADJUSTT($\mathbf{p}_d, T_2, T_d$);

$L^2 = \|(v_x, v_y, v_z) - \mathbf{p\prime}_s\|$;

$\mathbf{w} = \mathbf{w} + \|\mathbf{p\prime}_d - \mathbf{p\prime}_s\|/L^2$;

$\mathbf{h} = \mathbf{h} + 1/L^2$;

end for

$\mathbf{D} = \mathbf{w}/\mathbf{h}$;

$\mathbf{ipv} = (v_x, v_y, v_z) + \mathbf{D}$;

end if

The vertex shader accepts a vertex array containing, for each particle to be tracked, a vector $\mathbf{v}$ for the initial particle position (v_x, v_y, v_z) and the time stamp T_s of this incoming particle. The target destination time T_d as selected by the user is passed as a shader uniform value. The output per vertex is an updated position that tracks the given particle to time T_d. The newly calculated

position is stored in the interpolated position vector $(\mathbf{ipv})$ that is then passed to the fragment shader for rendering.

The algorithm starts by rounding the source time stamp of the incoming vertex T_s to the nearest simulation time step T_1. Next, each vertex is tested to see if it falls inside a user-specified tracking time frame $TIME\_FRAME$. The specified time frame reflects how long a particle is expected to exist in the flow, and therefore only particles that fall within a certain time window around the target time are tracked and rendered. Particles whose time stamps fall outside of $TIME\_FRAME$ are discarded by the shader and not rendered. This tracking time frame is a data dependent parameter that can be set by the user. For particles falling within $TIME\_FRAME$, the advection places them in an updated location by interpolating their local neighborhood's motion information.

We describe in Section 7.1 the spatio-temporal interpolation that performs this advection. To lower the time complexity of neighborhood lookups, we construct a forest of kd-trees for acceleration, which we describe in Section 7.2.

7.1 Spatio-temporal Interpolation

Given a query particle $\mathbf{v}$ that represents a seed being injected in a dynamic flow field, tracking calculations can be performed through interpolating motion information of the neighboring particle trajectories. The spatio-temporal interpolation technique we describe here is the same as the one we proposed in our previous work in [EGG$^+$15]. In which, we used this technique offline in a CPU-based preprocessing step to create a reference model that represents the expected behavior of flow phenomena. The generated set was stored in a time regular plume model, that was then passed to the GPU at a later stage for the advection of satellite data. In this novel approach, we drop the reference model creation step and perform the entire spatio-temporal interpolation in real-time on the GPU using spatial indexing and optimized neighborhood search in the unstructured simulation

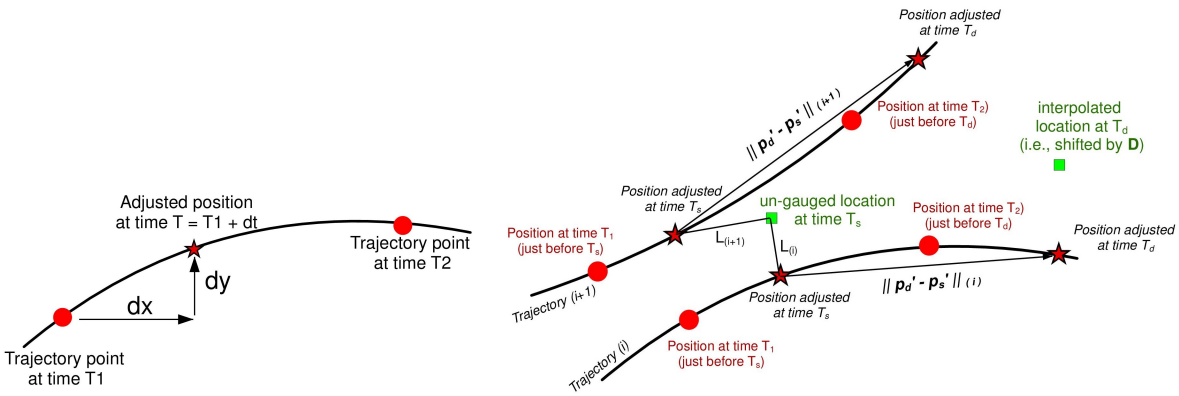


Figure 7.3: (left) Temporal Adjustment of neighboring simulation points to match the time stamp of the query point. (right) Spatio-temporal interpolation and advection of tracked particles along the un-gauged locations.

dataset. Again, the technique has two components: (i) along trajectory adjustment (ADJUSTT), and (ii) spatiotemporal interpolation (SPATIOT) (see Figure 7.3).

In this proposed interpolation technique, each pathline is viewed as the locus of motion of a particle at the different times on a time scale. The objective of the temporal adjustment module (ADJUSTT) is to determine the position along a given trajectory at any time instant within the time step. This is achieved through relative second degree polynomial fitting. Consider a trajectory where the three dimensional vector $\mathbf{p_1}$ denotes a position at time T_1 on a trajectory (Figure 7.3 left). With a simulation step size of ΔT the position after a certain time $dt < \Delta T$ on the same trajectory (i.e. at time $T_1 + dt$) can be obtained by adding the translation:

$$\mathbf{p\prime}_1 = \mathbf{p}_1 + \mathbf{d} \tag{7.1}$$

where **d** is the spatial displacement vector obtained from Equation 7.2.

$$\mathbf{d} = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \\ c_1 & c_2 \end{bmatrix} \times \begin{bmatrix} dt \\ dt^2 \end{bmatrix} \tag{7.2}$$

The coefficients $a_1, a_2, b_1, b_2, c_1, c_2$ of a second degree polynomial can be calculated during pre-processing using the least squares regression method [LH74] and added as attributes to the original flow particles.

The spatio-temporal interpolation module (SPATIOT) determines the translation during a certain period of any point of interest (e.g., sensor detection or a seeding point not originally available in the simulation) by interpolating the corresponding shifts of the nearest trajectories' points (in time and space). Spatial interpolation is based on the inverse distance weighted (IDW) principle [She68] in which closer points are given higher weights. In order to formulate the problem, consider a point of interest at source time T_s for which we need to determine the corresponding position at destination time T_d. The trajectory simulation dataset contains N_p polylines and total number of vertices N on a time scale where T_1 and T_2 are the nearest time steps to T_s and T_d, respectively. The interpolation proceeds as follows:

1. Select all points at time T_1 from the simulation. The number of selected points is equal to or less than the number of polylines N_p in the simulation data set.

2. Select the k nearest neighbors from the simulation step T_1 (recommended k =4-16 points).

3. Use ADJUSTT to slide the selected neighbors from their positions $\mathbf{p}_s$ at time T_1 to their corresponding adjusted positions $\mathbf{p\prime}_s$ at time T_s, at which the particle $\mathbf{v}$ was injected in the flow.

4. Slide the k nearest neighbors along their corresponding trajectories to fetch their positions at time step T_2, then use ADJUSTT again to slide these updated positions $\mathbf{p}_d$ from time T_2 to their corresponding positions $\mathbf{p\prime}_d$ at T_d.

5. Calculate the spatial shift vector $\mathbf{D}$ as the IDW average of the distance between the adjusted positions at time T_s and the corresponding positions at time T_d (Figure 7.3 right).

$$\mathbf{D} = \sum_{i=1}^{k} \frac{\|\mathbf{p\prime}_d - \mathbf{p\prime}_s\|_i}{L_i^2} \Big/ \sum_{i=1}^{k} \frac{1}{L_i^2} \tag{7.3}$$

where L_i is defined as the distance between the original vertex position (v_x, v_y, v_z) and the i-th nearest neighbor position at the source time step $\mathbf{p\prime}_s$.

For the sake of clarity in our notation, we note here that $\mathbf{d}$ denotes the displacement vector for along trajectory adjustment within the time step (Equations 7.1 and 7.2), whereas $\mathbf{D}$ denotes the weighted average of the spatial shifts of the k nearest neighbors between the source and destination times T_s and T_d respectively (Equation 7.3).

7.2 Acceleration Data Structures

The particle tracking mechanism, as summarized in Algorithm 1 and explained in Section 7.1, has two major bottlenecks which constitute the bulk of processing time: *(i)* identifying the k-nearest neighbors of a given query point in simulation step T_1, to retrieve a position vector $\mathbf{p}_s$ for each neighbor, and *(ii)* tracking these k neighbors through time to determine whether they exist or not at the destination time step T_2, and to fetch their corresponding positions $\mathbf{p}_d$ at that step. These two steps define two different access patterns to the flow field data residing in the GPU memory. Given the huge number of points typical to dynamic flow field simulation results, linearly searching the dataset in either case is unacceptable and will cause the GPU to timeout. Also keeping multiple copies of the flow data in shader memory is not an option.

We now explain the data structures we used to accelerate these two access patterns. Figure 7.4 depicts the data structures that are kept in Shader Storage Buffer Objects (SSBOs) to optimize memory access, and resolve these bottlenecks.

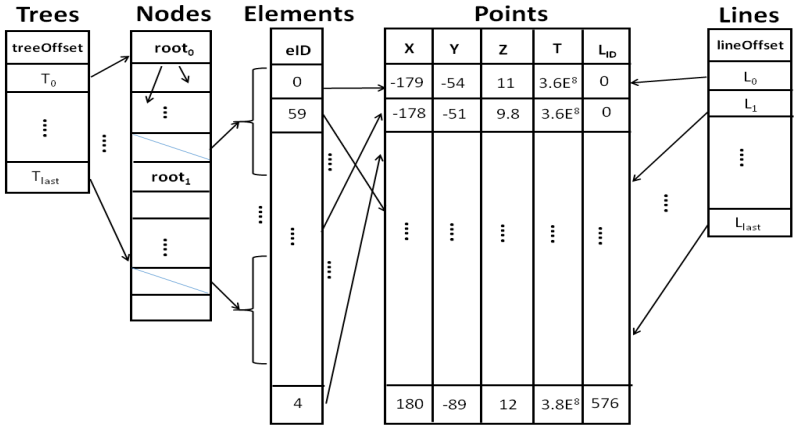


Figure 7.4: Data structures kept in GPU Shader Storage Buffer Objects (SSBOs) to achieve interactive particle tracking.

The first bottleneck is a typical k-Nearest Neighborhood (k-NN) problem. Much research has been proposed in the computer graphics and visualization community to optimize this search on the GPU. Traditional methods in flow visualization have commonly mapped particles to uniform grid buckets for fast spatial range queries. However, grid structures have the disadvantage of poorly adapting to the size and distribution of available particles. As a result, the memory footprint of a regular grid can be prohibitively large, which makes them a bad choice for GPU-based tracking.

Ma et al. [MWSJ14] used a $4D$ quadtree as a spatial indexing structure of dynamic flow fields. Nevertheless, much like grids, a quadtree is better suited for regular volumetric vector fields but does not adapt well for variable particle distributions that are typical in unstructured flow fields. kd-trees [ZHWG08] are well established in the field of ray traversal as highly adaptive spatial indexing data structures that optimize the search for ray-geometry intersection. They have also been adopted in different application domains, especially to solve the k-NN problem [Mou98].

A modification to the traditional binary kd-tree was suggested by Chandler et al. [COJ15] to better tailor it for the problem of particle tracking in dynamic flow. It was argued that, by adding a middle node, flow particles whose circles of influence overlap the splitting plane of the parent

node can be more efficiently retrieved. We have found this method to be problematic in many cases where a number of flow field particles are closely clustered in space. In such cases, the Surface Area Heuristic (SAH), as it seeks to minimize the cost of the split, fails to find a meaningful splitting plane that separates the particles. As a result, the kd-tree construction algorithm can loop indefinitely as it desperately attempts to bring the number of particles down to the maximum allowed number of primitives per leaf node. In our experiments with different datasets, we have found this situation most likely to occur in the middle node of the modified three-child kd-tree, since particles lying on the splitting plane of a parent node are most likely clustered in $3D$ space which leads to the scenario described above. Further, adding a third child node increases memory requirements to store the kd-tree and adds extra branching to the traversal mechanism.

We address these issues by constructing a forest of binary kd-trees using a combination of SAH and median splitting. That is, SAH is deployed as the main node splitting criteria. However, in the cases where SAH fails to find a meaningful split, our algorithm determines a splitting axis as the one with the largest span, sorts particles along this dimension and forces a split plane at the median's position.

Another design decision that contrasts our approach to state-of-the-art techniques is the treatment of time in the hierarchical structure construction. Ma et al. [MWSJ13] treated time as a fourth dimension when building the hierarchy, while Chandler et al. [COJ15] constructed a kd-tree for every time step and stored it separately. Both approaches are not acceptable in the case of GPU-based traversal for k-NN search in arbitrary time steps. Particularly, treating time as a fourth dimension adds the time scale to the cost function of SAH which deems it meaningless, since point distributions are altered by spanning a time range. Further, constructing and storing separate kd-trees requires passing them separately upon demand to the GPU according to the source particle time steps.

Since we perform tracking for thousands of particles from the query set $\mathbf{Q}$, in parallel, each with its own source time stamp, the number of kd-trees that needs to be accessed by the shader can be in the order of hundreds. We need to make use of the temporal coherence in the query set Q and keep as many kd-trees as we can possibly fit in GPU memory to allow the tracking at interactive rates and reduce communication with CPU memory to a minimal level. Since simulation data for all time steps is static, i.e. does not change throughout execution, we construct all the kd-trees for all time steps offline then move them as a forest to the GPU.

Let us now describe the SSBOs used to store flow data and their acceleration structures on the GPU (Figure 7.4):

- **Points:** is a sorted array of structures that stores position, time, and trajectory information for all the points in the dynamic flow field. Particles in this array are sorted by trajectory number. That is, point positions for the first particle are listed from the beginning to the end of this particle's pathline (all time steps), followed by the next particle's pathline, and so on. This simplifies tracking through time but makes spatio-temporal neighborhood extraction unintuitive for the shader.

- **Lines:** is a simple hash table that contains the offset at which data corresponding to each trajectory begins in the **Points** array, effectively making access to any given trajectory at any target time step an $O(1)$ operation.

- **Elements:** is an indexing vector to support the extraction of spatio-temporal neighborhood. Constructing this vector starts with a hashing function that uses time as key to map indices of points that fall within a simulation time step to a single bucket. These buckets are then concatenated in a memory block which makes up the initial ordering of the **Elements** array. Next, the **Elements** array is passed as an initial vector to the kd-forest construction algorithm, along with the number of particles in each time step. As the kd-forest construction proceeds, element indices are reordered within the memory segment of each time step separate from all other steps.

- **Nodes:** the kd-forest construction algorithm builds a kd-tree for every time step. All tree nodes for all time steps are then linearized in this one large **Nodes** array. Tree nodes corresponding to the first time step are followed by those of the second time step and so on. For compact representation, node data are encoded in a 32-bit unsigned data type each.
- **Trees:** this tree offsets array simply keeps track of where the root node for each kd-tree resides in the **Nodes** array.

7.2.1 Optimizing kd-forest Traversal

Traversing the kd-forest on the GPU starts with rounding the incoming vertex's time stamp to the nearest simulation step. The shader finds the root node for this time step's kd-tree from the **Nodes** array by looking up its offset in **Trees**. To optimize traversal, we implemented in the vertex shader the Best Bin First (BBF) traversal technique, as described in [Mou98].

As the vertex travels down the kd-tree, it traverses the child node that is closest to it in space along the splitting dimension, while the other child node is inserted in a priority queue for later traversal. Once a leaf node is reached, the list of k-NNs is updated with the newly visited points in the current leaf. Then the algorithm fetches the next unvisited node from the priority queue, which has the smallest distance to the query vertex. An early termination condition allows tree traversal to end at any point if the node acquired from the priority queue has a distance to the query vertex that is farther than a tolerance radius ϵ around the currently fetched neighborhood. In terms of neighborhood accuracy, setting $\epsilon = 0$ yields the exact same neighborhood results retrieved by a linear search. Larger ϵ values cause the algorithm to terminate earlier and yield approximations of the true neighborhood, which trades accuracy for performance.

Since nodes are dynamically inserted and removed from the priority queue as the tree is being traversed, maintaining the queue can become costly. A minimum heap is used to implement the queue, where the distance to the query vertex is used as key. That is, the root of the heap always

contains the node that is closest to the query vertex. However, maintaining a binary heap for every vertex that is being tracked can use up GPU memory and degrade performance; thereby imposing some limitations on the number of vertices that can be tracked in parallel. Appendix B elaborates more on the effect of the number of particles being tracked on performance measures like the average frames per second. Generally speaking, the size of the heap is determined by the number of nodes in the largest kd-tree of the forest, to accommodate for a worst case scenario in which all nodes of the biggest kd-tree need to be enqueued for traversal. This case does not happen in practice, because as the tree is being traversed only one child of every node is being enqueued while the other is expanded. However, we set the size of the heap using this safety margin to avoid leaks. One final optimization for the kd-tree traversal is the concept of incremental distance calculation that was suggested by Arya et al. [AM93]. We use this method in our shader implementation to reduce the number of floating point operations required for traversal.

Chapter 8

Illustration Subsystem

"J.A. could not resist sketching Khan El-Khalili, the Cairo bazaar that we entered to track down a few final items of equipment."

Egyptology. *Emily Sands. Cairo, Egypt. 1926.*

The purpose of illustrative visualization is summarized by Born et al. [BWF+10] as: *"The simplification of complex contexts, concentration on relevant features, and neglect of details that obstruct understanding."*

The tradeoff between physically-based rendering and illustrative techniques has been an area of active research. Physically based or Photorealistic rendering is sometimes desirable in the gaming and movie industries, to a certain extent. However, even in these applications more realism does not mean better viewer perception. One of the famous examples that illustrate this concept in the field of robotics and humanoid animation is the Uncanny Valley [MMK12], shown in Figure 8.1, which supports the hypothesis that human viewers experience revulsion, fear, and other iterations of distaste when human features look and move almost, but not exactly, like natural human beings.

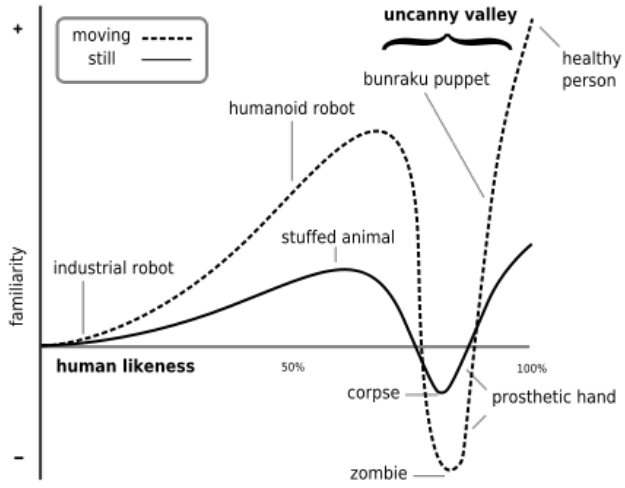


Figure 8.1: Uncanny Valley: viewers experience revulsion, dislike, fear and other iterations of distaste when human features look and move almost, but not exactly, like natural human beings.

In the field of Scientific Visualization, an ongoing debate sheds some light on the advantages and disadvantages of both. Physically-based techniques can support the creation of the cognitive model of the physical phenomenon in question through relating the data to the real event(s) and evolution through time. Global illumination techniques add to the viewers' perception of depth and spatial correlation among the different structures. However, attempting to emulate physical phenomena as they happened does not give enough insight into their dynamics, their inner structures, and does not answer the domain-science questions sought after. In order to reveal more structure and enhance user's perception, clutter reduction and ROI identification techniques aim to reduce the size of data being visualized to reduce the cognitive load of the user and highlight relevant features while disregarding irrelevant details.

Nonphoto-realistic and nonphysically-based techniques are motivated by the fact that all visualization techniques are essentially qualitative. Illustrative visualization techniques range in functionality from ones that merely alter color and opacity based on some attribute values [GRT13] to ones that modify the underlying geometry or create new geometry altogether [RGJ12].

These techniques offer an added value for application areas with time-variant data. This is because the representation of motion using a still image, a sequence of images, or an interactive animation can be more desirable that simple animations in cases where motion information needs to be conveyed in a more persistent and interactive way that does not rely on user's memory the way movie-like animations do. Interaction is a major design goal in time-variant visualization for the purpose of empowering domain experts with the ability to configure the way depiction techniques operate, edit visual and textual appearance of dynamics, and model from knowledge to images.

In our research, we focus on illustrative techniques that offer an extensive range of alternative visual representations of the data. This provides the expert with great flexibility in the choice for adequate visual encodings, which in turn serve the purposes of cognitive model construction, hypothesis generation, and testing. Therefore, our framework ensures direct communication from the exploration and reconstruction subsystems to the illustration subsystem, while also maintaining user control over illustration parameters.

8.1 Illustration for Exploration

As was shown in Figure 5.1, the iterative process of comprehension relies on the search for relations that matches a depicted image to existing knowledge in the user's brain. The ability to find a match in this search is crucial to understanding what the data represents. Therefore, it is important for the user to be able to create a multitude of pictorial representations in order to find that match.

In our system, we use OpenGL's Vertex and Geometry shaders to support the goals of this exploration and cognitive model construction phase and provide interactive illustrative mappings from attribute values (e.g. SO2 and ash indices) to color and opacity visual encodings. The result is a clutter free view of the data that enables domain experts to extract regions of interest matching

their mental model of the depicted phenomena from the data.

An example of this extraction process is shown in Figure 8.2. User-created color encodings of AIRS satellite readings are based on ash and SO2 indices found in the data. Color encoding, despite lack of photorealism, provides a depiction of how the attribute values contributing to the color calculation are distributed across the identified ROI. Next, the user interacts with attribute sliders to control the rendering opacity based on a combination of these values. In this case, opacity encoding yields a visually extracted subset of the data (Figure 8.2 (right)) that matches the expert's mental model of the target ash plume.

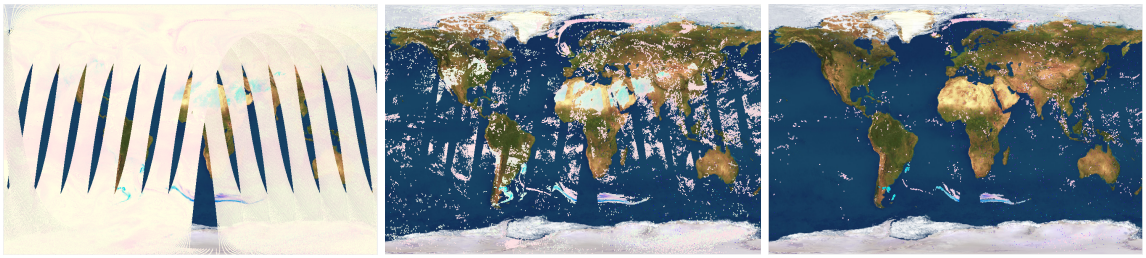


Figure 8.2: Different opacity encodings for AIRS satellite data captured in the first 12 hours of June 8th, 2011.

8.2 Illustration for Information Reconstruction

Depth perception is crucial for the understanding of three-dimensional features and their dynamic behavior. Reconstructing altitude information for satellite data is challenging and requires expert interaction. In order to aid expert perception of altitude information from $3D$ flow simulation, light interaction needs to be taken into consideration. Illuminating streamlines and/or pathlines is an effective way of communicating the three-dimensional structure of the flow and enhancing depth perception [ZPLG13].

Aside from the fact that global illumination calculations are expensive, more challenges arise in the case of simple geometric structures (i.e., lines) that make up flow streamlines or pathlines. Namely, the Phong/Blinn model of light calculation cannot be directly applied to $3D$ lines due to the fact that, unlike surfaces, a line segment does not have a uniquely calculated normal. To overcome this difficulty, several techniques have been proposed in the literature such as *cylindrical averaging* [SM04], *maximum reflection* [MPSS05], and *surface specification* [ZPLG13].

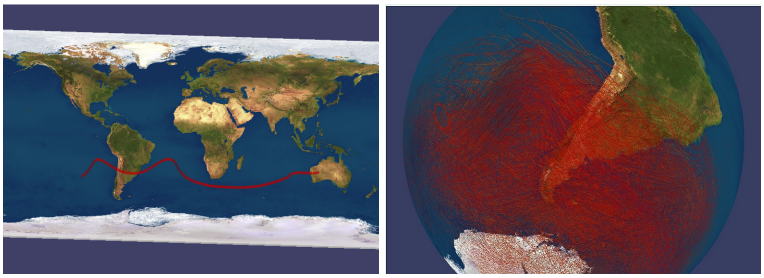


Figure 8.3: Illuminated pathlines to provide better perception of altitude information.

We provide a simple illumination scheme in our system with the possibility of plugging in any of the above mentioned schemes in the rendering program's fragment shader. This extensibility is supported by adding a simple functionality to the geometry shader, which calculates a normal value for each line segment of every streamline in parallel. These calculated normals are then passed to the fragment shader to perform lighting calculation. Figure 8.3 shows examples displaying simulation pathlines with a simple illumination scheme using directional light. The direction of light is passed as a shader uniform and can, therefore, be controlled interactively by the user.

In the remaining part of this dissertation, we present an evaluation of the proposed framework and system to generate accurate visual representations and support user understanding.

Part III

Evaluation and Discussion

System usability is a major component of our research. We have sought to strike a balance between ease of use and sophisticated GPU-based implementation. Interaction can provide a wide range of possibilities that can sometimes be overwhelming to the user. In order to avoid that, interaction and GUI components were designed to be intuitive and of relevance to the domain of choice. Discount usability was a primary tool we used by asking two graduate students to use the system and observing their interaction for early evidence of system usability. Feedback from the students was incorporated in the Graphical User Interface before we asked three domain experts to use the system, perform tasks and answer insight questions based on their interaction. The collected expert feedback is described in detail in Chapter 9.

Chapter 9

System Usability

"I'd like to think scientists need us– but do they? Did Newton need Blake?"

James Merrill.

The proposed framework adopts a human-centered approach, in which the domain expert is a central participant in the data fusion process. This makes the whole process an active problem-solving experience. Consequently, usability is a central consideration throughout the design and implementation cycle of a reference system that implements the framework. We collaborated with domain experts in the application domain from the early stages of our design, and throughout the implementation, testing, and evaluation phases. In this chapter, we reiterate on these collaborations. We describe the resulting usability considerations, and lessons learned.

9.1 Usability Considerations in the Precondition Phase

In the precondition phase of design, several interviews and observation sessions were scheduled with domain experts that aimed at exploring background in existing visualization methods. This led to an early identification of challenges that the domain experts were facing with the tools and techniques they are most familiar with (e.g., Paraview or ArcGIS). These existing methods paved the way for us to introduce new ideas and develop novel techniques for a better interactive user experience. Furthermore, brainstorming discussions helped pinpoint current challenges in existing tools and opportunities for contribution to the visualization literature.

We started by exploring collaboration options first with a team of biomedical engineers to address the problem of multi-volume time-variant data visualization from a neuroscience perspective. Early discussions helped identify shortcomings of the analysis tools that they were using in their quest to investigate human brain activity using Magnetoencephalography (MEG) and fMRI data to combine the high temporal resolution of the former with the high spatial resolution of the latter. There was a need for real-time source reconstruction of electromagnetic brain signals to provide live feedback to a visualization system during user interaction sessions. We designed and implemented a prototype system for the real-time reconstruction of brain signal sources using a GPU-based beamformer technique. Preliminary results showed great opportunities for real-time reconstruction and visualization of a simulated dataset. However, for data fusion purposes, the need for experimental data was crucial to our research, and that data was not available at the time of collaboration.

We have come to realize that the nature of the problem we are addressing in our research was a restricting factor for collaboration opportunities; particularly because of the types of datasets needed. The visual fusion of time-variant multi-sourced data introduces the need for both sensor-collected ground truth measurements as well as simulation datasets that can be tailored or interpolated to create a reference set that fills the gaps in the spatio-temporal domain.

Earth and Atmospheric sciences came as the next natural choice for a target domain. First, because of the growing need for analysis of remote-sensed data to probe the Earth's atmosphere. Second, because of the dire need to fuse data obtained from multiple sources in order to reconstruct missing information and integrate expert's knowledge in analyzing the dynamic behavior of physical phenomena under study.

Early discussions with a domain scientist and the availability of the dataset from the 2014 IEEE Scientific Visualization Contest, as a real-world problem that poses a set of challenges to both visualization researchers and to domain experts, helped the progress in this direction. Despite this, we have maintained a level of task abstraction that is high enough for our framework to be usable and for the proposed techniques to be useful for a wider variety of problems including the previously explored biomedical domain. Data loading/unloading capabilities, system compatibility with ComVis, interactive tunable visual encoding, generic clutter reduction and filtering techniques as well as Z-culling all serve this purpose.

In order to identify the shortcomings of existing tools both from the perspective of the application domain and that of visualization, we scheduled two observation sessions: one with an atmospheric scientist and one with a visualization research scientist. Two days before these observation sessions, we e-mailed the data description to each of the scientists and requested that they select a software that they are most familiar with to explore it. During the sessions, the atmospheric scientist used ArcGIS software and the visualization researcher used Paraview. No particular tasks were given during those preliminary sessions. Rather, each scientist was requested to explore the data at their own pace, then give their own insights on the data, where the shortcomings are, and what they believe would make their analysis less burdening and more accurate.

During data exploration, observations were taken to answer the following:

1. How often do users load/unload different datasets?

2. What would they like to learn about the data?

3. What questions are they not able to answer given the original form of the data?

4. What visual fusion tasks do they attempt?

5. What constitutes a bottleneck for their interaction with the data?

Some of the findings from these observation sessions were straightforward and predictable like integrating multiple datasets in one view, and frequently turning on and off the visibility of individual sets. This ability to turn on/off a data layer is common practice and enables the expert to compare among different viewing strategies and coverage patterns of different datasets.

However, some of the observed tasks were interestingly new, which led to exciting research directions. For example, we noticed that the atmospheric scientist was interested in learning about the temporal evolution of the AIRS data across several days. Using a color encoding scheme, he was able to visualize this information on a per 12 hour basis. The expert created a separate image for every 12 hours worth of AIRS data and then displayed these images as a sequence to understand how the plume evolves. Figure 9.1 depicts part of such sequence generated using ArcGIS.

The next question of interest to the atmospheric scientist was whether it was possible to shift AIRS detections in time to obtain an instantaneous image at different points in time. The main weakness he identified in AIRS data was that the data does not show a full image of all potential ash readings around the Earth at a certain time. Rather, the data only contains scattered readings at different time stamps. Questions that were identified as challenging to answer using existing data alone were: *whether a detection existed at a certain place that is not covered by the satellite readings at a specific time instance*, *whether different parts of the detected plume were actually connected*, and *whether a set of detections can be attributed to a particular eruption event*.

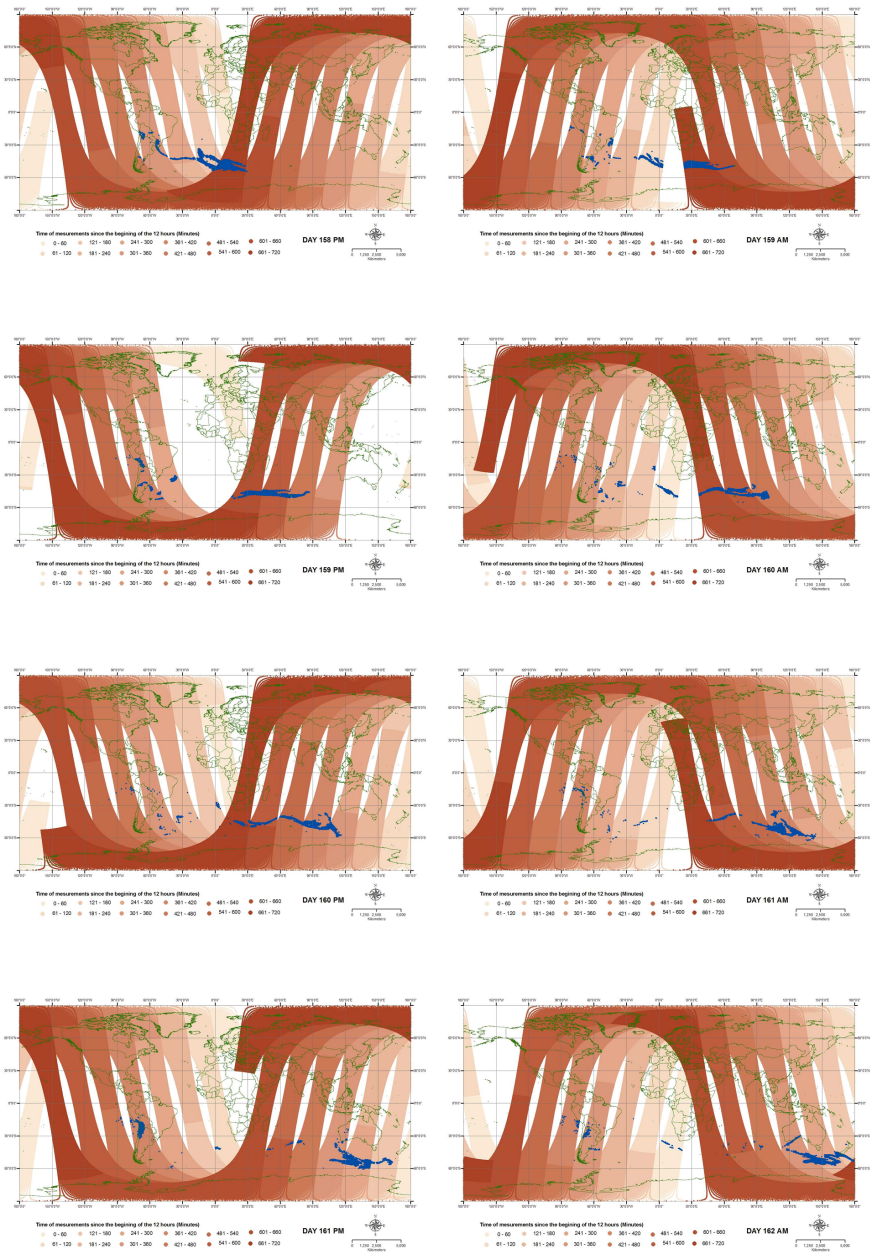


Figure 9.1: Sample 12 hour AIRS coverage on June 10th 2011. Images created with ArcGIS. Morning measurements are displayed on the left while afternoon ones are on the right. Shades of brown encode time while the ash plume is shown in blue.

From our observations, the visualization expert was more interested in combining multiple data modalities together in the visualization (*high-level visual fusion*). Whereas, the atmospheric domain scientist was more interested in exploring each dataset alone and then setting hypotheses about what would be potential methods for the computational fusion of the datasets (*low-level fusion*). However, there was great interest in the ability to understand temporal evolution for different datasets together from both experts. Such temporal evolution was not readily clear from the visualization systems they used. Generating a temporal sequence on ArcGIS took long times and lacked interaction. In ParaView, each AIRS data file (containing twelve hours of coverage) was considered a time step by the software. Filtering different time frames within any such time step was not possible without hard coding database queries. Such query formulation was difficult given that the user, in the exploration phase, is usually not trying to extract data about a specific time period, but rather trying to generate hypotheses about what constitutes spatio-temporal regions of interest. This inspired our design of time sliders that can be either synchronized or moved independently for flexible exploration of multiple time-varying datasets.

Our final observation during the sessions was that clutter was a major challenge faced by the experts that hindered their understanding of possible scenarios for plume evolution. Particularly, both experts were interested in filtering simulation trajectories to animate particles and gain a perspective on trends in the flow around the events and regions of interest. Questions that were identified as being of interest to the experts and that were challenging to answer without adequate clutter reduction included: *Whether a particular set of trajectories can be extracted based on its attribution to a specific eruption event*, whether it was possible to visualize simulated particles that flow between two regions of interest, and *whether this extracted simulated flow information can be integrated with AIRS detections for the sake of interpolation and time correction*.

9.2 Usability Tasks in the Core Phase

During the core design phase, usability considerations learned from the precondition phase greatly informed our framework design and system development. Guided by these considerations along with design requirements we derived in Section 5.1, the system was developed and presented to domain experts to obtain feedback. System usability was evaluated based on the tasks that we introduced in Section 1.3.2.

Three domain experts were requested to use the system and provide feedback. We scheduled individual meetings, about an hour long, with each expert. At the beginning of every meeting, the system was briefly introduced to the expert through a short video tutorial that explained the supported tools and features. Next, the expert was given a questionnaire and left to freely interact with the system to answer the questions. In what follows, we review the tasks and map them to components of the questionnaire that were structured to evaluate the system's ability to support these tasks. Questions included in the questionnaire are listed in Appendix A.

T1: Overview and browsing

The goal of this high level abstract task is to provide systematic support to user's understanding of the data characteristics and the challenges they present. In order to test that understanding, the questions target abilities to:

- **Identify and describe general trends:**
  Questions $Q1$ and $Q4$ ask the expert to choose, in generic terms, a description of the direction and type of flow carrying a feature of interest (the ash plume in this case). Such high level description attempts are a good starting point to set the foundation for a cognitive model describing the expected behavior of the data dynamics. They also help pinpoint shortcomings in the available data (e.g. the simulated flow field) in giving accurate representations. For example, one domain expert noted that it is hard to tell from the CLaMS data alone whether

the ash plume will initially travel toward the North or the East at the time immediately following the eruption.

- **Identify data uncertainties:**

  Questions $Q2$ and $Q3$ are geared toward exposing the uncertainties that are inherent to the available datasets in browsing different regions of interest and detecting events occurring in or around those ROIs. The expert is asked whether specific countries were affected by the ash plume or not and whether they can obtain more information about those events.

In answering these questions, the experts actively construct their cognitive (sub-OAR) model of what happened in the real world from the available visualizations, and from searching external resources for more information (e.g. a world atlas to review country locations). This iterative process corresponds to boxes 6, 10, 11, 12 and 7 in Figure 5.1. As an outcome, the expert comprehends what is available from the data and where it falls short in answering domain questions of interest. For example, two of the experts suggested that AIRS data needs to be interpolated and corrected in time to be able to accurately answer $Q3$.

T2: Estimation of relevant attribute ranges from the different modalities

$Q5$, $Q6$, $Q7$ and $Q8$ address this task. The goal is to strengthen the users cognitive model of the studied phenomena and of their manifestation in the data. Particularly, the exploration phase should enable the expert to *know what to look for* when identifying regions of interest using a combination of attribute filters and visual encodings. In order to test these abilities, the questions target the following:

- **Deduce filtering criteria:**

  The ability to reduce clutter in the visual display of different datasets is crucial in identifying regions of interest. Computational methods to extract such ROIs are used when the expert already has information about attribute ranges of interest to the extraction process. However, in most real-life analysis situations, this is not the case. This where visualization and

interaction play a crucial role in exploratory analysis. The user navigates among different visual encodings and a multitude of filtering criteria. Ideally, visual feedback based on the selected encodings and criteria should be displayed in real time as the user interacts with GUI controls. This real-time feedback generates a large number of depictions of the data to support the identification of objects based on the search for relations to existing knowledge, as outlined by boxes 2, 3, 4, and 5 in Figure 5.1.

- **Reflect on confidence in extracted ROIs:**
  This reflection further reinforces the user's cognitive model by ensuring understanding of the amount of evidence given in the data to support hypothesis generation. This understanding corresponds with box 6 in Figure 5.1. Question $Q6$, and feedback questions $FQ4$ and $FQ5$ assess this reflection.

In answering these questions, the expert generates hypotheses about what constitutes relevant attributes and value ranges of interest for ROI extraction. In addition to identifying ROIs and reducing the dataset size, this ability further strengthens comprehension of the phenomena under study and its manifestation in different data modalities. This understanding then translates into a reference dataset that the user creates by combining the extracted ROIs. Uncertainties about the data support decisions about what should and should not be included in the reference set.

T3: Time-enabled exploration and integration of multiple datasets

The idea here is to enable the user to extend the capabilities of individual datasets in portraying the dynamic behavior of the physical phenomena under study. Through displaying multiple datasets together and animating their progression over time, the system exposes the limitation of each in addressing behavioral questions through time navigation. Questions $Q6$, $Q7$, $FQ4$ and $FQ5$ enable the revealing of temporal gap characteristics, while complementing the users understanding of the ability to answer questions about dynamic behavior based on existing data.

- **Characterize temporal gaps:**
  Knowledge about the size of temporal gaps informs decisions about interpolation strategies that are used to fill those gaps.

- **Predict dynamic behavior:**
  By looking at different datasets evolving over time, the user has the ability to characterize the behavior of each dataset relative to the phenomena under study. This enables the generation of hypotheses about how the next set of measurements, for example, will cover of the spatio-temporal domain and how this newly acquired data can be integrated with other modalities.

T4: Investigation of interactions among different spatio-temporal regions and whether they can together be linked to phenomena of interest

Much of the tasks we have explained so far focus on identifying events or regions of interest but do not tackle interactions among them. $Q7$ is meant to test the user's ability to connect two different regions of interest with two probes and hypothesize on how the feature of interest (volcanic ash plume in this case) would travel from one region to another. This type of interaction analysis is relevant in situations, for example, where decisions need to be made about air traffic within a region of interest, based on whether or not it has the potential of being affected by the eruption event. The system is tested for its ability to support:

- **Generation of Hypotheses on ROI interaction:**
  Two probes are used to filter out flow trajectories that do not connect the regions, in order to remove clutter. At this point, the user is able ot hypothesize about expected interaction between these two regions based on displayed data. However, existing data in its raw format does not allow for the verification of the expected interaction. Therefore, without tracking only the hypothesis generation step of problem solving is possible but not the step of hypothesis testing. The user becomes well aware of this fact during initial exploration, while answering question $Q7$.

T5: Drilling down on specific ROIs and deriving a description of expected behavior

Similar to the interaction between multiple regions, as addressed by task T4, the user can drill down on a single specific ROI, generate hypotheses by visualizing raw datasets, and then test these hypothesis through inspecting reconstructed data within the region. The system is tested for the task of allowing the user to explain what happened in/around a specific ROI. Probe creation to answer Questions $Q1$, $Q2$, $Q3$ and $Q4$ addressed this task.

T6: Obtaining details on a specific region of interest

The system is further tested for its zooming ability to provide details about whether the phenomenon in question exists at a certain time and place, how long it remained there, and what happened next. Questions $Q9$ and $Q10$ specifically target this type of details-on-demand tasks by enabling more specific details to be acquired from the reconstructed (tracked) data. The user becomes aware of the level of detail obtainable through inspection of raw versus reconstructed data.

9.3 Reflections on Usability in the Analysis Phase

The final task specified for our usability evaluation aims at reflecting on the framework design and system implementation in light of their ability to support problem solving steps. Domain experts play a crucial role in this reflection and evaluation phase, as they are requested to answer a series of feedback questions that probe their level of confidence and comfort with the proposed tools. In addition, they are requested to reflect on their own cognitive model of the data through comparing what they initially understood through raw data to what they learned from the reconstructed data.

T7: Determining whether reconstruction results match hypothesized behavior, and modifying hypotheses accordingly

Feedback questions aim at aiding the user to reflect on their answers using raw data alone versus their conclusions from the reconstructed data. The goal is to capture users' confidence when given raw data alone versus when presented with corrected data based on a reference model.

Hypothesis generation and testing can be viewed as two separate components of the human inductive inference process [BG10]. In problems where specific domain questions are known a priori, visualization can be of little benefit to the application domain. In such cases, computational methods can be sufficient to answer those specific questions (e.g. using query languages). However, the challenge of scientific data analysis emerges from the complicated task of generating these specific application questions. Visualization has proven to be very useful for gaining insight [GGZL14] into large and complex data; consequently, empowering domain experts to formulate questions, generate hypotheses, and then test them in an iterative manner. To serve this purpose, interaction is key to make this iteration possible and efficient.

In high level information fusion, supported by Human-Computer Interaction, fusion is an iterative problem solving experience where domain experts comprehend the data, build hypotheses, test them, validate results and iterate back if necessary to obtain higher information content. In the remaining of this Chapter, we discuss how the process of hypothesis generation is greatly supported by interactive visual exploration tasks.

9.3.1 Discussion

Interactive visual exploration of scientific data is a key component in hypothesis generation. Turkay et al. [TLLH13] demonstrate this process in the context of exploring heterogeneous medical datasets. Through an interactive visualization system supporting multiple coordinated views,

they offer visualizations of data items accompanied by visualizations of dimensions to enable exploration of a multidimensional dataset. They note that without interaction, a typical analysis session for this type of heterogeneous multidimensional data, would only involve a subset of the dimensions. However, interaction enables the exploration of the entire dataset.

Our work is inspired by the concepts introduced Turkay et al. [TLLH13] and reinforced by our work with Reiner et al. [SEG+15]. We advocate the crucial role of interaction in building comprehension of the physical phenomenon under study and its use in generating hypotheses about the available datasets. Particularly, we are interested in hypotheses about ways to combine the strengths of available modalities and to overcome their weaknesses.

Examples of user-generated hypotheses from the interaction sessions with our developed system are:

H1 It is possible to identify whether a region of interest (ROI) is affected or not by an event

H2 It is possible to determine a set of query parameters to extract ROIs.

H3 Measured data provide partial truth to the analysis.

H4 Tracking and interpolation can reconstruct information.

H5 A reference data set can be created based on interpolation and tracking parameters that are deduced from exploration.

H6 More accurate information can be reconstructed through tracking.

As the user follows through with the exploration and analysis, they generate these hypotheses and then test them based on interaction with the visualization. This can potentially lead to generating new hypotheses or modifying existing ones. The result of this iterative process is a complete

cognitive model that encodes the user's understanding of the studied phenomena and the datasets representing it.

Chapter 10

Accuracy Analysis

"Men need images. Lacking them they invent idols. Better then to found the images on realities that lead the true seeker to the source."

Proverbs from the Temple of Karnak, Egypt.

Outward-facing validation involves presenting visualization results to the research community, and obtaining feedback. Our collected feedback both from domain experts and from the visualization research community has highlighted the importance of reconstruction accuracy. Domain experts looking at reconstructed data were impressed by the ability of the developed system to perform this reconstruction and rendering in real-time as they scrolled through time. When asked whether the visualized data met their expectations, the answer was positive. However, visual accuracy alone is not enough to establish the validity of the developed techniques. We developed a method for the non-visual computational evaluation of accuracy, which we describe in this Chapter.

We compare our results with raw archived VIS (Visible) and IR (Infra Red) images from geostationary weather satellites at a 3-hour temporal resolution, as obtained from NCDC (National

Climatic Data Center) at NOAA (National Oceanic and Atmospheric Administration). The raw images are from seven geostationary satellites (GOES11, GOES12, GOES13, METEOSAT7, METEOSAT9, FEN YUNG 2E, and MTSAT). For the sake of comparison, we pre-processed the raw images to have a common reference frame with our results. First, the raw images, obtained in image coordinates, are geo-referenced in the corresponding satellite vertical perspective projection. The images are then re-projected in the common coordinate system (i.e., geographic WGS1984 system). Next, a mosaic of the projected images is produced in the common coordinate system to generate one image covering the globe at every time step. Figure 10.1 depicts two sample geo-referenced images from GEOS12 and MTSTAT after re-projection in WGS1984.

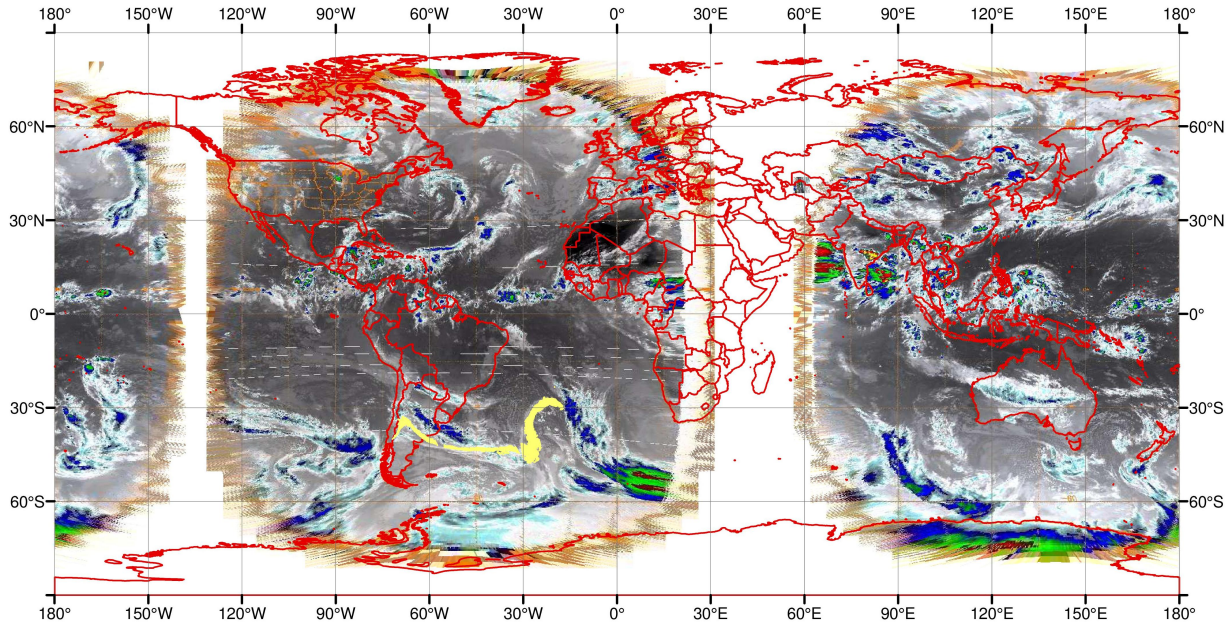


Figure 10.1: Two sample IR images at 20110606114500 UTC after re-projection in the World Geodetic System 1984 (WGS1984) common geographic coordinate system.

The next step in our validation process is to establish a ground truth plume to which the results of our tracking technique can be numerically compared. To this end, we asked a domain expert to manually delineate the plume regions on the pre-processed weather satellite images. Although this is subject to human judgment, it constitutes the best possible definition of the true shape of the

plume. This expert-guided approach is similar to the validation process described in [MVW05] for DTI bundles extraction and clustering evaluation. The delineated plume is then rasterized into a binary image with pixel value of 2 where the plume exists and 1 elsewhere. On the other hand, the time enabled scatter points produced by our techniques (i.e. the synthetic plume and AIRS corrected plume) are also converted to raster format to facilitate the comparison. A focal window (circular neighborhood with radius 0.1 degrees) is used to smooth the point scatters and reproduce our results in raster format. Again, plume pixels are set to a value equal 2 and global non-plume pixels are set to 1.

The accuracies of the synthetic plume and the AIRS corrected plumes, and their ability to emulate the exact shape of the volcanic plume are evaluated by calculating the cross correlation coefficient (as a measure of pattern matching) with the delineated plume on weather satellite images. Figure 10.2 plots the cross correlation over three consecutive days from June 5th to 8th, 2011, during the Puyehue Cordón Caulle eruption. The plumes represented by the curves are: *(a)* the reference plume model created from simulation only (red) *(b)* a tracked AIRS plume that uses the original CLaMS data for neighborhood lookup (black). This plume is constructed using both temporal adjustment and spatiotemporal interpolation (see Figure 7.2 for details). *(c)* and *(d)* represent two tracked AIRS plumes that use the synthetic plume for neighborhood lookup with neighborhood sizes 4 and 16 (green and violet, respectively).

The generated plume model (red curve) has the least accuracy. This is attributed to the fact that the simulated plume is obtained from interpolating CLaMS simulation only. Therefore, errors are progressively accumulated. Our technique updates AIRS corrected plumes every 12 hours with the newly acquired AIRS data (i.e. any errors are reset with every new AIRS data acquisition which is approximately 12 hours apart from the time stamp of every detection). The simulation-based correction has the highest correlation with ground truth. This is attributed to the temporal adjustment step made per detection (Figure 7.2*a*). That is, detections are not rounded to the nearest time step as is done in the reference-model-based tracking. It should be noted that the accuracy analysis performed here constitutes an assessment of both the CLaMS model and the spatiotemporal interpolation. Inaccuracies are attributed to the whole system of calculation including the CLaMS

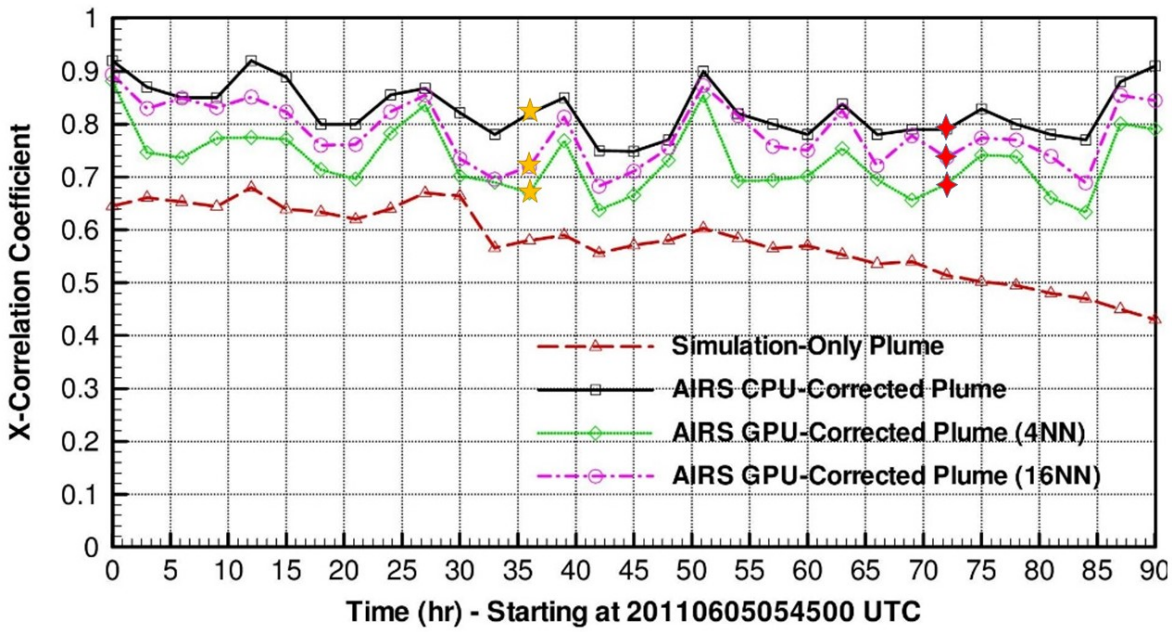


Figure 10.2: Cross correlation over 3 consecutive days between geostationary satellites' delineated plumes and four reconstructed plumes: the simulated model plume (red), CPU-corrected AIRS data using CLaMS for interpolation (black), and GPU-corrected AIRS data using plume for interpolation at neighborhood sizes of 4 (green) and 16 (violet).

model itself.

Since we have placed our results and the geostationary data in a common reference frame, we render both point sets together for visual comparison. Figure 10.3 shows a sample time of the generated plume model (yellow) and the original AIRS data (red) during the first 12 hours on June 8th, 2011. Both sets are displayed on top of IR data captured at 5:54 AM that same day. Differentiating between ash plumes and convective clouds in IR data could be difficult for the untrained eye so feedback from a subject matter expert is essential. Clearly, the plume model fills the gaps in the data but generates more points than is actually needed to represent the true ash plume. Figures 10.4 and 10.5 show two different cases at two sample time steps to compare our tracking results with IR images, Part *a* displays raw AIRS data captured over 12 hours of the day, with no correction. Parts *b*, *c*, and *d* display the same corrected plumes from Figure 10.2.

Sample case 1 (Figure 10.4): tracking results compared at time 5:45PM, June 6th, 2011, with the GEOS-12 IR image at the same time. In this case, the differences among the raw data and the corrected plumes are not significant as the raw AIRS data is captured within a small time interval. However, we note that the pattern shaped like a right-to-left question mark has a visible shift in raw AIRS data (as indicated by yellow arrows), while in the CPU corrected plume and the 16-NN GPU-corrected plume match IR data perfectly. More importantly, This sample case provides us with a visual sense of the calculated correlation coefficients. The cross correlation coefficients between IR delineated plume and the plumes shown in Figures 10.4*b*, *c*, and *d* are 0.82, 0.67, and 0.72 respectively (marked in yellow in Figure 10.2, time 36 hours).

Sample case 2 (Figure 10.5): tracking results are compared at time 5:45AM, June 8th, 2011, with the METEOSTAT-7 data at the same time. The difference between the raw AIRS data and the corrected plumes is significant because the raw data was acquired over a long time span (almost 12 hours time difference between the patterns on the left and the ones on the right in the raw data panel). The corrections in Figures 10.5*b*, *c* and *d* were able to reproduce the patterns in proper locations while filling the gaps. The cross correlation coefficients between IR delineated plume

and the results shown in Figures 10.5*b*, *c*, and *d* are 0.79, 0.68. and 0.73 respectively (marked in red in Figure 10.2, time 72 hours). This case highlights the strength of our tracking technique in detecting locations and times of dangerous corridors in the atmosphere that would otherwise go undetected by raw AIRS data.

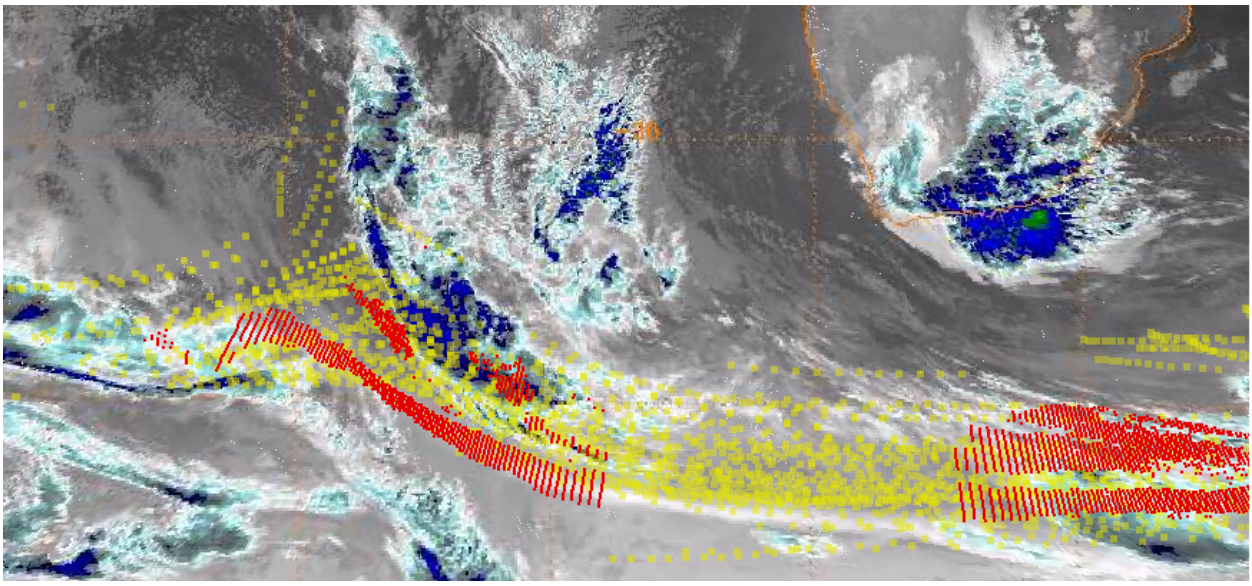


Figure 10.3: Generated plume (yellow) shown with raw AIRS data (red) from midnight to noon on June 8th, 2011.

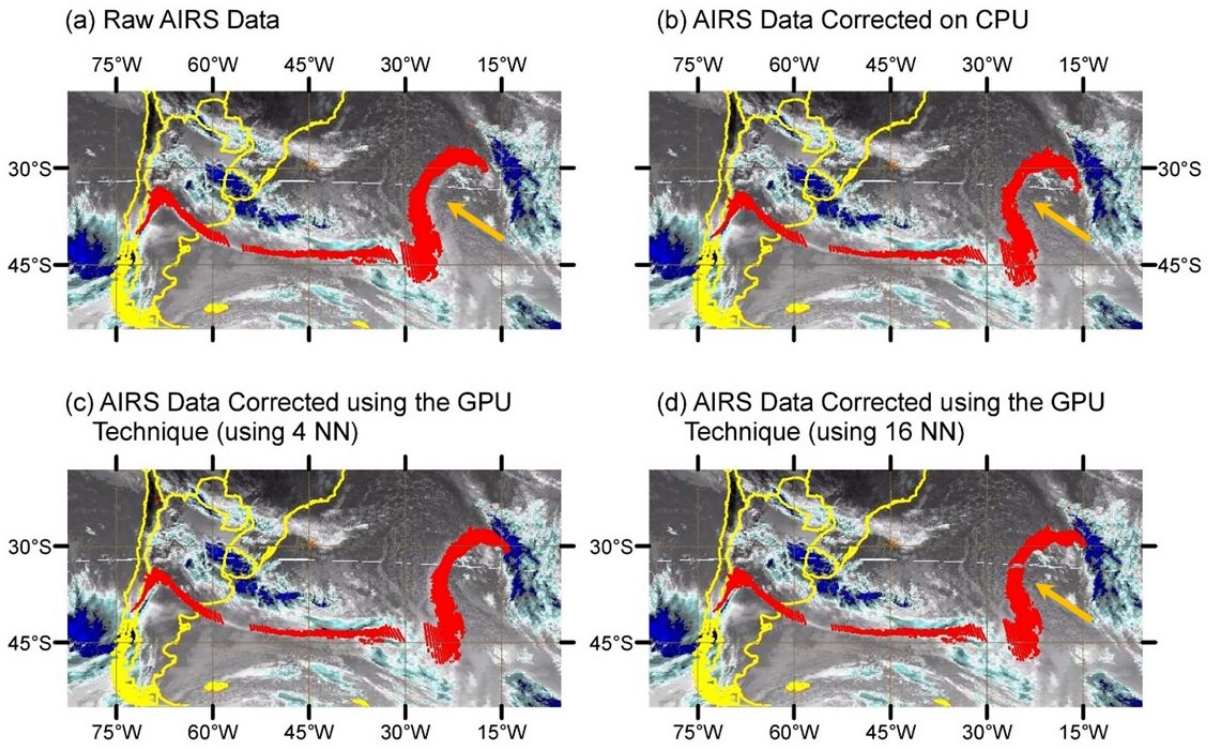


Figure 10.4: Comparison at 20110606174500 UTC between GOES-12 IR image, raw AIRS data, and three types of AIRS corrected plumes. (a) raw AIRS (no correction), (b) CPU corrected plume, (c) GPU plume calculated from 4 NN (nearest neighbors) , and (d) GPU plume calculated from 16 NN.

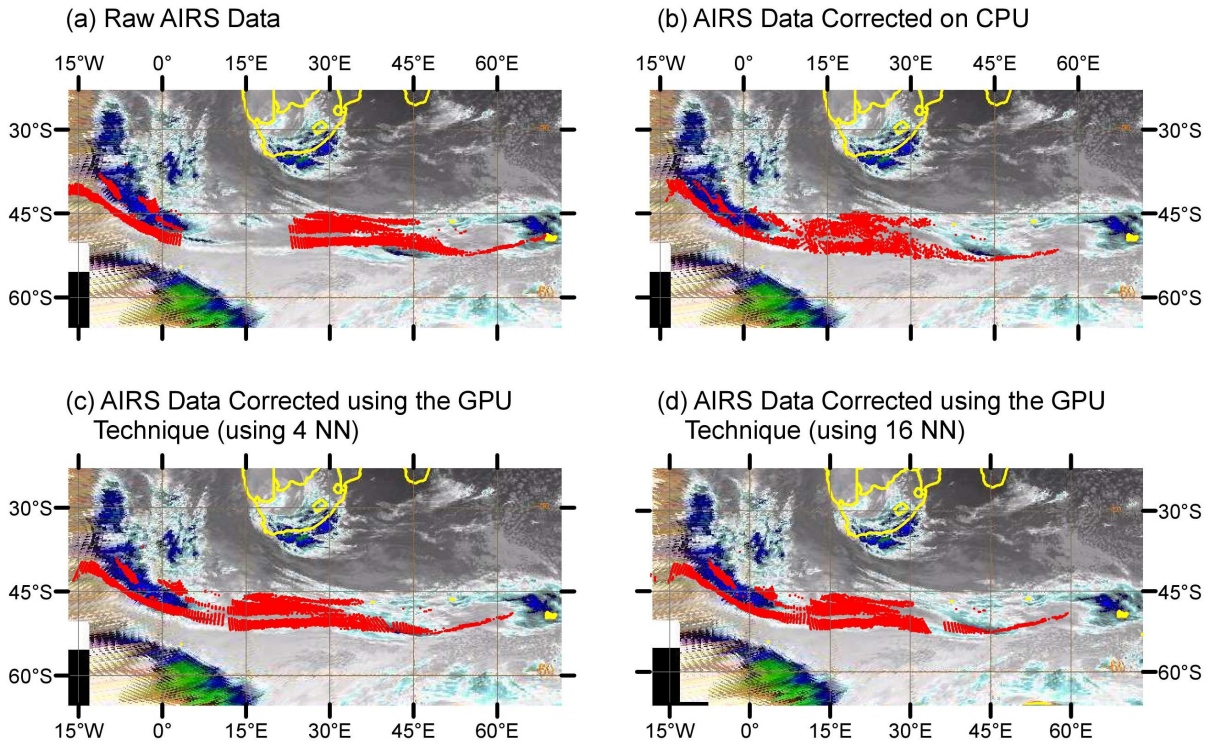


Figure 10.5: Comparison at 20110608054500 UTC between METEOSAT-7 IR image, raw AIRS data, and three types of AIRS corrected plumes. (a) raw AIRS (no correction), (b) CPU corrected plume, (c) GPU plume calculated from 4 NN (nearest neighbors), and (d) GPU plume calculated from 16 NN.

Chapter 11

Conclusion

We defined the problem of scientific data visualization in modern application domains in light of three axial data characteristics. Our methodology to address the problem was inspired by the design study methodology of problem-driven visualization research. We identified important research questions to address the problem and stated hypotheses.

In answering the identified questions, we presented a framework for the visualization and fusion of multimodal time-variant three-dimensional datasets. The proposed framework adopts a human-centered problem-driven solution approach, in which the human is a central participant and data fusion is an active problem solving experience. To provide a reference implementation of the framework, a visualization system was designed and developed to solve the problem of visual fusion within the context of atmospheric science domain.

System usability was evaluated to test Hypothesis $H1$. Since evaluation steps begin with *understanding*, an exploration subsystem was presented to support the iterative construction of a cognitive model that encodes this understanding. In this iterative process, the brain is viewed as an information processor whose operation speed needs to be matched with the interaction capabilities provided by the system. This has set the focus of our solution approach on interactivity and

motivated the development of GPU-parallel techniques.

We presented StreamProbe, a GPU-based interactive probing tool for dense dynamic flow fields, which empowers the user to reduce clutter and dynamically filter flow pathlines at any potential region of interest. This clutter reduction tool enabled the experts to generate hypotheses about flow parameters within specific regions, and set expectations for the behavior of the phenomena under study. This, in turn, led to the extraction of ROIs from satellite datasets, based on this expected behavior.

Once *understanding* of the studied phenomena and their manifestation in the available datasets are achieved, the next step in the framework enables the expert to communicate the constructed cognitive model from the brain back into the visualization system. This communication takes the form of interpolation and tracking parameters that enable the system to create a reference dataset, and then use this reference for the tracking and time-correction of extracted ROIs from the different modalities. The result of this reconstruction is a continuous interactive animation that calculates tracked locations and visualizes them interactively, to support real-time hypothesis testing.

We demonstrated this cognitive model construction process and parameter specification though a usability questionnaire. A sequence of analysis tasks were performed by domain experts, and feedback about their confidence in their answers aided by the offered visualization techniques was collected. Results have shown great potential of the proposed framework to support cognitive model and reference data construction. These results have proven hypothesis $H1$, that the construction of such cognitive model was made possible by exploration tasks and was communicated back to the system in the form of a reference set, and have paved the way for research on the ability of the framework to support this model construction process for a broader audience.

To make interactive tracking possible, intelligent data structures have been proposed to accelerate neighborhood matching and interpolation on the GPU. Highly interactive results were ob-

tained for thousands of points and tracking trajectories. Performance and accuracy of the results have proven that real-time interaction is achievable during the exploration, reconstruction, and enhancement of complex three-dimensional data, by leveraging GPU-based techniques and advanced acceleration data structures. This proves our second hypothesis $H2$.

The contributions of our work include the characterization of scientific data in light of three axial characteristics, and a classification of research in the literature based on this characterization. In addition, an interdisciplinary approach to the problem definition of scientific data visualization was presented from the perspectives of cognitive science, Human-Computer Interaction, data fusion, and computer graphics. Visualization goals were set to serve problem solving phases from cognitive model construction to hypothesis generation and testing and results checking. Concrete tasks were derived from the set goals. Most importantly, the proposed framework provides a generic process model for the visualization of multi-modal time-varying $3D$ datasets, that can guide future research in the field. A reference implementation was developed to test and evaluate the proposed interactive visualization techniques.

Bibliography

[AA90] ADELSON, Edward H. ; ANANDAN, Padmanabhan: *Ordinal characteristics of transparency*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1990

[AB94] ADAMS, Rolf ; BISCHOF, Leanne: Seeded region growing. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 16 (1994), Nr. 6, S. 641–647

[Adi] ADIV, Gilad: Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on*

[AM93] ARYA, Sunil ; MOUNT, David M.: Algorithms for fast vector quantization. In: *Data Compression Conference, 1993. DCC'93.* IEEE, 1993, S. 381–390

[AS11] ANGEL, Edward ; SHREINER, Dave: *interactive computer graphics a top-down approach with opengl (6th edition)*. 6th. Addison-Wesley, 2011

[Ban90] BANCHOFF, Thomas: *Beyond the third dimension: Geometry, computer graphics, and higher dimensions*. WH Freeman & Co., 1990

[BBF+11] BUSKING, Stef ; BOTHA, Charl P. ; FERRARINI, Luca ; MILLES, Julien ; POST, Frits H.: Image-based rendering of intersecting surfaces for dynamic comparative visualization. In: *The visual computer* 27 (2011), Nr. 5, S. 347–363

[BG10] BONAWITZ, Elizabeth B. ; GRIFFITHS, Thomas L.: Deconfounding hypothesis generation and evaluation in Bayesian models. In: *Proceedings of the 32nd annual conference of the cognitive science society*, 2010, S. 2260–2265

[BRW07] BOSSÉ, Éloi ; ROY, Jean ; WARK, Steve: *Concepts, models, and tools for information fusion*. Artech House, 2007

[BS95] BANKS, David C. ; SINGER, Bart A.: A predictor-corrector technique for visualizing unsteady flow. In: *Visualization and Computer Graphics, IEEE Transactions on* 1 (1995), Nr. 2, S. 151–163

[BWF+10] BORN, S. ; WIEBEL, A. ; FRIEDRICH, J. ; SCHEUERMANN, G. ; BARTZ, D.: Illustrative Stream Surfaces. In: *Visualization and Computer Graphics, IEEE Transactions on* 16 (2010), Nr. 6, S. 13291338

[CBC10] CIPILOGLU, Zeynep ; BULBUL, Abdullah ; CAPIN, Tolga: A framework for enhancing depth perception in computer graphics. In: *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization* ACM, 2010, S. 141–148

[CL93] CABRAL, Brian ; LEEDOM, Leith C.: Imaging vector fields using line integral convolution. In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* ACM, 1993, S. 263–270

[COJ15] CHANDLER, J. ; OBERMAIER, H. ; JOY, K.I.: Interpolation-Based Pathline Tracing in Particle-Based Flow Visualization. In: *Visualization and Computer Graphics, IEEE Transactions on* 21 (2015), Jan, Nr. 1, S. 68–80. `http://dx.doi.org/10.1109/TVCG.2014.2325043`. – DOI 10.1109/TVCG.2014.2325043. – ISSN 1077–2626

[CPP+10] CAO, Yong ; PATNAIK, Debprakash ; PONCE, Sean ; ARCHULETA, Jeremy ; BUTLER, Patrick ; FENG, Wu-chun ; RAMAKRISHNAN, Naren: Towards chip-on-chip neuroscience: fast mining of neuronal spike streams using graphics hardware. In:

Proceedings of the 7th ACM international conference on Computing frontiers ACM, 2010, S. 1–10

[Das94] DASARATHY, Belur V.: *Decision fusion*. Bd. 1994. IEEE Computer Society Press Los Alamitos, CA, 1994

[Das97] DASARATHY, Belur V.: Sensor fusion potential exploitation-innovative architectures and illustrative applications. In: *Proceedings of the IEEE* 85 (1997), Nr. 1, S. 24–38

[EGG+14] ELSHEHALY, Mai ; GRACANIN, Denis ; GAD, Mohamed ; ELMONGUI, Hicham ; MATKOVIC, Kresimir: StreamProbe: A novel GPU-based selection technique for interactive flow field exploration. In: *Proceedings of IEEE VIS 2014* IEEE, 2014, S. 110–119

[EGG+15] ELSHEHALY, Mai ; GRAČANIN, Denis ; GAD, Mohamed ; ELMONGUI, Hicham G. ; MATKOVIC, Kresimir: Interactive Fusion and Tracking For Multi-Modal Spatial Data Visualization. In: *Computer Graphics Forum* (2015). `http://dx.doi.org/10.1111/cgf.12637`. – DOI 10.1111/cgf.12637

[EHK+04] ENGEL, Klaus ; HADWIGER, Markus ; KNISS, Joe M. ; LEFOHN, Aaron E. ; SALAMA, Christof R. ; WEISKOPF, Daniel: Real-time volume graphics. In: *ACM Siggraph 2004 Course Notes* ACM, 2004, S. 29

[EHK+06] ENGEL, Klaus ; HADWIGER, Markus ; KNISS, Joe M. ; REZK-SALAMA, Christof ; WEISKOPF, Daniel: *Real-time volume graphics*. Ak Peters Natick, 2006

[Elv92] ELVINS, T.T.: A survey of algorithms for volume visualization. In: *ACM Siggraph Computer Graphics* 26 (1992), Nr. 3, S. 194–201. – ISSN 0097–8930

[FD08] FRIENDLY, M ; DENIS, DJ: *Milestones in the history of thematic cartography, statistical graphics, and data visualization. Retrieved July 13, 2008*. 2008

[FMHC07] FANG, Zhe ; MOELLER, Torsten ; HAMARNEH, Ghassan ; CELLER, Anna: Visualization and exploration of time-varying medical image data sets. In: *Proceedings of Graphics Interface 2007* ACM, 2007, S. 281–288

[FN13] FOO, Pek H. ; NG, Gee W.: High-level Information Fusion: An Overview. In: *J. Adv. Inf. Fusion* 8 (2013), Nr. 1, S. 33–72

[GGZL14] GOMEZ, Steven R. ; GUO, Hua ; ZIEMKIEWICZ, Caroline ; LAIDLAW, David H.: An insight-and task-based methodology for evaluating spatiotemporal visual analytics. In: *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on* IEEE, 2014, S. 63–72

[GHSR14] GRIESSBACH, S. ; HOFFMANN, L. ; SPANG, R. ; RIESE, M.: Volcanic ash detection with infrared limb sounding: MIPAS observations and radiative transfer simulations. In: *Atmospheric Measurement Techniques* 7 (2014), Nr. 5, 1487–1507. http://dx.doi.org/10.5194/amt-7-1487-2014. – DOI 10.5194/amt–7–1487–2014

[GRT13] GÜNTHER, Tobias ; RÖSSL, Christian ; THEISEL, Holger: Opacity optimization for 3D line fields. In: *ACM Transactions on Graphics (TOG)* 32 (2013), Nr. 4, S. 120

[GSFT14] GÜNTHER, T. ; SCHULZE, M. ; FRIEDERICI, A. ; THEISEL, H.: Visualizing the Aftermath of Volcanic Eruptions. Paris, France : IEEE Visualization 2014, 2014. – Winning entry

[Han98] HANSON, Andrew J.: Computer graphics beyond the third dimension. In: *Course Notes for SIGGRAPH* 98 (1998)

[HGM14] HOFFMANN, Lars ; GRIESSBACH, Sabine ; MEYER, Catrin I.: *Volcanic emissions from AIRS observations: detection methods, case study, and statistical analysis*. http://dx.doi.org/10.1117/12.2066326. Version: 2014

[HH89] HELMAN, James ; HESSELINK, Lambertus: Representation and display of vector field topology in fluid flow data sets. In: *Computer* 22 (1989), Nr. 8, S. 27–36

[HH91] HELMAN, James L. ; HESSELINK, Lambertus: Visualizing vector field topology in fluid flows. In: *IEEE Computer Graphics and Applications* 11 (1991), Nr. 3, S. 36–46

[HHT00] HALL, MJ ; HALL, SA ; TATE, T: Removing the HCI bottleneck: how the human computer interface (HCI) affects the performance of data fusion systems. In: *Proc. 2000 Meeting of the MSS, Nat. Symp. Sensor and Data Fusion*, 2000, S. 89–104

[HM04] HALL, David L. ; MCMULLEN, Sonya A.: *Mathematical techniques in multisensor data fusion*. Artech House, 2004

[HR09] HOWARD, Ian P. ; ROGERS, Brian J.: *Seeing in depth*. Oxford University Press, 2009

[HTS07] HARRISON, Steve ; TATAR, Deborah ; SENGERS, Phoebe: The three paradigms of HCI. In: *Alt. Chi. Session at the SIGCHI Conference on Human Factors in Computing Systems San Jose, California, USA*, 2007, S. 1–18

[IEE14] IEEE: *http://www.viscontest.rwth-aachen.de/index.html*. `http://www.viscontest.rwth-aachen.de/index.html`. Version: 2014

[IEGC08] ISENBERG, Tobias ; EVERTS, Maarten H. ; GRUBERT, Jens ; CARPENDALE, Sheelagh: Interactive exploratory visualization of 2d vector fields. In: *Computer Graphics Forum* Bd. 27 Wiley Online Library, 2008, S. 983–990

[JL97] JOBARD, Bruno ; LEFER, Wilfrid: Creating evenly-spaced streamlines of arbitrary density. In: *Visualization in scientific computing* 97 (1997), S. 43–55

[JL01] JOBARD, Bruno ; LEFER, Wilfrid: Multiresolution Flow Visualization. In: *WSCG (Posters)* Citeseer, 2001, S. 34–35

[JMM+06] JOHNSON, C. ; MOORHEAD, R. ; MUNZNER, T. ; PFISTER, H. ; RHEINGANS, P. ; YOO, TS: NIH/NSF Visualization Research Challenges January 2006 / National Institute of Health/ National Science Foundation. IEEE Computer Society Press, 2006. – Forschungsbericht

[Kaj86] KAJIYA, James T.: The rendering equation. In: *ACM SIGGRAPH Computer Graphics* Bd. 20 ACM, 1986, S. 143–150

[Kar98] KARAKOWSKI, J: Towards Visual Data Fusion. In: *Military Sensing Series National Symposium on Sensor and Data Fusion International Open Session* (1998)

[Kle04] KLEIN, Lawrence A.: *Sensor and data fusion: a tool for information assessment and decision making*. Bd. 324. Spie Press Bellingham^ eWA WA, 2004

[KLM+08] KEHRER, Johannes ; LADSTADTER, Florian ; MUIGG, Philipp ; DOLEISCH, Helmut ; STEINER, Andrea ; HAUSER, Helwig: Hypothesis generation in climate research with interactive visual data exploration. In: *Visualization and Computer Graphics, IEEE Transactions on* 14 (2008), Nr. 6, S. 1579–1586

[KMG+06] KONYHA, Zoltan ; MATKOVIC, Kresimir ; GRACANIN, Denis ; JELOVIC, Mario ; HAUSER, Helwig: Interactive visual analysis of families of function graphs. In: *Visualization and Computer Graphics, IEEE Transactions on* 12 (2006), Nr. 6, S. 1373–1385

[Kru90a] KRUEGER, Wolfgang: The application of transport theory to visualization of 3D scalar data fields. In: *Proceedings of the 1st conference on Visualization'90* IEEE Computer Society Press, 1990, S. 273–280

[Kru90b] KRUEGER, Wolfgang: Volume rendering and data feature enhancement. In: *ACM SIGGRAPH Computer Graphics* Bd. 24 ACM, 1990, S. 21–26

[KS91] KALIVAS, Dimitrios S. ; SAWCHUK, Alexander A.: A region matching motion estimation algorithm. In: *CVGIP: Image Understanding* 54 (1991),

Nr. 2, 275 - 288. http://dx.doi.org/http://dx.doi.org/10.1016/1049-9660(91)90068-Z. – DOI http://dx.doi.org/10.1016/1049–9660(91)90068–Z. – ISSN 1049–9660

[KWm12] KIRK, David B. ; WEN-MEI, W H.: *Programming massively parallel processors: a hands-on approach*. Newnes, 2012

[Lar08] LARAMEE, Zhenmin Peng Robert S.: Vector Glyphs for Surfaces: A Fast and Simple Glyph Placement Algorithm for Adaptive Resolution Meshes. In: *Vision, Modeling, and Visualization 2008: Proceedings, October 8-10, 2008, Konstanz, Germany* (2008), S. 61

[LC87] LORENSEN, William E. ; CLINE, Harvey E.: Marching cubes: A high resolution 3D surface construction algorithm. In: *SIGGRAPH Comput. Graph.* 21 (1987), August, 163–169. http://dx.doi.org/http://doi.acm.org.ezproxy.lib.vt.edu:8080/10.1145/37402.37422. – DOI http://doi.acm.org.ezproxy.lib.vt.edu:8080/10.1145/37402.37422. – ISSN 0097–8930

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

[LH74] LAWSON, Charles L. ; HANSON, Richard J.: *Solving least squares problems*. Bd. 161. SIAM, 1974

[LHD+04] LARAMEE, Robert S. ; HAUSER, Helwig ; DOLEISCH, Helmut ; VROLIJK, Benjamin ; POST, Frits H. ; WEISKOPF, Daniel: The State of the Art in Flow Visualization: Dense and Texture-Based Techniques. In: *Computer Graphics Forum* Bd. 23 Wiley Online Library, 2004, S. 203–221

[LHM06] LEWIS, Daniel ; HAROZ, Steve ; MA, Kwan-Liu: Layout of multiple views for volume visualization: A user study. In: *Advances in Visual Computing*. Springer, 2006, S. 215–226

[LRTM07] LAM, H. ; RUSSELL, D. ; TANG, D. ; MUNZNER, T.: Session Viewer: Visual Exploratory Analysis of Web Session Logs. In: *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, 2007, S. 147–154

[LWS13] LI, Yifei ; WANG, Chaoli ; SHENE, Ching-Kuang: Streamline similarity analysis using bag-of-features. In: *IS&T/SPIE Electronic Imaging* International Society for Optics and Photonics, 2013, S. 90170N–90170N

[Mat05] MATLIN, Margaret W.: *Cognition. Hoboken*. 2005

[Max95] MAX, Nelson: Optical models for direct volume rendering. In: *Visualization and Computer Graphics, IEEE Transactions on* 1 (1995), Nr. 2, S. 99–108

[MDR91] MONGA, Olivier ; DERICHE, Rachid ; ROCCHISANI, Jean-Marie: 3D edge detection using recursive filtering: application to scanner images. In: *CVGIP: Image Understanding* 53 (1991), Nr. 1, S. 76–87

[MFGH08] MATKOVIC, Kresimir ; FREILER, Wolfgang ; GRACANIN, Denis ; HAUSER, Helwig: ComVis: A coordinated multiple views system for prototyping new visualization technology. In: *Information Visualisation, 2008. IV'08. 12th International Conference* IEEE, 2008, S. 215–220

[MGKH09] MATKOVIC, Kresimir ; GRACANIN, Denis ; KLARIN, Borislav ; HAUSER, Helwig: Interactive visual analysis of complex scientific data as families of data surfaces. In: *Visualization and Computer Graphics, IEEE Transactions on* 15 (2009), Nr. 6, S. 1351–1358

[MKG+02] MCKENNA, Daniel S. ; KONOPKA, P ; GROOSS, JU ; GUNTHER, G ; MÜLLER, R: A chemical lagrangian model of the stratosphere (CLAMS). In: *12th Conference on the Middle Atmosphere*, 2002

[MLP+10] MCLOUGHLIN, Tony ; LARAMEE, Robert S. ; PEIKERT, Ronald ; POST, Frits H. ; CHEN, Min: Over Two Decades of Integration-Based, Geometric Flow Visu-

alization. In: *Computer Graphics Forum* Bd. 29 Wiley Online Library, 2010, S. 1807–1829

[MMK12] MORI, Masahiro ; MACDORMAN, Karl F. ; KAGEKI, Norri: The uncanny valley [from the field]. In: *Robotics & Automation Magazine, IEEE* 19 (2012), Nr. 2, S. 98–100

[Mou98] MOUNT, David M.: ANN programming manual / Technical report, Dept. of Computer Science, U. of Maryland. 1998. – Forschungsbericht

[MPSS05] MALLO, Ovidio ; PEIKERT, Ronald ; SIGG, Christian ; SADLO, Filip: Illuminated lines revisited. In: *Visualization, 2005. VIS 05. IEEE* IEEE, 2005, S. 19–26

[MS03] MULLER, W. ; SCHUMANN, H.: Visualization methods for time-dependent data-an overview. In: *Simulation Conference, 2003. Proceedings of the 2003 Winter* Bd. 1 IEEE, 2003. – ISBN 0780381319, S. 737–745

[MSS99] MASUCH, Maic ; SCHLECHTWEG, Stefan ; SCHULZ, Ronny: Speedlines: depicting motion in motionless pictures. In: *ACM SIGGRAPH 99 Conference abstracts and applications* ACM, 1999, S. 277

[MTHG03] MATTAUSCH, Oliver ; THEUSSL, Thomas ; HAUSER, Helwig ; GRÖLLER, Eduard: Strategies for interactive exploration of 3d flow using evenly-spaced illuminated streamlines. In: *Proceedings of the 19th spring conference on Computer graphics* ACM, 2003, S. 213–222

[Mun09] MUNZNER, T.: A Nested Model for Visualization Design and Validation. In: *Visualization and Computer Graphics, IEEE Transactions on* 15 (2009), Nov, Nr. 6, S. 921–928. http://dx.doi.org/10.1109/TVCG.2009.111. – DOI 10.1109/TVCG.2009.111. – ISSN 1077–2626

[MVW05] MOBERTS, B. ; VILANOVA, A. ; WIJK, J.J. van: Evaluation of fiber clustering methods for diffusion tensor imaging. In: *Visualization, 2005. VIS 05. IEEE*, 2005, S. 65–72

[MWSJ13] MA, Jun ; WANG, Chaoli ; SHENE, C ; JIANG, Jingfeng: A Graph-Based Interface for Visual Analytics of 3D Streamlines and Pathlines. In: *IEEE Transactions on Visualization and Computer Graphics* (2013)

[MWSJ14] MA, Jun ; WANG, Chaoli ; SHENE, Ching-Kuang ; JIANG, Jingfeng: A Graph-Based Interface for VisualAnalytics of 3D Streamlines and Pathlines. In: *Visualization and Computer Graphics, IEEE Transactions on* 20 (2014), Nr. 8, S. 1127–1140

[ND05] NIENHAUS, Marc ; DOLLNER, Jurgen: Depicting dynamics using principles of visual art and narrations. In: *Computer Graphics and Applications, IEEE* 25 (2005), Nr. 3, S. 40–51

[Nyq24] NYQUIST, Harry: Certain factors affecting telegraph speed. In: *American Institute of Electrical Engineers, Transactions of the* 43 (1924), S. 412–422

[PB07] PREIM, Bernhard ; BARTZ, Dirk: *Visualization in medicine: theory, algorithms, and applications*. Morgan Kaufmann, 2007

[PKR+57] POLYA, George ; KADDOUCH, Robert ; RENOU, Marcel ; COMTOIS, Manon ; DUBOIS, Marie-Josée: How to solve it: a new aspect of mathematical method/G. (1957)

[PL09] PENG, Zhenmin ; LARAMEE, Robert S.: Higher Dimensional Vector Field Visualization: A Survey. In: *TPCG*, 2009, S. 149–163

[PPH+08] PARK, Seung I. ; PONCE, Sean P. ; HUANG, Jing ; CAO, Yong ; QUEK, Francis: Low-cost, high-speed computer vision using NVIDIA's CUDA architecture. In: *Applied Imagery Pattern Recognition Workshop, 2008. AIPR'08. 37th IEEE* IEEE, 2008, S. 1–7

[PS93] PAGENDARM, Hans-Georg ; SEITZ, B: An algorithm for detection and visualization of discontinuities in scientific data fields applied to flow data with shock waves. In: *Scientific Visualization: Advanced Software Techniques* (1993), S. 161–177

[RdE90] ROSSOW, W. B. ; DEL GENIO, A. D. ; EICHLER, T.: Cloud-tracked winds from Pioneer Venus OCPP images. In: *Journal of Atmospheric Sciences* 47 (1990), September, S. 2053–2084. http://dx.doi.org/10.1175/1520-0469(1990)047<2053:CTWFVO>2.0.CO;2. – DOI 10.1175/1520–0469(1990)047¡2053:CTWFVO¿2.0.CO;2

[RGJ12] RATTNER, Alexander S. ; GUILLEN, Donna P. ; JOSHI, Alark: Generalized Framework and Algorithms for Illustrative Visualization of Time-Varying Data on Unstructured Meshes / Idaho National Laboratory (INL). 2012. – Forschungsbericht

[RPS01] REINDERS, Freek ; POST, Frits H. ; SPOELDER, Hans J.: Visualization of time-dependent data with feature tracking and event detection. In: *The Visual Computer* 17 (2001), Nr. 1, 55-71. http://dx.doi.org/10.1007/PL00013399. – DOI 10.1007/PL00013399. – ISSN 0178–2789

[Sab88] SABELLA, Paolo: A rendering algorithm for visualizing 3D scalar fields. In: *ACM SIGGRAPH computer graphics* Bd. 22 ACM, 1988, S. 51–58

[SBM92] SPRINGMEYER, Rebecca R. ; BLATTNER, Meera M. ; MAX, Nelson L.: A Characterization of the Scientific Data Analysis Process. In: *Proceedings of the 3rd Conference on Visualization '92*. Los Alamitos, CA, USA : IEEE Computer Society Press, 1992 (VIS '92). – ISBN 0–8186–2896–0, 235–242

[SEG+15] SPLECHTNA, Rainer ; ELSHEHALY, Mai ; GRAČANIN, Denis ; URAS, Mario ; BÜHLER, Katja ; MATKOVIĆ, Krešimir: Interactive interaction plot, Supporting Parameter Space Exploration in a Design Phase. In: *The Visual Computer* 31 (2015), Nr. 6-8, 1055-1065. http://dx.doi.org/10.1007/

s00371-015-1095-x. – DOI 10.1007/s00371–015–1095–x. – ISSN 0178–2789

[Sel06] SELIKOFF, Nathan: Digital chronophotography. In: *International Conference on Computer Graphics and Interactive Techniques: ACM SIGGRAPH 2006 Sketches* Bd. 30, 2006

[She68] SHEPARD, Donald: A two-dimensional interpolation function for irregularly-spaced data. In: *Proceedings of the 1968 23rd ACM national conference* ACM, 1968, S. 517–524

[Shn93] SHNEIDERMAN, Ben: 1.1 direct manipulation: a step beyond programming languages. In: *Sparks of innovation in human-computer interaction* 17 (1993), S. 1993

[Shn96] SHNEIDERMAN, Ben: The eyes have it: A task by data type taxonomy for information visualizations. In: *Visual Languages, 1996. Proceedings., IEEE Symposium on* IEEE, 1996, S. 336–343

[SLF+13] STEVENSON, John A. ; LOUGHLIN, Susan C. ; FONT, Anna ; FULLER, Gary W. ; MACLEOD, Alison ; OLIVER, Ian W. ; JACKSON, Ben ; HORWELL, Claire J. ; THORDARSON, Thor ; DAWSON, Ian: UK monitoring and deposition of tephra from the May 2011 eruption of Grímsvötn, Iceland. In: *Journal of Applied Volcanology* 2 (2013), Nr. 1, S. 1–17

[SM04] SCHUSSMAN, Greg ; MA, Kwan-Liu: Anisotropic volume rendering for extremely dense, thin line data. In: *Proceedings of the conference on Visualization'04* IEEE Computer Society, 2004, S. 107–114

[SMM12] SEDLMAIR, Michael ; MEYER, Miriah ; MUNZNER, Tamara: Design study methodology: Reflections from the trenches and the stacks. In: *Visualization and Computer Graphics, IEEE Transactions on* 18 (2012), Nr. 12, S. 2431–2440

[SP97] SADARJOEN, I ; POST, Frits H.: Deformable surface techniques for field visualization. In: *Computer Graphics Forum* Bd. 16 Wiley Online Library, 1997, S. C109–C116

[SPY94] SETHI, Ishwar K. ; PATEL, Nilesh V. ; YOO, Jae H.: A general approach for token correspondence. In: *Pattern Recognition* 27 (1994), Nr. 12, 1775 - 1786. http://dx.doi.org/http://dx.doi.org/10.1016/0031-3203(94)90093-0. – DOI http://dx.doi.org/10.1016/0031–3203(94)90093–0. – ISSN 0031–3203

[SSEH03] SCHPOK, Joshua ; SIMONS, Joseph ; EBERT, David S. ; HANSEN, Charles: A Real-time Cloud Modeling, Rendering, and Animation System. In: *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. Aire-la-Ville, Switzerland, Switzerland : Eurographics Association, 2003 (SCA '03). – ISBN 1–58113–659–5, 160–166

[SSZC94] SAMTANEY, Ravi ; SILVER, Deborah ; ZABUSKY, Norman ; CAO, Jim: Visualizing features and tracking their evolution. In: *Computer* 27 (1994), Nr. 7, S. 20–27

[SW] SILVER, D. ; WANG, X.: In: *Visualization '96. Proceedings.*

[TB96] TURK, Greg ; BANKS, David: Image-guided streamline placement. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* ACM, 1996, S. 453–460

[TLLH13] TURKAY, Cagatay ; LUNDERVOLD, Arvid ; LUNDERVOLD, Astri J. ; HAUSER, Helwig: Hypothesis generation by interactive visual exploration of heterogeneous medical data. In: *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*. Springer, 2013, S. 1–12

[USG] USGS ; U.S. GEOLOGICAL SURVEY (Hrsg.): *Ash Properties and Dispersal by Wind*. http://volcanoes.usgs.gov/ash/properties.html, Abruf: 13.07.2015

[VRFW14] VAQUERO, Ricardo Manuel M. ; RZEPECKI, Jan ; FRIESE, Karl-Ingo ; WOLTER, Franz-Erich: Visualization and user interaction methods for multiscale biomedical data. In: *3D Multiscale Physiological Human*. Springer, 2014, S. 107–133

[VW91] VAN WIJK, Jarke J.: Spot noise texture synthesis for data visualization. In: *ACM SIGGRAPH Computer Graphics* Bd. 25 ACM, 1991, S. 309–318

[VWPSP96a] VAN WALSUM, T. ; POST, F.H. ; SILVER, D. ; POST, F.J.: Feature extraction and iconic visualization. In: *Visualization and Computer Graphics, IEEE Transactions on* 2 (1996), Nr. 2, S. 111–119. http://dx.doi.org/10.1109/2945.506223. – DOI 10.1109/2945.506223

[VWPSP96b] VAN WALSUM, Theo ; POST, Frits H. ; SILVER, Deborah ; POST, Frank J.: Feature extraction and iconic visualization. In: *Visualization and Computer Graphics, IEEE Transactions on* 2 (1996), Nr. 2, S. 111–119

[Wan07] WANG, Yingxu: The OAR Model of Neural Informatics for Internal Knowledge Representation in the Brain. In: *IJCINI* 1 (2007), Nr. 3, S. 66–77

[War00] WARE, Colin: *Information visualization: Perception for Design*. Bd. 2. Morgan Kaufmann San Francisco, 2000

[WG10] WANG, Yingxu ; GAFUROV, Davrondzhon: The cognitive process of comprehension: A formal description. In: *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)* 4 (2010), Nr. 3, S. 44–58

[Wri07] WRIGHT, Helen: *Introduction to Scientific Visualization*. Springer, 2007

[WWS03a] WOODRING, J. ; WANG, Chaoli ; SHEN, Han-Wei: High dimensional direct rendering of time-varying volumetric data. In: *Visualization, 2003. VIS 2003. IEEE*, 2003, S. 417–424

[WWS03b] WOODRING, Jonathan ; WANG, Chaoli ; SHEN, Han-Wei: *High dimensional direct rendering of time-varying volumetric data*. IEEE, 2003

[WYM08] WANG, Chaoli ; YU, Hongfeng ; MA, Kwan-Liu: Importance-driven time-varying data visualization. In: *Visualization and Computer Graphics, IEEE Transactions on* 14 (2008), Nr. 6, S. 1547–1554

[YLC13] YU, Li ; LU, Aidong ; CHEN, Wei: Visualization and analysis of 3D time-varying simulations with time lines. In: *Journal of Visual Languages & Computing* 24 (2013), Nr. 5, S. 402–418

[ZHWG08] ZHOU, Kun ; HOU, Qiming ; WANG, Rui ; GUO, Baining: Real-time KD-tree construction on graphics hardware. In: *ACM Transactions on Graphics (TOG)* Bd. 27 ACM, 2008, S. 126

[ZPLG13] ZHANG, Wenyao ; PEI, Yuezhu ; LIU, Beichen ; GUAN, Mengyuan: A Streamline Illumination Method for 3D Flow Fields. In: *Image and Graphics (ICIG), 2013 Seventh International Conference on* IEEE, 2013, S. 252–257

[ZWM13a] ZHENG, Lin ; WU, Yingcai ; MA, K: Perceptually Based Depth-Ordering Enhancement for Direct Volume Rendering. (2013)

[ZWM13b] ZHENG, Lin ; WU, Yingcai ; MA, Kwan-Liu: Perceptually-Based Depth-Ordering Enhancement for Direct Volume Rendering. In: *Visualization and Computer Graphics, IEEE Transactions on* 19 (2013), Nr. 3, S. 446–459

Appendix A

Usability Test

Based on the tasks we identified for the visual fusion of $3D$ time-variant multi-modal datasets, we designed a usability questionnaire, which was presented to domain experts. The goal was to test the system's ability to: *(i)* support the construction of a cognitive model that can be embodied in a reference dataset based on hypothesized data behavior; then *(ii)* test the ability of the user to verify the hypothesized behavior through interactive tracking. Our usability questionnaire constitutes of two major parts that encompass these two tests. This Appendix contains the list of questions and questionnaire layout as was presented to the experts.

User ID: _____
Date : _____

Part I: Visual Exploration of Multiple Datasets

On June 4th, 2011 a volcano called "Puyehue Cordón Caulle" erupted in Chile at the following coordinates:

- **Longitude (x)** = -72.117
- **Latitude (y)** = -40.590

Please select simulation trajectories passing through a probe of radius =1 around the eruption location during the first 12 hours of June 4th. Please answer the following:

Q1: In what direction is the plume initially expected to travel?

- ○ North ↑
- ○ East →
- ○ South ↓
- ○ West ←

Q2: What countries are expected to be immediately affected by the ash plume?

(Check all that applies)

- ☐ Argentina
- ☐ Peru
- ☐ Ecuador
- ☐ Brazil
- ☐ Uruguay

User ID: _____
Date : _____

Q3: For each of the following countries, write the date at which you expect the plume to have affected them (write "Never" if you think a country was never affected)

(mm/ dd/ yyyy hh:mm)

- South Africa : _____
- Australia : _____
- New Zealand : _____

Q4: How would you describe the flow carrying the plume?

- ☐ Turbulent (with chaotic changes, high variability)
- ☐ Laminar (in parallel layers, with no disruption)
- ☐ Neither

User ID: _____
Date : _____

Feedback Questions (1/3)

FQ1: Which of the above questions was easiest to answer using the visualization?

_____

FQ2: Which of the above questions was hardest to answer using the visualization?

_____

FQ3: (optional) What visualization features/interactions would help answer the questions more efficiently (faster and with less mental effort) ?

_____
_____
_____

User ID: _____
Date : _____

Q5: In order to extract the volcanic plume for "Puyehue Cordon Caulle" volcano, please specify the threshold value for:

- Sulfate Aerosol (SO_2): _____
- Ash index : _____

Q6:

(a) On what day does the detected plume travel away from the eruption location and no more detections are captured around the site of eruption?
(mm/ dd/ yyyy hh:mm)

_____

(b) Please rate your confidence in answering this question using this tool (1 = not at all confident, 7 = extremely confident)

Not at all Confident 1	2	3	4	5	6	Extremely Confident 7
○	○	○	○	○	○	○

Q7: Using a second probe, please select a Region Of Interest (ROI) within search range = 10 centered at coordinates (46 E, 47 S). What is the date and time at which: (mm/ dd / yyyy hh:mm)

(a) The simulation trajectories reach the ROI: ___ /___ / _____ ___: ___
(b) The detected plume enters the ROI: ___ /___ / _____ ___: ___
(c) The detected plume exists the ROI: ___ /___ / _____ ___: ___

Q8: Using perspective view, what is the altitude range in which the plume is expected to travel between the two probed locations?

_____ to _____ km.

User ID: _____
Date : _____

Feedback Questions (2/3)

FQ4: Did your answer for Q7 include two different dates and/or times for the simulation (a) and the actual detections (b)?

- ☐ Yes
- ☐ No

If yes, which of the two times do you think more accurately describes what happened in the real world?

- ☐ (a) Simulation
- ☐ (b) Detections

FQ5: Based on your exploration so far, how would you rate your confidence of the system's ability to provide the following information:

(1 = not at all confident, 7 = extremely confident)

1. The time during which the volcanic eruption lasted:

Not at all Confident 1	2	3	4	5	6	Extremely Confident 7
○	○	○	○	○	○	○

2. The altitude range of the resulting plume:

Not at all Confident 1	2	3	4	5	6	Extremely Confident 7
○	○	○	○	○	○	○

User ID: _____
Date : _____

3. The spatial range around the eruption location that contains simulation trajectories relevant to the eruption event:

Not at all Confident 1	2	3	4	5	6	Extremely Confident 7
○	○	○	○	○	○	○

4. The exact time at which the plume reached a region/country of interest:

Not at all Confident 1	2	3	4	5	6	Extremely Confident 7
○	○	○	○	○	○	○

User ID: _____
Date : _____

Part II: Data Visualization and Interactive Tracking

Q9:

(a) On what day does the detected plume travel away from the eruption location and no more detections are captured around the site of eruption?
(mm/ dd/ yyyy hh:mm)

_____

(b) Please rate your confidence on answering this question from the visualization (1 = not at all confident, 7 = extremely confident)

Not at all Confident 1	2	3	4	5	6	Extremely Confident 7
○	○	○	○	○	○	○

Q10: For the Region of Interest (ROI) defined within search radius = 10 centered at coordinates (46, - 47), please specify the date and time at which:

(mm/ dd/ yyyy hh:mm)

(a) The detected plume enters the ROI: ___ /___ / _____ ___: ___
(b) The detected plume exits the ROI: ___ /___ / _____ ___: ___

Q11: What is the altitude range of the plume on 6/6/2011?

_____ to _____ km

Q12: For each of the following countries, write the date at which you expect the plume to have affected them (write "Never" if you think a country was never affected) (mm/ dd/ yyyy hh:mm)

- South Africa : _____
- Australia : _____
- New Zealand : _____

User ID: _____
Date : _____

Feedback Questions (3/3)

FQ6: Based on the tracking tool, how would you rate your confidence in the system's ability to provide the following information:

(1 = not at all confident, 7 = extremely confident)

1. The time during which the volcanic eruption lasted:

Not at all Confident 1	2	3	4	5	6	Extremely Confident 7
○	○	○	○	○	○	○

2. The altitude range of the resulting plume:

Not at all Confident 1	2	3	4	5	6	Extremely Confident 7
○	○	○	○	○	○	○

3. The exact time at which the plume reached a region/country of interest:

Not at all Confident 1	2	3	4	5	6	Extremely Confident 7
○	○	○	○	○	○	○

Comments/Suggestions (optional):

_____
_____
_____
_____

Appendix B

GPU Performance

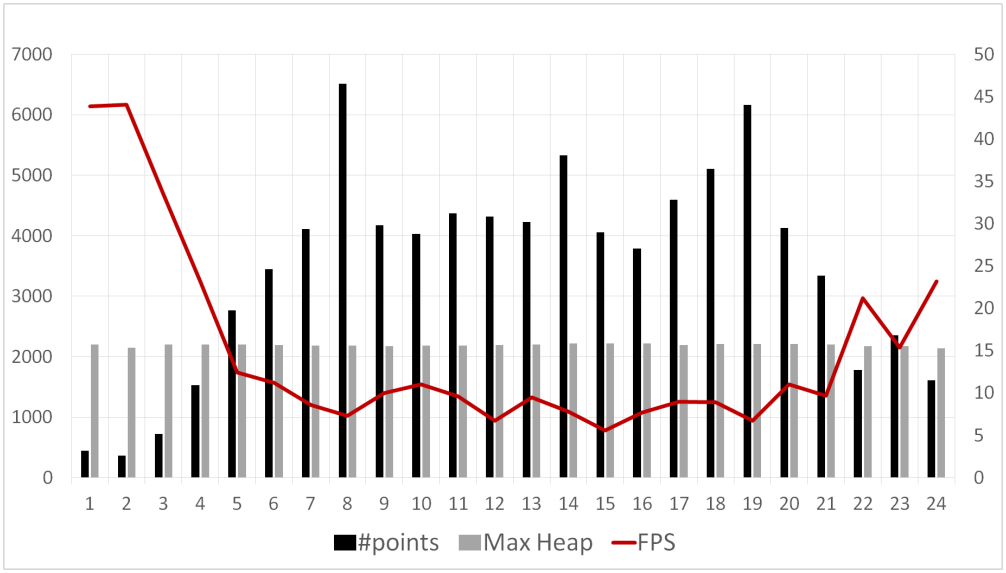
We used a laptop computer with Intel Core i7 CPU and NVIDIA GeForce 740M GPU with 4GB total graphics memory. The GPU tracking runs in OpenGL's vertex shader (GLSL version 4.50). When generating a reference model, the sample size (the number of points generated at each time step) has a major effect on the tracking algorithm's efficiency in terms of memory usage and speed. Since the model generation algorithm creates $sample\_size$ points at every time step, the number of points a time step t of the plume model is equal to $(t+1) \times sample\_size$, where $t = 0, \ldots, t_{max}$. We used linear nearest neighbor search within a time step which is $O(t \times sample\_size)$. This can be improved upon with the kd-tree forest structure we proposed. An advantage of the generated reference model, over kd-tree based neighborhood lookup of irregular simulation trajectories [COJ15], is that looking up a neighbor point's location at the destination time step is straight forward in the regular plume as we store the point at the same offset within every time step. The indices of the neighborhood points are directly used to fetch their updated locations at the destination time step. In the case of irregular simulation data, we needed auxiliary data structures to store trajectory IDs, and the offset at which each trajectory's vertices exist in memory (the **Lines** tables in Figure 7.4). After experimenting with different sample sizes, we have found that generating five points at a time resolution of one hour yields satisfactory results in terms of accuracy and performance.

Using the reference model, average frame rates are 17.7, 15.7, 10.2 and 6.2 for 1-NN, 4-NN, 8-NN and 16-NN, respectively. The system performs best with small neighborhoods because it maintains a list of k-NNs sorted by distance from query point. Upon inserting a new neighbor, it shifts all neighbors that are farther away from the query point upward in the list. This sorting step affects algorithm performance. The use of a binary heap could help address this problem to support larger neighborhoods.

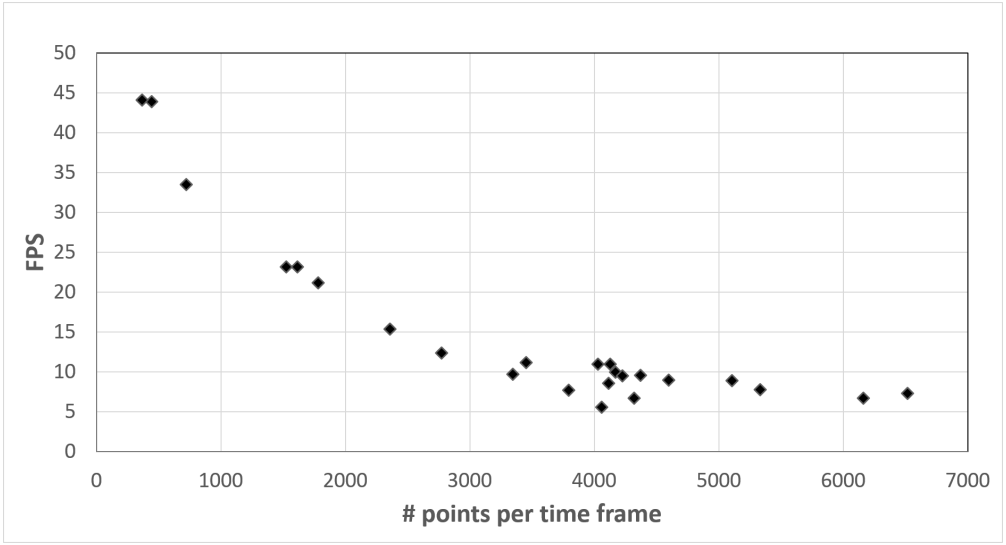
Without using a reference model, GPU tracking is implemented in OpenGL's vertex shader (GLSL version 4.50), and performance is measured by average frames per second (FPS) across a time span of twelve days following the volcanic eruption (from June 4th until June 16th, 2011). Figure B.1(a) displays measured parameters in 12 hours time frames sampled across this time span including number of particles to be tracked in each time frame (black), maximum heap size needed per tracking point in each (grey), and average frames per second (red line).

The relevance of the maximum heap size stems from the fact that it sets a minimum requirement for the amount of memory used per shader instance (i.e. per GPU thread) to perform BBF traversal of the kd-tree corresponding to its source time step and obtain the spatial nearest neighborhood within this step. We note, however, that it is nearly constant across time frames because it is determined by the number of points that exist in the simulation steps, which does not change significantly from one time frame to the next. The major effect on performance, however, is inflicted by the number of points in the tracking set that fall within a given time frame. This can be attributed to the fact that as the number of points increases, the GPU tries to accommodate their individual memory requirements to maintain a priority queue for the BBF traversal for each. The trend in the relationship between the size of the query subset and frames per second can be clearly observed in the scatter plot in Figure B.1(b).

We tested the StreamProbe technique using a laptop computer with Intel Core i7 CPU with 8GB memory and GeForce GT 740M/PCIe/SSE2 GPU with 1792MB total graphics memory, OpenGL version 4.4.0 and GLSL 4.40 NVIDIA via Cg compiler. We use a simple vertex array that contains



(a)



(b)

Figure B.1: Twelve days of real-time tracking ash detections from 06/04/2011 to 06/16/2011: (a) relationship between the number of points per time frame, maximum heap size, and fps, (b) the effect of the number of particles alone on average fps.

Table B.1: Pathline array size and construction time

No. Pathlines	Size in Memory (KB)	Time (sec)
600	4.68	3.88
800	6.25	9.25
1800	14.06	9.29
2400	18.75	10.7
3600	28.125	39.8
4800	37.5	36.5
5400	42.18	36.7
5817	45.44	39.5

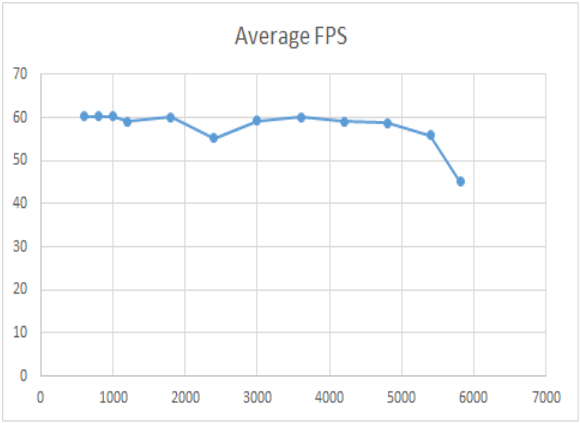


Figure B.2: Average frames per second for different numbers of pathlines.

all the vertices in a dataset in a vertex buffer on the GPU and a CPU-side vector of structures that holds indices of the first and last vertices of each streamline. This vector is constructed by the CPU once before the render. Table B.1 shows the time needed for the CPU to construct this vector of structures and its size in memory for different dataset sizes. When compared to the amount of time it takes to construct streamline hierarchies [MWSJ13], our approach is much more scalable and efficient in both space and time. The average frame rates are invariant to the size of the data with up to five thousand streamlines (Figure B.2). For datasets with larger size, the frame rate drops but remains interactive.

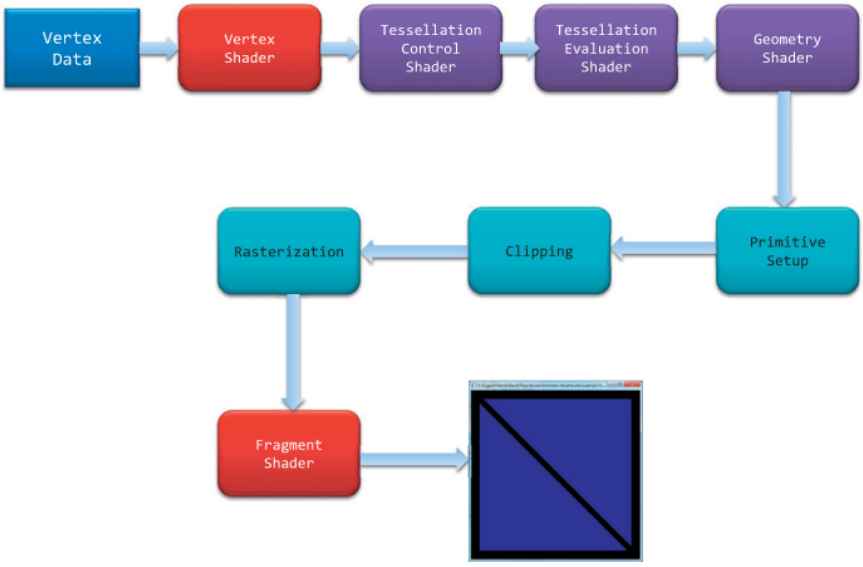


Figure B.3: OpenGL's Rendering Pipeline.