In Situ Summarization and Visual Exploration of Large-scale Simulation Data Sets

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Soumya Dutta, B.Tech., M.S.

Graduate Program in Computer Science and Engineering

The Ohio State University

2018

Dissertation Committee:

Prof. Han-Wei Shen, Advisor

Prof. Rephael Wenger

Prof. Jen-Ping Chen

Abstract

Recent advancements in the computing power have enabled the application scientists to design their simulation study using very high-resolution computational models. The output data from such simulations provide a plethora of information that need to be explored for enhanced understanding of the underlying phenomena. Large-scale simulations, nowadays, produce multivariate, time-varying data sets in the order of petabytes and beyond. Traditional post-processing based analysis utilizing raw data cannot be readily applicable, since storing all the data is becoming prohibitively expensive. This is because of the bottleneck stemming from output data size and I/O compared to the ever-increasing computing speed. Hence, exploration and visualization of such extreme-scale simulation outputs are posing significant challenges.

This dissertation addresses the aforementioned issues and suggests an alternative pathway by enabling *in situ* analysis, i.e., in-place analysis of data, while it still resides in supercomputer memory. We embrace the *in situ* technology and adopt simulation time data analysis, triage, and summarization using various data transformation techniques. The proposed methods process data as the simulation generates it and employ different analysis techniques to extract important data properties efficiently. However, the amount of work that can be done *in situ* is often limited in terms of time and storage since overburdening the simulation with additional computation is undesired. Furthermore, while some application domain driven analyses fit well for an *in situ* environment, a wide range of visual-analytics

tasks require longer time involving iterative exploration during post-processing. Therefore, to this end, we conduct *in situ* statistical data summarization in the form of compact probability distribution functions, which preserve essential statistical data properties and facilitate flexible and scalable post-hoc exploration.

We show that the reduced statistical data summaries can work as a replacement for the raw data and are able to perform important tasks such as feature detection, extraction, and tracking. To study the prospect of the proposed data summaries, in one application, we demonstrate that by using the statistical data summaries, complex features such as vortices, the eye of a hurricane can be extracted and tracked over time. In another scenario, we validate that, by employing statistical anomaly-based analysis using the summary data, we can detect the development of flow instabilities in a high-resolution computational fluid dynamics (CFD) simulation. We demonstrate that using the statistical distribution-based data summaries, various information-theoretic measures can be estimated which enhance multivariate time-varying feature exploration by quantifying the importance of scalar values and scalar value combinations. Furthermore, to reduce the workload of post-hoc analysis while studying the parameter sensitivity of high-resolution simulations using various input parameter settings, we devise techniques for *in situ* feature classification which significantly reduces the memory footprint in the post-hoc analysis. The proposed technique learns the feature-specific properties using a fuzzy rule-based algorithm in an initial off-line analysis stage using a known simulation data and performs feature classification *in situ* for other unknown simulation runs. For each unknown simulation, we store only a small amount of feature-specific summarized data which is visualized, compared, and analyzed interactively in the post-hoc exploration.

Dedicated to my family

Acknowledgments

First, I would like to extend my sincere gratitude to my advisor Prof. Han-Wei Shen for standing by me through the challenging years of my graduate studies and shaping my research. Without his vision, motivation, advice, and able guidance this dissertation would not have been possible. I would like to sincerely thank my dissertation committee members, Prof. Rephael Wenger and Prof. Jen-Ping Chen for providing valuable insights, suggestions, and comments which have helped me greatly to improve this thesis. I also thank Prof. Alvaro Montenegro for serving as the graduate school representative for my defense. I really appreciate their time and patience for helping me to fulfill my dream.

I must also acknowledge and thank everyone in the Data Science at Scale (DSS) team of Los Alamos National Laboratory (LANL) for allowing me to do three summer internships there. I sincerely thank Dr. James Ahrens, Dr. Jonathan Woodring, and Dr. Emily Casleton for mentoring me while I was an intern at LANL. I am grateful for the opportunities they have given me, and for their valuable suggestions which have helped to improve my research. I would also like to acknowledge all the helps I got from Dr. Boonthanome Nouanesengsy, especially for organizing the exciting summer trips which provided the fresh air needed to continue the demanding graduate life.

I feel privileged to be a part of the GRAVITY research group, led by Prof. Shen, for the past several years where I have worked and interacted with my fellow lab-mates. I have benefited immensely from all the frequent discussions among us on various research topics.

Thus, I would like to take this opportunity to thank all my lab-mates and seniors, Abon Chaudhuri, Ayan Biswas, Chun-Ming Chen, Subhashis Hazarika, Wenbin He, Xiaotong Liu, Tzu-Hsuan Wei, Ko-Chih Wang, Cheng Li, Xin Tong, Junpeng Wang, Kewei Lu, and Jiayi Xu.

I would like to take some time to extend my sincere gratitude to Prof. Bidyut B. Chaudhuri and Prof. Madhurima Chattopadhyay for giving me the opportunity to work under them during my final undergraduate year. Their guidance, constant encouragement, and help instilled the values of research in me and provided the platform to embark on my graduate studies at Ohio State. I also thank all my teachers from my undergraduate university and high schools who have helped me throughout my student life.

This dissertation would not have been possible without the endless unconditional love, care, support, and encouragement, my father Gobinda Chandra Dutta and my mother Nirmala Dutta have showered over me. I do not think that I can ever repay them for all the sacrifices they have made for me and I will be indebted to them forever. I can only look up to them and offer my deepest respect and hope that I can live up to their expectations and make them proud. This dissertation is as much a realization of my dreams as theirs.

Next, I would like to extend my love to my wife, Auditi, and thank her for all her relentless love, support, and encouragement and for putting up with me in my busy graduate life. Being a Ph.D. student herself, she has been instrumental to act as the ideal partner that I needed to get through my Ph.D. years. I consider myself truly lucky to have her in my life.

I acknowledge the much-required support that came from my elder sister Mukuta Dutta and my brother-in-law Niladri Dutta. The happy face of my niece Anisa has always made me smile, no matter how tough the situation I was in. I would also like to thank my parents-in-law, my brother-in-law Dr. Subhadip Pramanik and sister-in-law Jhuma Pramanik, and

all my family members back in India, especially my cousin brothers Pankaj Dutta and Sandip Dutta for their support, help, and concern. A big thanks to all my close friends, Ayan Acharya, Dhrubojyoti Roy, Aniket Chakrabarti, Aritra Sengupta, Swarnendu Biswas, Bortik Bandyopadhyay, Rajaditya Mukherjee, Subhro Ganguly, Shilpika Banerjee, Soham Ghosh, and Poonam Singh for always encouraging me to walk the extra mile. I have cherished all the happy memories that we created together, time and again, during my graduate life, away from home, and never felt alone. Without them, this journey would have been a dull and boring experience. Last but not the least, I thank all my seniors and other friends in Columbus, specifically Ayan Biswas, Amitava Das, and Jaideep Banerjee. All of them have become almost an extended family to me and helped me in so many ways through these years. I am really blessed to have friends and seniors like them.

I conclude this write up by acknowledging all who I could not mention within this short space.

Vita

April 13, 1987	Born, Kolkata, India.
2009	B. Tech., Electronics and Communication and Engineering, West Bengal University of Technology, Kolkata, India.
February 2010-July 2011	Assistant Systems Engineer (Trainee), Tata Consultancy Services Limited, Kolkata, India.
2011-13	Graduate Teaching Associate, The Ohio State University.
May-July 2015	Summer Research Intern, Data Science at Scale Team, Los Alamos National Laboratory.
May-August 2016	Summer Research Intern, Data Science at Scale Team, Los Alamos National Laboratory.
May-July 2017	Summer Research Intern, Data Science at Scale Team, Los Alamos National Laboratory.
May 2017	M.S., Computer Science and Engineering, The Ohio State University, Columbus, USA.
2013-Till date	Graduate Research Associate, The Ohio State University.

Publications

Research Publications

Soumya Dutta, Han-Wei Shen, and Jen-Ping Chen. In Situ Prediction Driven Feature Analysis in Jet Engine Simulations, In *IEEE Pacific Visualization (PacificVis 2018)*, April 2018.

Soumya Dutta, Xiaotong Liu, Ayan Biswas, Han-Wei Shen, and Jen-Ping Chen. Pointwise Information Guided Visual Analysis of Time-varying Multi-fields, In *SIGGRAPH Asia Symposium on Visualization*, November 2017.

Soumya Dutta, Jonathan Woodring, Han-Wei Shen, Jen-Ping Chen, and James Ahrens. Homogeneity Guided Probabilistic Data Summaries for Analysis and Visualization of Large-Scale Data Sets, In *IEEE Pacific Visualization (PacificVis 2017)*, April 2017.

Soumya Dutta, Chun-Ming Chen, Gregory Heinlein, Han-Wei Shen, and Jen-Ping Chen. In Situ Distribution Guided Analysis and Visualization of Transonic Jet Engine Simulations, In *IEEE Transactions on Visualization and Computer Graphics (IEEE Vis 2016)*, October 2016. **[IEEE SciVis 2016 Best Paper Honorable Mention]**.

Soumya Dutta and Han-Wei Shen. Distribution Driven Extraction and Tracking of Features for Time-varying Data Analysis, In *IEEE Transactions on Visualization and Computer Graphics (IEEE Vis 2015)*, October 2015.

Tzu-Hsuan Wei, Soumya Dutta, and Han-Wei Shen. Information Guided Data Sampling and Recovery using Bitmap Indexing, In *IEEE Pacific Visualization (PacificVis 2018)*, April 2018.

Gregory Heinlein, Jen-Ping Chen, Chun-Ming Chen, Soumya Dutta, and Han-Wei Shen. Statistical Anomaly Based Study of Rotating Stall in a Transonic Axial Compressor Stage, In *Turbomachinery Technical Conference and Exposition (ASME Turbo Expo, GT 2017)*, July 2017.

Subhashis Hazarika, Soumya Dutta, and Han-Wei Shen. Visualizing the Variations of Ensemble of Isosurfaces, In *IEEE Pacific Visualization (PacificVis Notes 2016)*, April 2016.

Chun-Ming Chen, Soumya Dutta, Xiaotong Liu, Gregory Heinlein, Han-Wei Shen, and Jen-Ping Chen. Visualization and Analysis of Rotating Stall for Transonic Jet Engine

Simulation, In *IEEE Transactions on Visualization and Computer Graphics (IEEE Vis 2015)*, October 2015.

Ayan Biswas, Soumya Dutta, Han-Wei Shen, and Jonathan Woodring. An information-aware framework for exploring multivariate data sets, In *IEEE Transactions on Visualization and Computer Graphics (IEEE Vis 2013)*, October 2013.

Instructional Publications

Garrett A. Aldrich, Soumya Dutta, and Jonathan Woodring. OpenMC In Situ Source Convergence Detection, In *Los Alamos National Laboratory Internal Publication (LA-UR-16-23217).*, May 2016.

Soumya Dutta, Xiaotong Liu, Ayan Biswas, Han-Wei Shen, Yifan Hu, James Giuliani, and Jen-Ping Chen. Pointwise Information Analysis for Multivariate Time-varying Feature Identification, In *OSU-CISRC-6/14-TR13.*, June 2014.

Fields of Study

Major Field: Computer Science and Engineering

Studies in:

Computer Graphics and Visualization	Prof. Han-Wei Shen
High Performance Computing	Prof. Ponnuswamy Sadayappan
Theory of Computer Science	Prof. Anastasios Sidiropoulos

Contents

List of Tables

List of Figures

Chapter 1: Introduction

1.1 Background and Motivation

The necessity of novel and efficient techniques for handling the extreme-scale data analysis challenges has become prominent in the recent years. With the ever-increasing computing capability, the size of data sets produced from various application domains has reached the order of petabytes. Soon we will enter the era of exascale computing. As a result, analysis of such extreme-scale data sets using traditional techniques are getting prolonged and overwhelming. However, timely analysis and interactive visualization of such large data are still strongly desired by the scientists so that they (1) are able to focus on the relevant and important section of the data; (2) can quickly reach to a conclusive decision; (3) can minimize their efforts in processing the data for result generation and concentrate primarily on the scientific discovery.

Modern day scientific simulations run on hundreds and thousands of cores in distributed memory environment and produce various types of complex data sets. These simulations aim at studying different types of physical phenomena such as (a) rotating stall in transonic jet engines; (b) evolution and tracking of a hurricane core or a tornado; (c) propagation of earthquake shock-waves from its epicenter; (d) formation and characteristics of eddies in oceans; (e) interaction of viscous fingers in fluid mixing etc. Due to the sheer size of these

data sets, conventional analysis and visualization methods employed on the raw data take a significant amount of time and effort for generating any analyzable result. Hence, as the size of data keeps increasing, it has become evident that traditional post-processing based analysis with raw data is not going to remain as a suitable option anymore. This is because of the bottleneck caused by the slower I/O compared to the ever-increasing computing speed and massive data sizes. To keep up with the pace and keep the data size tractable, the application scientists have been unwillingly skipping regular intervals of time steps while storing the data into disks. Even though this strategy helps to reduce the size of the data output, but, the scientists are completely oblivious to what they are missing by doing such sparse temporal sampling. Furthermore, these issues regarding big data analytics get magnified many-fold when studying data from an ensemble of simulation runs is necessary for input parameter sensitivity study, model testing, and model validation. Note that, in this case, each simulation run with a different input parameter combination will produce a separate large-scale data set. Comparative analysis of such ensemble of simulation outputs following traditional post-processing techniques will soon become intractable.

1.2 Research Problems

As the size of data grows rapidly, use of traditional techniques for data visualization is getting more and more difficult. The necessity of novel techniques, specifically tailored for handling very large data sets, has become evident. For taming the big data avalanche, therefore, data triage and summarization of important data properties are going to pave the path forward. Since a significant amount of data reduction is essential for allowing a scalable post-hoc analysis, the questions are now (a) how should we summarize the data? and (b) what should we store such that (1) the summarized data is compact and significantly

reduced in size; (2) preserves important characteristics of the data as much as possible; (3) post-processing based analysis and visualizations still remain flexible and scalable; and (4) uncertainty quantification using the reduced data is possible for conveying the trustworthiness of the generated results to the users.

The capability of running the simulations using high-resolution and high-precision physical models has enabled the application scientists to simulate non-trivial scientific phenomena with great detail. As a result, even though the accuracy of the simulations increases, the complexity of the scientific data sets is also increased significantly with the size. For example, a study using a recently developed ocean model MPAS-O [158] showed that by increasing the resolution of the mesh new eddies in the ocean are captured. These new eddies conform with the observed data and the simulation only produces them accurately with very high-resolution meshes. Therefore, efficient detection and exploration of such complex scientific features over time are also posing new challenges for the application scientists. Besides this, owing to the very complexity of the scientific phenomena being studied, a precise descriptor of features (the regions of interest (ROI)) is often unavailable. As a result, features such as the eye of a storm, circulating vortex cores in a flow field, rapidly propagating earthquake shock-waves, rotating stall cells in jet engines are difficult to be separated by hard threshold values. Therefore, scientists need novel visual-analytics systems where they can directly interact with the data and locate the feature of interest based on their initial vague hypotheses. However, repeating this process manually for extreme-scale time-varying data sets is both tedious and impractical.

Oftentimes, to obtain a detailed knowledge about the behavior of the simulation models under different input conditions, application scientists study the model output under various input parameter settings which require them to run an ensemble of simulations. This type of

study is particularly essential when the sensitivities of some important model parameters are not entirely known and can have a significant impact on the model output. For example, the knowledge about the impact of the throttle parameter in the simulation of a transonic jet engine is critical for the safe operation of the engine. This is because some values of the throttle parameter can make the engine unstable and cause permanent damage due to the inception of rotating stall. In such cases, the scientists run the simulation multiple times with different input throttle values and then conduct comparative studies on the important data features extracted from individual simulation outputs for drawing a conclusion. However, since each simulation produces a large-scale data set of its own, storing all the raw data from all the simulations for off-line exploration will soon become prohibitive.

Besides identifying important regions of interest in the data sets, the experts also search for data values and data value combinations of multiple variables which show positive or negative association for getting an insightful understanding of the characteristics of the features. Hence, quantification of the importance of individual data values, and data value combinations of multiple variables has gained importance in the recent years. Study of multi-variables based on their value combinations allows researchers to understand how the total shared information among variables is distributed within all of its values. It enables users to identify salient value combinations and predict their behavior over space and time. Traditional correlation studies on variables only provide the knowledge about their high-level interactions, but, a guideline to study the relationships of specific value combinations in time-varying multi-field data sets is still missing.

1.3 Proposed Solutions

To address the aforementioned issues, we propose a new data exploration pipeline based on a technology called ***in situ* analysis**, i.e., in-place analysis of data, while it is being produced and resides in computer memory. Compared to the traditional post-processing based strategies where first the raw data is stored into disks and then later used for exploration, *in situ* analysis has the key advantage of being able to touch the full resolution raw data without even storing it into disks. As a result, important data features can be efficiently extracted and summarized into a compact format and stored instead of keeping the full resolution raw data. By performing *in situ* data processing, expensive data movement can be reduced significantly while maximizing the resource utilization. This can decrease the effort in post-hoc analysis phase by keeping the memory footprint and storage low. All of these benefits come with some obvious restrictions. Since the analysis code is run along with the simulation, it is expected to scale with the simulation without consuming too much additional memory. Also, the computational overhead of the *in situ* data processing is always desired to be a small fraction of the actual simulation time without overburdening it. As a result, a wide range of scientific exploration tasks that require hypothesis generation, validation, and verification involving iterative processing with expert-in-the-loop, cannot be performed entirely in the *in situ* environment. Thus, data triage and summarization during the *in situ* data processing are going to be imperative to make the post-processing based analysis and visualization flexible, scalable, and pragmatic.

In this dissertation, we propose various novel *in situ* data summarization strategies to cater to the needs of the wide range of scientific applications. When the experts want to study a specific feature in the simulation data, we propose strategies which will detect such features directly in the *in situ* environment and output only the reduced feature-specific summarized

information, which can be analyzed during the post-hoc analysis. This scheme is essential when the application scientists want to study the impact of different input parameters on the simulation output for model validation and require running the simulation multiple times with different parameter combinations. In such cases, instead of storing raw data for each of the simulations, we identify the important features *in situ* and only output the feature-specific small amount of data which is compared and contrasted during the post-hoc analysis. By doing so, we can analyze and visualize the results almost instantly as soon as the simulation is finished. However, since we store only the feature-specific information, this type of summarization is not suitable for answering broader queries which are not related to those specific features. Therefore, even though this feature-specific summarization scheme can reduce the post-hoc exploration time significantly, there is a need of a more general purpose data summarization scheme which can facilitate other data analysis and visualization tasks and still keep the post-processing tractable. To address this, in this thesis, we demonstrate that the statistical data summarization in the form of probability distributions can facilitate exploratory post-hoc data analysis and visualization. Analysis using probability distributions is becoming a promising choice in the visualization community in recent years, and many visualization problems have benefited from such stochastic approaches [29,52,74,89,94,151]. In our work, we demonstrate that a wide range of visual exploration tasks can be performed using the distribution-based summary data in the post-processing phase. To demonstrate the effectiveness of the proposed data summaries, we develop new algorithms based on statistical sampling and data partitioning schemes that can reconstruct the full resolution data from the statistical summary data with minimal errors. We use distributions to model both the local and global statistical properties of the data and analyze the spatiotemporal distributions to perform feature detection, analysis, and tracking.

1.3.1 *In situ* Feature Prediction and Summarization for Exploratory Data Analysis

To obtain detailed knowledge about the behavior of the models under different input conditions, experts often study the model output under various parameter configurations which require them to run an ensemble of simulations. While studying such simulation outputs containing complex events like the evolution of rotating stall in jet engine compressors, the formation of viscous fingers in a mixing fluid etc., scientists rely upon feature-specific information derived from the simulation output. Efficient extraction and exploration of such information demand (a) Understanding about the target features; (b) Effective methods for robust and scalable detection of such features; and (c) Summarization and interactive exploration capability through appropriate visual encodings. However, as the complexity of the data sets is growing, several previous works in this context have acknowledged that defining features with precise descriptors is becoming increasingly difficult due to the intricate nature of the simulated phenomenon [14, 103]. Therefore, in our work, we introduce an analysis technique that enables *in situ* characterization of imprecisely defined features using a knowledge-base constructed in an off-line learning phase. Initially, we take input from the domain expert who investigates a known data to highlight regions of importance directly from the data without any precise feature descriptor. By employing a fuzzy pattern learning algorithm, we capture the multivariate relationship of several variables from the expert-labeled sample points. After learning, the prediction of features under unknown and new parameter settings takes place *in situ*. We show that, through *in situ* feature prediction, we bypass the expensive I/O and output only feature-specific data summaries which allow interactive and timely post-hoc analysis. Details of this technique are provided in Chapter 3. Note that, in this case, only the information regarding the target feature/event of interest

is kept for post-hoc analysis and the majority of the data processing and analysis work are done *in situ*. However, when the experts do not have a predetermined set of tasks and will need longer time for hypothesis verification and feature discovery, the aforementioned pure *in situ* analysis based feature classification strategy may not be suitable. For such cases, we propose a more general purpose strategy discussed in the following.

1.3.2 *In situ* Local Distribution-based Compact Data Summaries for Extreme-scale Visual-analytics

To deal with the extreme-scale data sets, where purely post-processing based exploration using full resolution raw data is too expensive and performing the same analysis tasks *in situ* will overburden the simulation, we propose new *in situ* data triage and summarization schemes for making the post-hoc analysis and visualization flexible and scalable. In this dissertation, we demonstrate that by keeping only the local region-based statistical data distributions as a replacement of the raw data, a wide range of analysis and visualization tasks can be performed. Here we have to consider two important aspects of creating such local region-based distributions. **First**, since the size of the data is very large, we aim at a compact representation of distributions with a fast computation time and small memory footprint. Popular distribution estimator histogram computes the distributions quickly, but its storage requirement makes it unsuitable for our work. Another non-parametric estimator Kernel Density Estimation (KDE) also requires higher storage and is also computationally expensive. Parametric model Mixture of Gaussians (GMM), in this context, presents a suitable choice for modeling the data distributions. Use of GMMs [13] is well known for data classification [89, 103, 149] and we show that the GMMs can be computed *in situ* efficiently by coupling the computation with the simulation directly. Furthermore, based on the Gaussian properties, GMMs allow efficient computation by directly using the mixture

components [136]. **Second**, local regions created from the partitioning of the data ideally should be as coherent as possible with respect to their data values. This data coherency in the local regions will ensure a more accurate and less uncertain statistical summarization of the regions using distributions because the variation of data values inside each region will be reduced. Therefore, a comprehensive study for finding suitable and fast data decomposition schemes for the local region-based statistical data summarization is performed. We show that storing data in this summarized representation enables flexible post-hoc analysis with the much-needed capability of uncertainty quantification [23, 94, 112–114]. The details of different distribution-based data summarization schemes and various data partitioning techniques used in our work are discussed in Chapter 4.

1.3.2.1 Distribution Data Driven Flow Instability Detection in Extreme-scale Fluid Simulation Data

For demonstrating the applicability of our proposed distribution-based data summarization methods, in this thesis, we depict the successful application of our technique for detecting flow instabilities in a very large-scale computational fluid dynamics (CFD) simulation. By performing analysis only using the local region-based distribution data, which was generated *in situ*, we show that our proposed method can detect the inception and development of rotating stall phenomenon in the transonic turbine compressors with high precision. The phenomenon of rotating stall initiates from local airflow disturbances among the engine compressor blades but grows quickly to become destructive to the engine. It is challenging to predict this event since the signs of stall inception are subtle and non-trivial to be detected robustly. We use a state-of-the-art CFD simulation which is capable of accurately modeling the behavior of transonic compressors throughout their operational range, i.e., choke to stall. However, the computational cost and the amount of data produced is quite significant.

Traditional post-processing analysis utilizing raw data cannot be readily applicable since storing all the raw data is not a viable option. Through *in situ* statistical data summarization in the form of distributions, we show that flexible and exploratory post-processing based stall analysis is achievable. We exploit the variation in region-based distributions over space and time and detect statistically anomalous regions which show early signs of rotating stall. For validating the suspected locations in the spatial domain, we utilize uncertain isocontour algorithms. Positive feedback from the expert confirms the efficacy and benefits of the proposed method and also demonstrates the capability of *in situ* processing in analyzing extreme-scale data sets in an effective way. Details of this work are discussed in Chapter 5.

1.3.2.2 Distribution Data Driven Extraction and Tracking of Vaguely Defined Features in Large-scale Time-varying Data Sets

We present another use of our proposed data summaries in tracking vaguely defined features in large time-varying data sets with a novel confidence-based feature extraction and tracking algorithm. As the scientific data sets are getting more and more complicated, obtaining precise descriptors of features in such data sets is becoming a non-trivial task for the scientists. A majority of the feature tracking algorithms proposed in the past have a general assumption that the definition of the feature is predetermined and hence the feature extraction process is deterministic. Therefore, given only a vague feature description, automatic detection and tracking of such regions require novel algorithmic approaches. In this work, we show that by using probability distribution functions estimated from a user highlighted region from an initial time step, as a measure of the feature, we can efficiently extract and track such features using only our distribution-based summary data consistently over time. Using the distribution information, we generate a feature-aware classification field for each time step where regions with higher values signify the higher possibility of

containing the user defined feature. Applying a threshold on the possibility values based on user's requirement, we are able to segment the classification field and focus on the feature of interest. Finally, we employ a distance based tracking algorithm which identifies the target features over consecutive time steps. We provide the detailed description of our distribution-data based tracking algorithm in Chapter 6.

1.3.3 Global Distribution-based Data Models for Exploration of Time-varying Multi-fields using Information Theoretic Measures

In the above sections, we proposed the applicability of local region-wise distribution-based data models as an efficient data summarization scheme to enable exploration of extreme-scale data sets. In this section, we discuss the use of global data distributions for conducting statistical association analysis in time-varying multi-field data sets. Use of information theoretic measures [38] for quantifying the importance of variables and scalar values has proven to be useful in the past [15, 21, 57, 67, 141, 146]. Since the problem of interest in this context is to study the associativity/correlation of scalar values and the combination of scalar values of multiple variables over the data domain, we employ global distributions of multi-fields for estimating their behavioral characteristics over the entire data domain. A key advantage of the probabilistic data models is that various information theoretic measures can be directly estimated from such distribution-based data summaries efficiently. For identifying important variables and quantifying their interaction over space and time, we use mutual information and two of its decomposition in this work. For allowing a refined and lower-level multivariate analysis involving scalar value interactions of multiple variables, we propose a method based pointwise mutual information (PMI) for detecting the amount of information sharing between different scalar value combinations. Study of multi-variables based on their value combinations allows researchers to understand how the

total shared information among variables is distributed within all of its value combinations. It enables the experts to identify salient scalar value combinations and predict their behavior over space and time. A detailed discussion of this work is presented in Chapter 7.

In summary, this dissertation is organized as follows: In Chapter 2, we discuss the relevant previous research works. Chapter 3 provides the details of the proposed *in situ* feature classification and analysis technique. Chapter 4 presents various distribution-based data summarization techniques in detail. In this chapter, we also discuss the effectiveness of different data partitioning schemes for creating local partition-based statistical data summarizations. Applications of the proposed distribution-based summary data are demonstrated in Chapter 5 - 7. Finally, we conclude this dissertation in Chapter 8 and discuss several future research scopes.

Chapter 2: Background and Related Work

In this chapter, we discuss the previous research works on several topics such as *in situ* data visualization, statistical methods in visualization, multi-variate data exploration etc. which are relevant to this dissertation and have influenced our work. In the beginning, Section 2.1 covers previous works in the area of *in situ* visualization. The background on statistical and distribution-based visual analysis is presented in Section 2.2. These works demonstrate the usefulness of data distributions in analysis and visualization of various types of scientific data sets. Next, in Section 2.3 we provide the relevant works that have benefited from using predictive algorithms. Existing feature tracking techniques have been summarized in Section 2.4. The related works in the area of multivariate time-varying data analysis and visualization are discussed next in Section 2.5. Finally, we conclude this chapter in Section 2.6 by covering the body of research work that has used various information theoretic measures extensively for effective visual analysis of complex scientific simulation data sets.

2.1 *In Situ* Data Processing, Analysis, and Visualization

The necessity of *in situ* analysis is becoming more and more prominent, as the size of data output from high-resolution simulations is out-pacing post-processing and visualization capabilities. One of the early attempts of *in situ* visualization was made by Haimes [58] to

visualize large unsteady data sets. For enabling *in situ* capability in Paraview, Fabian et al. proposed CATALYST library [46]. Similarly, run-time visualization with LibSim using VisIt was introduced by Whitlock et al. [155], and in another work, Lofstead et al. added ADIOS as an *in situ* visualization framework [91]. Vishwanath et al. enhanced simulation time data analysis by proposing GLEAN [142]. An *in situ* sort-and-B-spline error-bounded lossy data compression scheme, ISABELA, was proposed by Lakshminarasimhan et al. in their work [82]. Later, *in situ* multi-resolution data compression support was added to ISABELA by Lehmann and Jung in [85]. A zero-copy data structure was introduced by Woodring et al. [159]. *In situ* eddy analysis in ocean simulation models was demonstrated by Woodring et al. [158]. Yu et al. enabled high-quality *in situ* visualization of combustion data in their work [165]. A study on the greenness (i.e., power, energy, and energy efficiency) of *in situ* techniques compared to its post-processing counterparts done by Adhinarayanan et al. can be found in [2] which reports the percentage of energy that can be saved by performing *in situ* analysis.

However, visualization tasks which require exploratory data analysis, verification, and validation by adding experts in the loop cannot be done using traditional pure *in situ* approaches. A flexible post-hoc analysis will still be needed for scientific discovery. Hence, a new *in situ* processing based paradigm is emerging where the task-specific important simulation data is massively reduced via *in situ* processing, and flexible scalable post-hoc analysis is done on the reduced data [34]. The visualization community has begun to embrace this new paradigm and has proposed several such schemes [86, 144]. A sampling-based method for interactive visualization of cosmology data was used by Woodring et al. [157]. Ahrens et al. adopted an *in situ* image-based approach called Cinema [3] for

feature exploration during post-hoc analysis. For a more comprehensive review on *in situ* data visualization, please refer to the state-of-the-art report by Bauer et al. [9].

2.2 Statistical Methods and Distribution-based Visualization

Statistical analysis methods for data exploration and visualization has numerous applications in visualization community. Use of distribution-based methods for exploring scientific data sets has become an emerging trend in the visualization domain. For visualizing spatial distribution data sets, Kao et al. [76], Luo et al. [94] and Potter et al. [114] visualized distribution data sets by displaying statistical summaries such as means, standard deviations and skews in color, height field, or glyphs. Potter et al. [113] utilized summary plots which enhance box plots with moments and histograms in a higher dimension. Kniss et al. proposed statistically salient volume data visualization [79]. Multivariate data exploration was conducted using local statistical data complexity by Jänicke et al. [68]. A study of Non-parametric distribution models and their applicability was discussed in [111]. A fuzzy matching based feature extraction method was proposed by Johnson and Huang [74]. Efficient range distribution query algorithms using integral histograms [29] and wavelet transforms [84] yielded valuable statistical information from data. For analyzing FTLE in distribution data sets, Guo et al. [55] used distribution-based data models for uncertainty quantification. Recently, use of GMM-based feature analysis has received increasing attention due to its compact representation capability and its close relation to clustering. Wang et al. [149] utilized GMMs for transfer function design in time-varying data sets. Liu et al. [89] exploited GMMs for stochastic sampling-based volume rendering on the GPU.

Use of local region-based distribution models has gained increased attention in recent years. The local region-based distribution models have shown to be capable of preserving

statistical properties of the data compactly and the reduced distribution data summaries can be used efficiently for analyzing and visualizing large-scale scientific data sets. For designing transfer functions, Lundstrom et al. [93] used local histograms. Wei et al. [151] presented efficient local histogram search using bitmap indexing for feature analysis. For large data summarization, Thompson et al. [135] made use of distribution-based of hixels, which stored a histogram per data block to preserve uncertainty information due to data down-sampling. A regular block-wise approach was taken by Gu and Wang for a graph-based analysis of time-varying data [53].

In the area of uncertainty analysis in visualization, Brodie et al. [23] and Bonneau et al. [16] recently provided thorough reviews. For visualizing spatial distribution data sets, Kao et al. [76], Luo et al. [94], and Potter et al. [114] used 2D distribution data sets by displaying statistical summaries such as means, standard deviations and skews in color, height field, or glyphs. Potter et al. [113] proposed summary plots which extend box plots with moments and histograms in higher dimension visualizations. To visualize 3D distribution data sets, flickering the color according to the distribution samples is used in volume rendering [89,135]. Uncertain isosurface extraction provides a further understanding of data at a specific isovalue. Pöthkow et al. [110] computed the level crossing probability of adjacent points, which was extended for computing cell-wise level crossing probability [112]. Athawale et al. [5] devised closed-form computation of level-crossing probabilities for nonparametric distribution data sets.

2.3 Predictive Feature Exploration in Visualization

Prediction-based techniques for feature exploration and visualization under uncertainty have been shown quite effective in many previous visualization works. Since the features

in scientific simulations are getting more and more intricate in nature, feature extraction using hard threshold values or fixed data value ranges are becoming difficult. As a result, visualization experts have started using uncertainty-aware predictive techniques for exploring data features. A predictor-corrector based approach for detecting feature correspondence during feature tracking was proposed in [100]. Interactive predictive parameter space analysis was demonstrated in the works of Berger et al. [10] where the technique provided guidance to the users for locating interesting parameter regions. Prediction of uncertainty in volume visualization pipeline was obtained using a possibility-based approach [49]. Similar to our approach, various rule-based techniques were used by many researchers for performing predictive feature analysis. A Rule guided feature tracking using inductive learning was demonstrated by Banerjee et al. [8]. A framework for hypothesis generation and verification using a fuzzy logic based learning approach was introduced by Fuchs et al. [50]. A fuzzy rule-based efficient approach for tracking molecular particles was shown by Jiang et al. [73]. A fuzzy set based possibilistic marching cubes algorithm was proposed by He et al. in [61]. Automatic error controlling during volume rendering was achieved using a fuzzy controller [87].

More recently, use of machine learning techniques in visual analytics applications are emerging as one of the promising paths forward for the researchers [96]. A majority consensus-based vortex analysis framework was proposed by Biswas et al. [14]. Zhang et al. [166] used AdaBoost algorithm for improving vortex extraction. Kaumpf et al., in a recent work, visualized the confidence in the clustering process while analyzing data [81]. A detailed study on the effects of dimensionality reduction techniques and clustering on high dimensional data was done by Wenskovitch et al. [153]. For sophisticated higher-dimensional classification of volume data sets, Tzeng et al. [137] utilized neural network

based techniques. Comparison of visualization-driven data labeling schemes and machine learning guided data labeling techniques was demonstrated by Bernard et al. [11].

2.4 Feature Extraction and Tracking

Feature extraction and tracking is an important problem for scientific data visualization. For tracking volumetric features in scientific data, Samataney et al. [123] proposed a correspondence based approach. By exploiting volume overlapping, Silver and wang [130] tracked volume features with high accuracy as well. Ji and Shen used earth mover's distance to design a globally optimum feature tracking algorithm [72]. In another work, Ji et al. used higher dimensional isosurfaces for tracking volume features [71]. For tracking features in distributed AMR data sets, Chen et al. used feature tree as a visualization representation of tracked features [31]. Tzeng and Ma [138] proposed a machine learning approach for automatically learning and tracking features in large-scale simulation data. Ozer et al. recently proposed techniques for tracking a group dynamic features together as a collection, where the problem of tracking was modeled as activities in scientific data [104, 105]. Using a predictor-corrector method, Muelder and Ma introduced a new algorithm for efficient feature tracking [101]. To quantify goodness in feature correspondence, Reinder et al. introduced an attribute-based feature tracking algorithm for scientific data sets [117]. In a recent work, Sauer et al. utilized particle information for enhanced feature extraction and tracking in joint particle/volume data sets [125]. Their method allowed to track features in data sets when sufficiently dense temporal sampling is not available. A TAC based distance field was used effectively in the works of Lee and Shen for analyzing time-varying features [83]. Using the knowledge from all the time steps, a global feature tracking technique based on merge tree comparison was proposed by Saikia and Weinkauf in [122]. Theisel and Seidel proposed

a method for tracking features like a saddle, source, and sinks in time-varying vector field directly using streamlines [134]. Garth et al. in another work presented techniques for tracking vector field singularities [51]. A survey of feature tracking algorithms also can be found in [109] by Post et al.

2.5 Multivariate Time-varying Data Analytics

Multivariate data analysis is a well-researched area for its applicability in a wide range of problems. As presented by Oliveira et al. [42] in their survey of visual data exploration, multivariate data analysis can be broadly subdivided into four categories: geometric projection, pixel-oriented techniques, hierarchical display, and applications of iconography. Wong and Bergeron presented an extensive survey of multivariate visualization [156]. Jänicke et al. [65] performed the analysis of multivariate data by transforming high dimensional data into more tractable 2D space. A high dimensional brushing for the multivariate data sets was proposed by Martin and Ward [98] which enabled user interaction in the exploration process. A Nugget Management System (NMS) was proposed by Yang et al. [162] where the nuggets are referred to as the data that are of interest to the user. Bramon et al. used specific mutual information measures for multi-modal data fusion [19].

Exploration of time-varying data sets to extract features has been a challenging task, and researchers have investigated this in the past. Woodring et al. [160] used wavelet transform on the time-varying data to generate a series of curves and cluster them to find time-varying trends. Akiba et al. [4] used time histogram for designing transfer functions to reveal features of time-varying data and provide feedback for simultaneous rendering. Younesy et al. [164] proposed the differential time histogram which allowed for efficient handling of queries for large time-varying data and also provided an error bound on the data visualization.

Jänicke et al. [75] used the density-driven Voronoi tessellation for dividing the unsteady data into classes that show different behaviors. Wang et al. [147] used information theory to extract importance curves from the blocks of data which allowed effective classification and visualization of features. In a recent work by Wang et al. [145], the authors used transfer entropy for selection of variables and visualize the information transfer that revealed the causal relationships in the data.

2.6 Information Theory in Visualization

Information theory [38] has been used extensively for solving many different problems in visualization. For estimating various information theoretic measures, data distributions are computed first and then different probability values are used to estimate the information theory metrics. Information theoretic measures such as entropy, mutual information have found widespread use in visualization community. For polygonal scenes, Vázquez et al. [140] used the viewpoint entropy for automatic computation of good viewing positions. Borodoloi and Shen [17] used entropy for selecting informative voxels to construct a view selection algorithm for volume rendering. Viola et al. [141] proposed another information theoretic approach to select the most expressive viewpoint by maximizing the mutual information. Information theoretic approaches have also been used for light source placement for varying camera positions [54] and analysis of scene visibility and radiosity complexity [47]. Jänicke and Scheuermann [66] used information theory for the analysis of unsteady flow features by using ε-machine, a highly compressed abstract representation of the flow, where the causal states depict different dynamics within the data and the edges between them represent different transitions and likelihood.

In Information theory, mutual information quantifies the amount of reduction in the uncertainty of one random variable while observing another random variable. It can also be interpreted as the information overlap or correlation between two random variables. The applicability of mutual information in visualization can be broadly classified into 5 categories (1) multi-modal registration and fusion; (2) transfer function design and volume exploration; (3) streamline selection; (4) isosurface analysis; and (5) other applications. Firstly, in the field of medical visualization, mutual information has been extensively used for registration and fusion of 3D data sets [37, 59, 62, 63, 97, 107]. Second category is in the context of volume exploration and transfer function designing where mutual information has been used previously. Given a feature, identification of the best view and its smooth transition are achieved in [141] by using mutual information. By calculating mutual information between the intensity values of data and color pixels, Bramon et al. [22] construct an observation channel which is used to measure the information transfer between input data values and output pixels. Recently researchers have successfully applied mutual information to analyze flow field features which is the third set of applications. They use mutual information for the selection of streamlines based on coherent view direction [95, 133]. The fourth class of application of mutual information is isosurface analysis. Bruckner et al. [24] have proposed a similarity based exploration of isosurfaces of univariate data using mutual information. They create the isosurface similarity maps where mutual information is used to measure the similarity among different isosurfaces. A level-set based method has been presented in [152] to analyze the representativeness of an isosurface of a volumetric data. Finally, mutual information has also been used for analyzing scene complexity [47], shape complexity [118] and viewpoint selection and mesh saliency analysis [48].

A majority of the aforementioned works use mutual information to measure the information shared between variables which highlight the average interaction between variables since mutual information only gives a single value as a measure of shared information between two variables. Even though [24, 152] have tried to analyze isosurfaces and their similarities, their works deal with univariate data sets. Haidacher et al. [56] have extended the analysis in the multi-modal domain for analyzing multi-modal surface similarities. To facilitate analysis of multivariate data sets based on their scalar values, researchers have sought after specific mutual information (SMI) for its ability to quantify information of a scalar value given another variable. Here, we classify the use of specific mutual information in three broad classes (1) multi-modal data fusion, (2) selection of isosurfaces based on multivariate relations, and (3) multi-modal transfer function design. In the first category, Bramon et al. [19] use the specific mutual information to fuse multi-modal data sets. Recent works of Biswas et al. [15] fall under the second category where a framework is presented to create an information map based on specific mutual information, which can be used to select isovalues of one variable which are either predictable or uncertain with respect to the other variable. In the final category, specific mutual information based transfer function design is introduced [20].

Chapter 3: *In Situ* Feature Classification and Summarization for Exploratory Data Analysis

Efficient feature exploration in large-scale data sets using traditional post-hoc analysis approaches is becoming prohibitive due to the bottleneck stemming from I/O and output data sizes. This problem becomes more challenging when an ensemble of simulations are required to run for studying the influence of input parameters on the model output. As a result, scientists are inclining more towards analyzing the data *in situ* while it resides in the memory. In this work, we study the evolution of rotating stall in jet engines using data generated from a large-scale flow simulation under various input conditions. Since the features of interest lack a precise descriptor, we adopt a fuzzy rule-based learning algorithm for efficient and robust extraction of such features. For scalable exploration, we advocate for an off-line learning and *in situ* prediction driven strategy that facilitates in-depth study of the stall. Task-specific information estimated *in situ* is visualized interactively during the post-hoc analysis which reveals important details about the inception and evolution of stall.

Working with a simulation, TURBO [32], a state-of-the-art Navier-Stokes based Computational Fluid Dynamics (CFD) code developed in NASA, as a specific use case, we study the evolution of rotating stall in jet engines under different parameter settings. Rotating stall is characterized by local airflow disturbances, which grow rapidly and become destructive to the engine. So, for a stable engine operation, experts employ a safe throttle setting which is

modeled by the parameter *corrected mass flow rate (CMF)* in TURBO. By changing it, the performance of the engine can be improved by minimizing the aerodynamic loss. However, as the best possible operating range of CMF is currently unknown, the experts want to study the impact of CMF in stall inception. Note that, a traditional post-hoc analysis with raw data [30] will not scale, because just a single run of TURBO generates data in the order of tens of TBs. Besides this, the feature of interest, i.e., the stall cell lacks precise descriptor making the problem even more challenging. Therefore, the expert wants to identify the stalled regions using multiple variables to make the detection robust.

To address the aforementioned issues, we introduce an analysis technique for *in situ* detection of imprecisely defined features. The domain expert initially investigates a known simulation data to highlight stalled and stable regions interactively. Then by applying a fuzzy pattern learning algorithm, we capture the multivariate relationship of several important variables from the expert-labeled sample points. The relationship is modeled using a fuzzy rule-based system allowing us to translate the knowledge of the expert into the knowledge-base of an intelligent system. After learning, prediction of stall under different parameter settings takes place *in situ* using the inference algorithm of the fuzzy rule-based system. Besides this, since the stalled regions act as blockages to the normal airflow, it is hypothesized that the stalled passages will demonstrate lower mass flow rate compared to the healthy passages. However, as the traditionally used global mass flow rate is just a single aggregated value, it can only detect whether the stall has happened, and the confirmation of stall from the global mass flow rate comes quite late. So, along with the fuzzy system guided stall detection, we also conduct a passage-wise local mass flow rate based stall analysis to verify its effectiveness. We show that, with our proposed strategy of off-line learning and *in situ* feature prediction, the expert can study the influence of parameters like CMF on the

output in a timely manner with minimal effort. Through *in situ* analysis, we are able to cut down the expensive I/O and output only derived feature-specific information which allows interactive post-hoc analysis. So, our contributions are threefold:

1. We introduce an off-line learning and *in situ* prediction based analysis strategy which provides a fuzzy learning based solution for exploring features in very large-scale simulations.

2. We successfully demonstrate an approach of *in situ* detection of imprecisely defined features and show its applicability using a real-world flow simulation TURBO for studying the evolution of rotating stall in transonic jet engines under varying input parameter configurations.

3. We develop visual-analytics systems, which on one hand allow experts to interactively specify their region of interest directly from the data, and on the other hand, facilitate effective investigation of the feature-specific data, derived *in situ*, for simulation profiling.

3.1 Background, Domain Requirements, and Overview

3.1.1 Application Background

Understanding the evolution of rotating stall is critical for the aerodynamics scientists to prevent permanent engine failure. To gain insights about rotating stall, a numerical CFD simulation code TURBO has been developed at NASA which can model stall with high accuracy. The rotor in this configuration consists of 36 blade passages, as shown in left sub-image of Figure 3.1. In order to view the structure of the passages a zoomed in view of a subset of the full rotor is depicted on the right sub-image of Figure 3.1. In this figure we

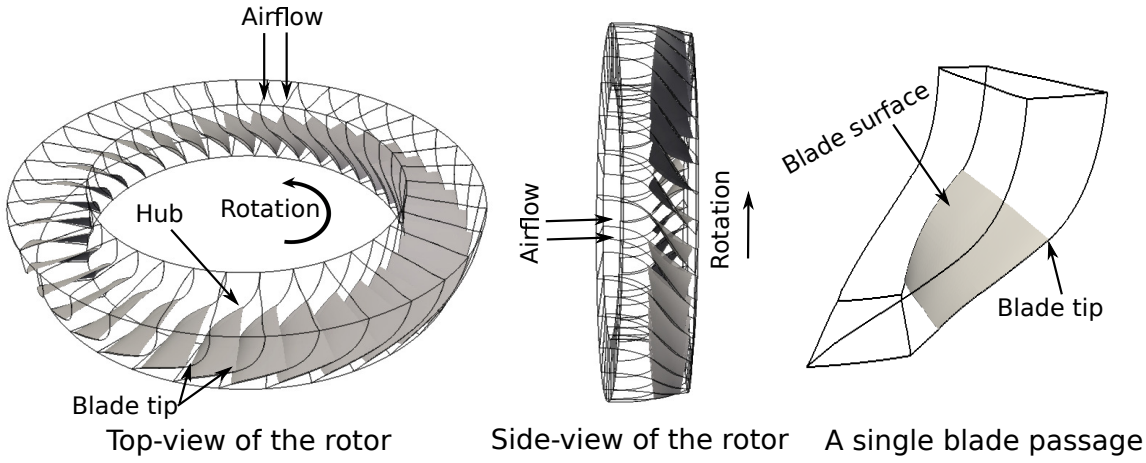


Figure 3.1: A schematic diagram of the rotor structure of the compressor stage.

highlight the different domain specific components of a blade passage. Using data generated from TURBO [32], experts want to investigate various conditions which may initiate stall. This mandates a comprehensive ensemble study within a parameter space consisting of the various flow controlling and blade designing parameters. Among them, engine's throttle setting, modeled by the parameter CMF, is a prime candidate for investigation. By setting a proper CMF value the performance of a compressor can be boosted with increased fuel efficiency as well as improved reliability. Unfortunately, certain CMF values can also trigger instability. Hence, a detailed understanding of the impact of CMF in stall inception will allow scientists to push the performance limit while still maintaining the engine safety. The benefits of improved understanding of stall could yield compressor designs that can operate closer to their maximum efficiency. In this context, it is to be noted that, existing visual analytics guided stall exploration strategies [30, 32] that utilize raw data and follow a post-hoc approach are not easily applicable in our scenario as they do not scale due to (a) I/O bottleneck; and (b) the large size of the output data. So, running TURBO multiple

times with different CMF values will produce data sets which ideally can only be analyzed cost-effectively in an *in situ* environment. However, the spatial anomaly-based analysis in [30], if done *in situ*, will require a large amount of additional data communication, which will slow down the simulation.

3.1.2 Domain Specific Requirements

Given the aforementioned limitations of existing methods, we have identified the following requirements:

1. The expert first wants to know if the current value of CMF has led the simulation to a stalled condition. If it has resulted in a stall then what time step ranges have started showing signs of a stall and how the phenomenon has evolved over time?

2. Which part of the rotor is affected by the stall and how the stalled regions span in the spatial domain?

3. How can multiple variables together be used instead of a single variable to detect the stalled regions more reliably?

4. Can we analyze and visualize the derived information about the simulation and make decisions such as whether to run the simulation for a longer time? This will facilitate efficient use of the CPU cycles when the study needs to be performed under a constrained resource budget.

3.1.3 Overview of the Proposed Approach

Figure 3.2 shows a schematic view of the proposed analysis pathway. Acknowledging the fact that the features of interest are non-trivial to be defined by specific descriptors, we

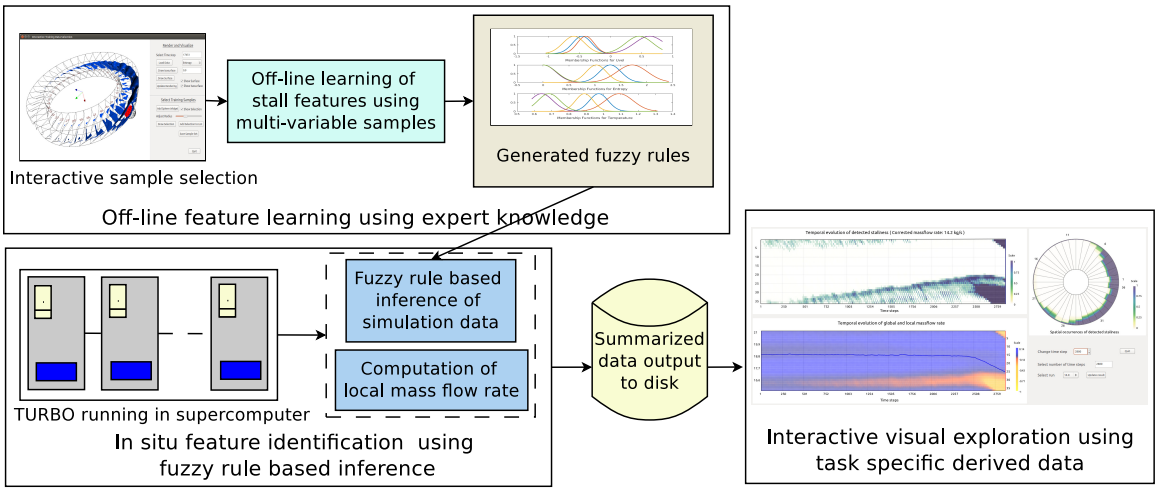


Figure 3.2: A schematic diagram of the proposed analysis method. Our off-line learning and *in situ* prediction based analysis strategy enables the study of the evolution of features in large-scale data sets in an effective and timely manner.

provide a visual-analytics tool through which the expert can explore the data and directly select their target region. By collecting the sample points marked by the expert, we construct a sample set where for each sample point we measure three variable values. We employ a fuzzy rule-based pattern learning algorithm to learn the multivariate pattern from the sample set. The fuzzy rule base is then employed *in situ* for predicting the stall when the simulation is run with unknown parameter settings. The fuzzy system assigns a classification score to each point which represents the *stallness* of that point. Besides this, we also compute the local mass flow rate of each blade passage. The task-specific information about stallness and local mass flow rate are stored into disks which are significantly smaller in size compared to the raw data. Finally, the *in situ* derived information is explored post-hoc using an interactive visualization that enables profiling the simulation and reveals important details about stall inception.

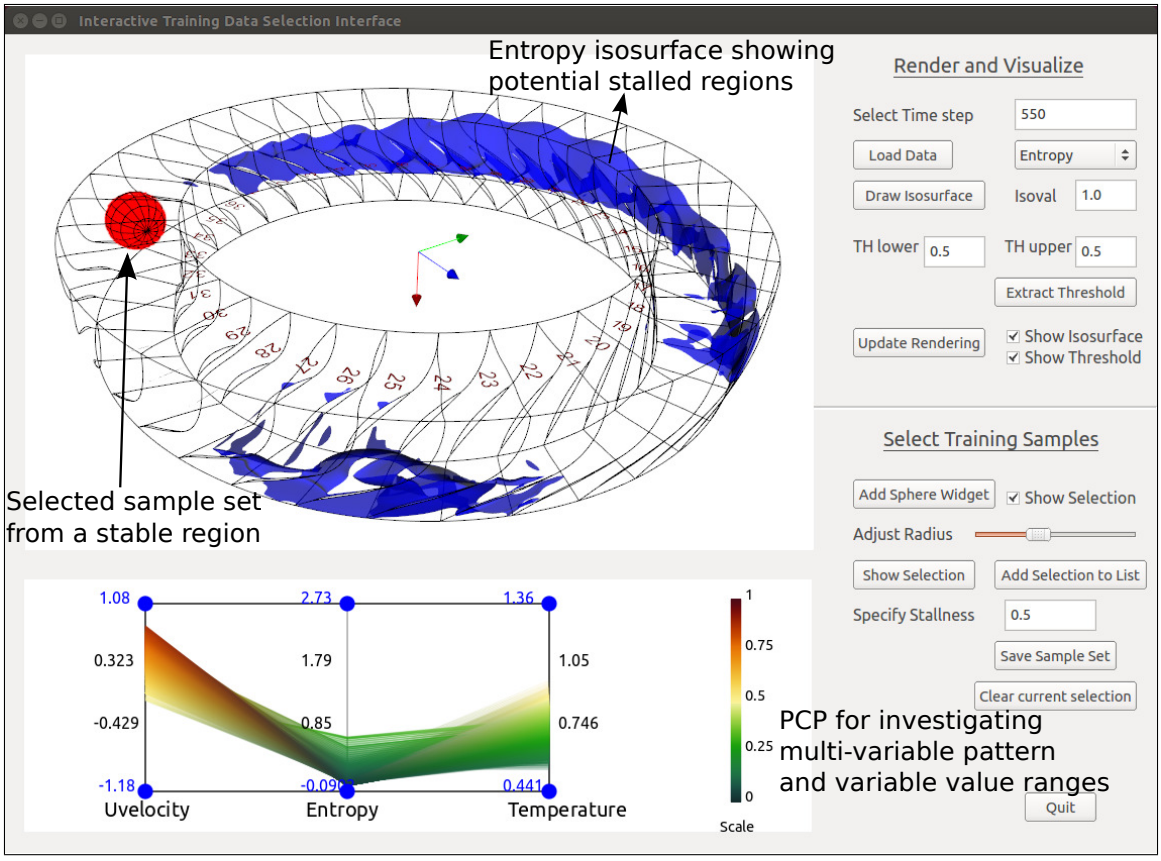


Figure 3.3: Interactive interface for selecting the region of interest from data.

3.2 Interactive Training Data Generation

In the absence of precise/hard feature descriptors the usefulness of the labeled samples, collected directly from the data by the expert, for efficient classification of vortices was demonstrated in previous studies [14,166]. Recently, it was also shown that visual interactive labeling of multiple data points can be used to generate training data for learning algorithms [11]. We follow the similar strategy and allow the expert to locate their features directly in the data. As hard classification of features is not possible, the expert labels a confidence value reflecting the degree of stallness for the selected points by specifying a value between

0 and 1, where 1 means definitely stall, and 0 indicates the stable condition. This strategy ensures a tight coupling of the expert-knowledge into our system [166].

We provide a visual-analytics system which the expert uses day-to-day for exploring the regions of interest. Since stall cells are associated with flow separation characterized by reverse axial flow, the expert is interested in locations where Uvelocity (Uvel) is negative. However, near the blade tip, a small amount of reverse flow always exists which does not indicate stall. Only when the amount of reverse flow grows significantly and blocks normal airflow, it can lead to a stall. Hence, instead of relying on a single variable, the expert also wants to explore entropy and temperature variables for locating stalled regions. This is because, based on the experts' domain knowledge, the stalled regions generally show high entropy and temperature values, i.e., they have a direct correlation, and Uvel values appear to be negative and low. Hence, Uvel is expected to have a negative correlation with the other two variables. The goal is to identify a sample set from which the above multivariate relationship among the three variables can be estimated using a set of fuzzy rules, such that the rules can be applied to infer the degree of stallness for unknown data set. Using our interface, the expert analyzes potential regions in the data using isosurfaces and thresholding techniques on multiple important variables and uses domain knowledge to iteratively refine the regions which best represents the stall cells. Our tool provides an adjustable 3D sphere widget for highlighting such regions. In Figure 3.3, we depict the interface where the red points show a selection using the sphere widget. The interactive Parallel Coordinates Plot (PCP) allows the expert to study the multivariate pattern among the selected variable values where the line colors reflect their scalar values.

A detailed investigation of several time steps from a known data set reveals that the stalled regions as mentioned above indeed demonstrate high entropy and temperature values,

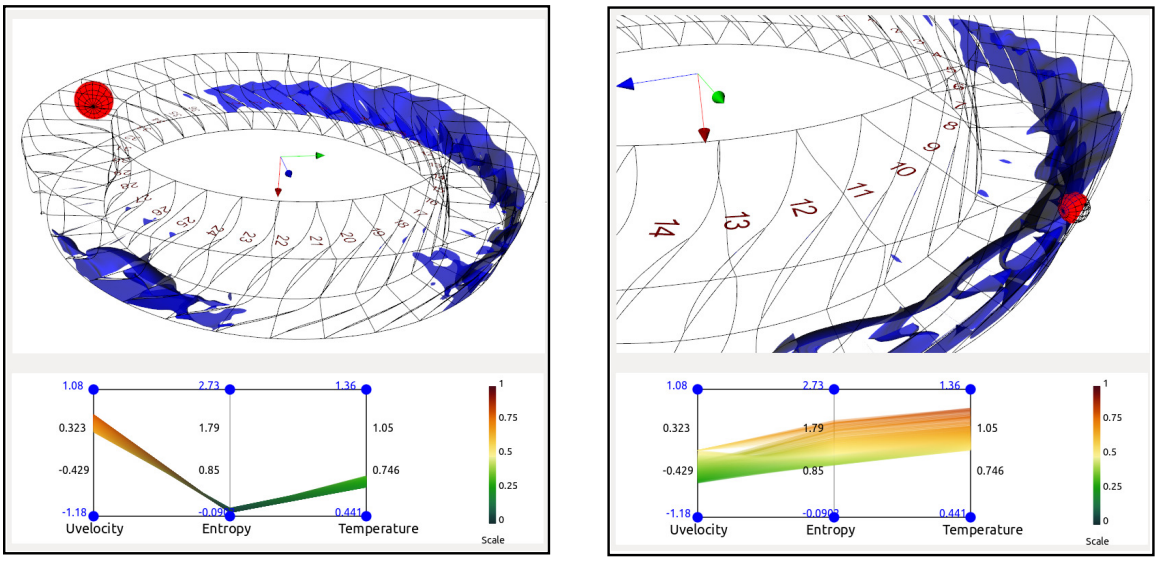


Figure 3.4: Interactive selection of samples for training where the stalled regions are roughly shown using a high entropy value isosurface. In the left image, the sample points highlighted (within the sphere in red) are selected from a stable region, whereas, in the right image the samples are picked from a stalled region. The differences in patterns among the three selected variable values shown in the PCP are notable.

whereas, an almost opposite relationship is found in the stable regions where the Uvel is positive and high, and entropy and temperature values are low. The left image in Figure 3.4 shows the case where the approximate stalled region is highlighted by an isosurface of a high entropy value of 0.9 and the selected sample points are marked in red coming from a stable region. The multivariate relationship of these points is revealed in the PCP below showing the correlation among the three variables. Observe that, the Uvel is high and positive and both entropy and temperature demonstrate low values. Hence, the stallness value of these sets of points is set to 0.1 by the expert indicating a low chance of a stall. In contrast, the right image in Figure 3.4, we see sets of points selected from a stalled region that is enclosed by the entropy isosurface. Here, the pattern is quite different as seen in the PCP below. The Uvel is negative indicating reverse flow, and entropy and temperature values are high. So,

the degree of stallness of these sets of points is set to 0.9 reflecting high confidence that the selected region is stalled. Following this approach, the expert can investigate several time steps and mark regions that are either stalled or stable and also specify the degree of stallness based on their domain knowledge. All these points are collected for constructing the fuzzy rule base.

3.3 Off-line Learning For Fuzzy Rule Generation

In this work, we deal with features that cannot be defined precisely. As a result, the expert usually employs visual analytics methods and finally roughly locates the features with a certain degree of confidence. However, manual identification of such features from thousands of time steps is undesired, especially when the data size is very large. So, robust and intelligent feature detection algorithms are essential for automatic classification. Also, since our target environment for feature detection is *in situ*, we require algorithms which offer fast classification capability with a low memory footprint. A fuzzy rule-based system (FRBS) in this case presents a suitable choice. For robust detection of imprecisely defined stall cells, we use an FRBS consisting of a knowledge-base and an inference engine. Use of an *in situ* friendly FRBS allows us to compactly transfer the domain knowledge of the expert into an intelligent system and automate feature classification.

3.3.1 Fuzzy Clustering Guided Rule Generation

In order to extract the relationship between the input values and their corresponding output values, the expert specified sample points are first clustered using a fuzzy-c-means (FCM) [12] clustering algorithm. Note that, each point in our training set is a 4-tuple: {Uvel, entropy, temperature, stallness}, where the first three values are observations of variable values and are treated as input to the fuzzy system, and the fourth component is the output

fuzzy classification score. By clustering the input and output values together, the generated clusters group input points which also have similar output. Efficacy of the FCM algorithm in extracting cluster structures from high dimensional data has been demonstrated previously in [106] and has been adopted in many cluster based rule generation techniques. The algorithm generates a membership matrix along with the cluster centers. The membership matrix contains the membership of each point to all the clusters. The sum of membership values across all the clusters for each point is 1. So, given $X = \{x_1, x_2, .., x_n\} \in \mathbb{R}^p$ as the input to the FCM, the algorithm produces a set of centroids $V = \{v_1, v_2, .., v_c\}$ and a membership matrix M by minimizing the objective function:

$$\psi_r(M,V) = \sum_{k=1}^{n} \sum_{i=1}^{c} m_{ik}^r \|x_k - v_i\|^2 \tag{3.1}$$

where r is a weighting exponent (typically = 2), n is the number of points in the training set, and c is the number of clusters. The dimension of the output membership matrix M is $c \times n$ and the element m_{ik} represents the membership of k^{th} point in i^{th} cluster.

As each cluster encapsulates a specific relationship among the grouped input-output sample points, each such cluster is formalized into a *IF-THEN* predicate-based fuzzy rule with a standard form: *IF (antecedent) THEN (consequent)*. The antecedent clause of each rule consists of several atomic sub-clauses and are connected using fuzzy T-norm conjunction operators that aim to estimate the degree of "closeness" of the given input component to the corresponding sub-clauses of the rule. We use the fuzzy operator "AND" in this work as the fuzzy conjunction operator. To quantify this degree of "closeness", each sub-clause in the rule is modeled by a fuzzy membership function. Note that, different types of membership functions can be used here such as triangular, trapezoidal, Gaussian etc. In this work, we have used Gaussian membership function (GMF) because it is differentiable everywhere and has only two parameters which can be tuned efficiently [106]. Therefore,

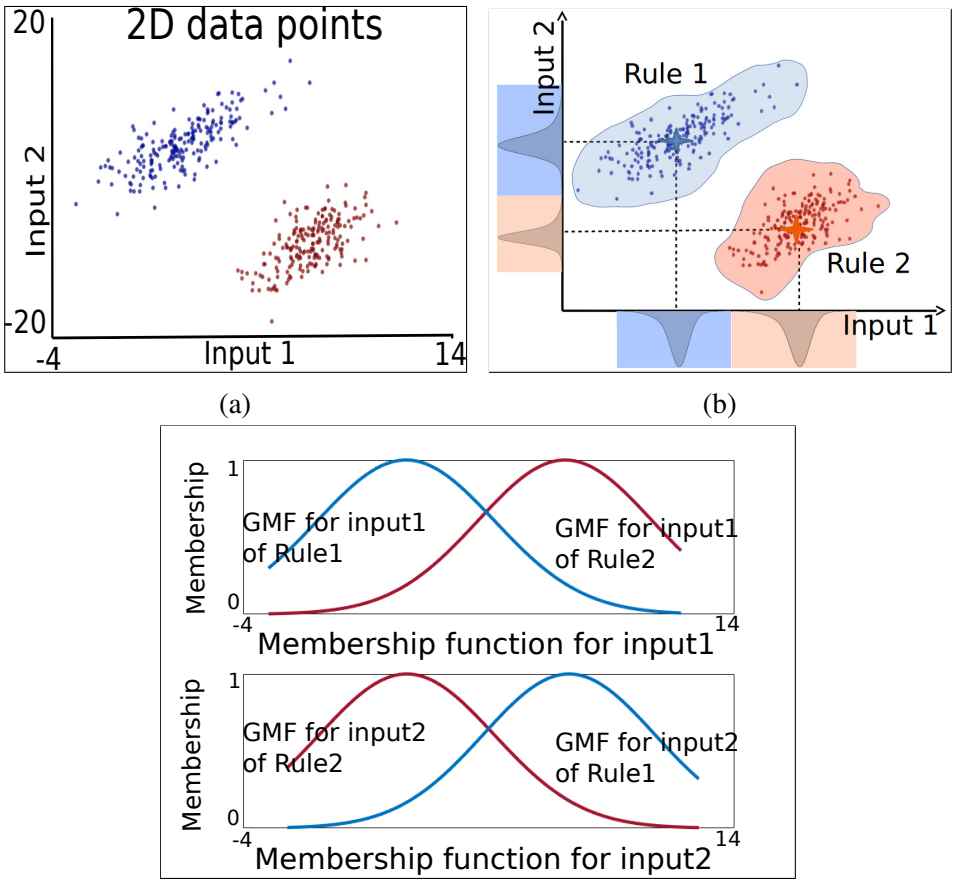


Figure 3.5: Figure 3.5a: A synthetic bivariate training data generated from two 2D multivariate Gaussian distributions centered at (2,8) and (8,2) respectively; Figure 3.5c: Trained Gaussian membership functions (GMF) for the sample bivariate data; Figure 3.5b: Conceptual scheme of fuzzy clustering based rule identification.

the degree of closeness in each sub-clause is estimated by evaluating the GMF associated with it. This process is also known as the "fuzzification", which changes a real scalar value into a fuzzy value. Formally a GMF is defined as:

$$gmf(x) = exp^{-\frac{(x-\bar{x})^2}{2\sigma^2}} \tag{3.2}$$

where $\bar{x}$ is mean and σ is the standard deviation. The estimated centroid of each cluster then becomes the suitable choice for mean values of the corresponding GMFs and the standard deviation of each GMF is computed as:

$$\sigma_i^j = \sum_{u=1}^{n} \frac{-(x_i^u - v_i^j)^2}{2 \cdot log(m_i^u)} \tag{3.3}$$

where σ_i^j is the standard deviation of the i^{th} clause of j^{th} rule, m_i^u is membership of u^{th} sample point obtained from the membership matrix M, and n is the number of points in the training set.

To demonstrate this concept of rule generation, we created a synthetic 2D data set shown in Figure 3.5a which was obtained by randomly sampling points from two 2D multivariate Gaussian distributions centered at $(2,8)$ and $(8,2)$ respectively. It can be observed that the 2D point set has two clusters. The output value for points coming from the first Gaussian was set to 0 and for the points sampled from second Gaussian was set to 1.0 for experimentation. Hence, the training set, in this case, is a set of 3-tuples. Since each cluster is translated to a fuzzy rule, the rule for the first cluster can be written in the form: "IF (($input_1$ is close to $input_1$ of cluster $center_1$) AND ($input_2$ is close to $input_2$ of cluster $center_1$)) THEN the point has output value close to 0". Similarly, the second rule for the other cluster will be: "IF (($input_1$ is close to $input_1$ of cluster $center_2$) AND ($input_2$ is close to $input_2$ of cluster $center_2$)) THEN the point has output value close to 1". In Figure 3.5b, we show the schematic diagram using the 2D synthetic data where the mean values of the membership functions are estimated by the cluster centroids and each cluster is interpreted as a fuzzy rule. The estimated GMFs are depicted in Figure 3.5c. It can be observed that each rule with two sub-clauses has two membership functions (denoted by blue for rule 1 and orange for rule 2). With the estimated GMFs, we can construct the antecedent part of the rules. Next,

we demonstrate how the estimation of the parameters for the output prediction function is achieved which is the consequent part of the rule.

3.3.2 Estimation of Parameters for the Output Function

Since the inference algorithm of the fuzzy system will be run *in situ*, we model the output variable using a linear function allowing computationally efficient inferencing. We adopt the well-known *Takagi-Sugeno fuzzy rule-based system* (TS-FRBS) which has been shown effective in modeling various dynamic systems in the past [115, 119]. The output in a TS-FRBS is represented as a linear function of input variables. The value of this output indicates the confidence of the system for the input tested. Now, assuming the output variable y^j is a linear function of the input variables for the j^{th} rule, the output function $g(\cdot)$ can be represented as:

$$y^j = g(x_1^j, ..., x_q^j) = p_0^j + p_1^j \cdot x_1^j + ... + p_q^j \cdot x_q^j \tag{3.4}$$

where $p_0^j, p_1^j ... p_q^j$ are the coefficients of the function $g(\cdot)$, and q is the number of input dimensions. Given the estimated GMFs and the sample training data, the output parameters $p_0^j, p_1^j ..., p_q^j$ are computed by optimization with respect to the training data and this optimization reduces to a linear least square estimation problem as described in [106]. By solving this least square problem we obtain the parameters for output function, i.e., the consequent part of the rule. Therefore, with the GMFs, the fuzzy rules, and the model for the output variable learned from the training data, characterization of the fuzzy system is complete. Next, we describe how the fuzzy rules are combined to infer output values for unknown input data.

3.3.3 Fuzzy Rule-Based Feature Classification

The output response of a TS-FRBS is represented as a linear function of input variables. Formally, given a specific input $(x_1,...,x_q)$, and a set of fuzzy rules $R^j,\ \ (j = 1,2,..,c)$, the value of output y^j is inferred as follows. First the input is evaluated through each of the rules and a degree of match is computed which is called the *firing strength* of that rule for the input. The firing strength α^j of j^{th} rule is computed as:

$$\alpha^j = (gmf_1^j(x_1^j)\ \ \wedge\ \ gmf_2^j(x_2^j)\ \ ...\ \ \wedge\ \ gmf_q^j(x_q^j)) \tag{3.5}$$

where gmf_1^j, gmf_2^j ... gmf_q^j are the membership functions of the form described in Equation 3.2, $\wedge$ is a fuzzy conjunction operator. We have used the "AND" as the fuzzy conjunction operator since all the sub-clauses in the rule need to satisfy simultaneously. This fuzzy "AND" operator is also known as the "Product t-norm" and is typically realized by algebraic product of the membership values. Intuitively, the firing strength estimates the degree of match of rule R^j for the given input by conjunction of the membership contributions coming from the evaluation of the sub-clauses using the Product t-norm. Hence, if most of the sub-clauses of a rule have satisfied strongly for a given input, then the firing strength of that rule for the input will be high. Note that, for any given input, the fuzzy rule base produces j output values and the final output response y for that input is computed as the average of all y^j values weighted by their firing strengths as:

$$y = \frac{\sum_{j=1}^{c} \alpha^j \cdot y^j}{\sum_{j=1}^{c} \alpha^j} \tag{3.6}$$

To explain the above inference algorithm, the example 2D synthetic data are shown in Figure 3.5a is used. The estimated GMFs presented in Figure 3.5c. The blue GMFs represent rule 1 and the orange GMFs represent rule 2. Given an unknown input point $(1.7, 7.8)$, as it is closer to blue cluster, it will generate high firing strength for rule 1 and low

firing strength for rule 2. Furthermore, we also expect the final output value, i.e., the fuzzy classification score to be very close to zero since the points centered at $(2, 8)$ in the training set was assigned output value zero. By applying the inference algorithm presented above, we obtain the output value as 0.0043, which is very close to zero as expected. Similarly, when another input point $(7.1, 2.6)$ is tested, the output is 0.8540 which is close to 1 and can be considered as part of a cluster 2. Finally, when testing an input point $(5, 5)$ which cannot be classified strongly to any of the clusters, the fuzzy system produces an output of 0.4921 indicating its fuzzy classification. For conditions like these, when a hard classification is not possible, use of a fuzzy system helps us to perform a confidence driven classification considering the uncertainty.

3.4 In Situ Feature Detection for Stall Analysis

We employ the aforementioned fuzzy rule-based system for the classification of stall impacted regions. We construct the rule base using an expert selected training set containing observations from both stalled and stable regions as discussed in Section 3.2 and 3.3. The training is first done off-line using a pre-generated simulation data set which contains the stalled condition. After all the parameters of the fuzzy system and the rules are estimated, we employ the inference algorithm in the *in situ* environment for other unknown simulation data sets. Note that, we directly apply the learned system from one pre-generated simulation data to other cases without repeating the learning again. We also estimate the local mass flow rate for each passage. It is hypothesized that, as the stall cells block the normal airflow through passages, the mass flow rate of the stalled passages will be lower compared to the stable ones. So, the analysis of the local mass flow rate over time can be used as a complementary source of information to that of the fuzzy system based stall detection for enhancing the

overall effectiveness and robustness of our stall analysis. Below we discuss how we estimate these two measures during the *in situ* run and enable flexible stall exploration in detail.

3.4.1 Fuzzy Rule-Based *In Situ* Stall Prediction

The fuzzy inference system works on each data point by assigning a value which reflects its "stallness". The input to the fuzzy inference system for each point is a 3-tuple: {Uvel, entropy, temperature}. After the application of the inference algorithm on all the points, we can construct a new classification field where the scalar value at each point will reflect the stallness of the point. Higher values of stallness indicate higher chances of being part of a stall cell. As stated in Section 3.1.2, one of the primary goals of the expert is to study the evolution of stall impacted regions over time, and furthermore, the expert wants to know whether these stalled regions are detected near the blade tip or close to the hub. To answer these questions, we summarize the classified stallness field in two specific ways. First, we aggregate the points that have stallness greater than an expert specified stallness threshold for a passage which reflects the stallness of the whole passage. Secondly, to estimate their relative spatial extent inside each passage, we aggregate all the points along the axis spanning from hub to tip by counting the number of points for each segment along this direction using the same stallness threshold used above. These two types of aggregated information for each passage are stored in the disk and visualized during post-hoc analysis for studying the evolution of stall phenomenon.

3.4.2 Local Mass Flow Rate Estimation

Besides classification of data using their predicted stallness, we further estimate the local mass flow rate for each passage during the simulation run. **Mass flow rate** is a measure of the air mass passing through the compressor stage per unit time [60] and can be computed

as $\dot{m} = \rho\vec{v} \cdot \vec{A}$, where ρ is the density, $\vec{A}$ is the area vector of a flow path cross-section of the inlet or exit of the compressor, and $\vec{v}$ is the flow velocity. The global value of mass flow rate in stable condition remains consistent over time, however, when the simulation generates stall, it drops rapidly. While the indication of stall measured by global mass flow rate is discernible, it appears too late to apply any stall preventive measures. Hence, we estimate per passage local mass flow rate for each time step. The scientist hypothesizes that by analyzing the local mass flow rate evolution among blade passages, the indications, and development of stall will be more effective compared to the global mass flow.

3.5 Visual Interface for Studying Stall Evolution

For post-hoc visual analysis of stall, we have developed an interface shown in Figure 3.6. The goal is to provide insights about the evolution of the simulation by presenting feature specific information derived *in situ* using fuzzy rule system and local mass flow rate. Our interface depicts information through three different views:

Evolution of temporal stallness. The temporal overview shows the time-varying evolution of stall impacted regions detected by the fuzzy system prediction (top chart in Figure 3.6). The x-axis represents time and the y-axis is mapped to the blade passages. Each cell in this plot reflects the degree of stallness of the passage at a time step. The primary purpose of this plot is to present the overall temporal trend of the detected stalled regions. This helps the expert to (a) Locate the time step ranges when signs of stall start appearing; (b) The blade passages that get affected by the stall; (c) The overall temporal pattern of the propagation of the detected stalled regions from one passage to another. However, note that this plot does not reveal the location of the detected regions in the spatial domain.

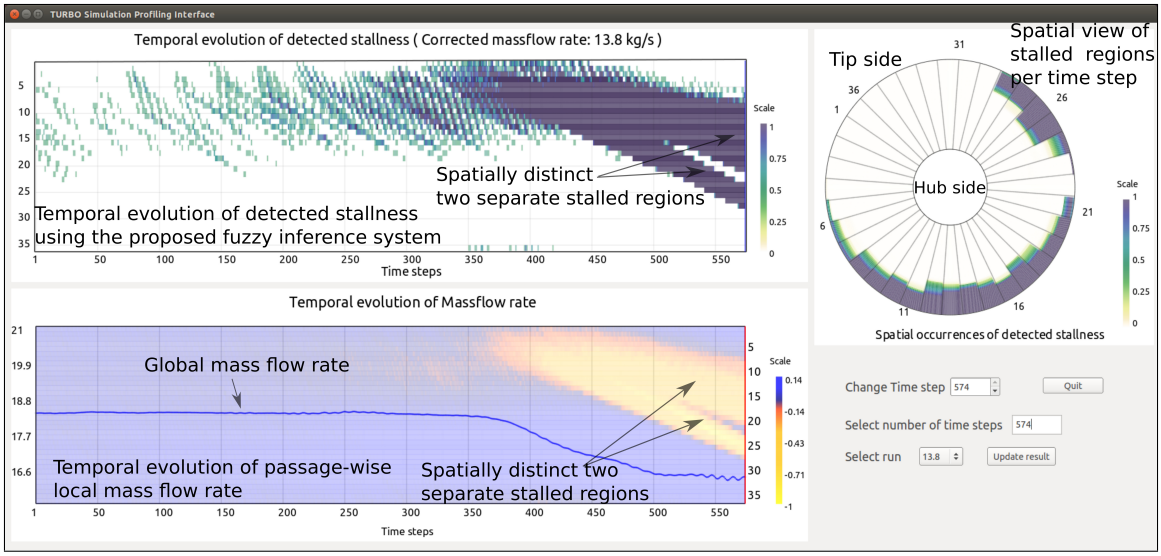


Figure 3.6: Visual analytics interface to study the stall features (CMF = 13.8 kg/s) estimated *in situ* through fuzzy rule-based system and local mass flow rate computation.

Spatial view of stallness. To show the spatial locations of the detected stalled regions for each time step, a 2D circular plot is presented in the top right panel. Here, the detected regions in each passage are laid out radially center-to-outward, i.e., from hub to the tip of the compressor stage. Presenting information in this intuitive format which has a direct correspondence to the actual physical rotor helps the expert to easily comprehend the spatial organization of the stall cells at each time step. This is why a radial layout was preferred in our work compared to a linear layout for showing this information. Furthermore, a radial layout also offers a compact use of the screen space for creating visualizations. From Figure 3.6, we observe that a majority of the stalled regions are detected near the blade tip. Note that, this spatial aggregated 2D view shows only the relative spatial extent of the stalled regions from the hub to the tip. In this view, at each level of aggregation from the hub to the tip, the color of each semi-circular segment reflects the number of stalled points. However,

it does not distinguish regions within each semi-circular band and uses a single aggregated value to represent the stallness. To show more detailed locations in this spatial view, we can segment each semi-circle into further smaller segments and keep the number of stalled points from each such small segment. This will increase the overall storage of our *in situ* summarized information while making the visualization more precise.

Evolution of local mass flow rate. The information derived from mass flow rate is shown on the bottom left panel where the x-axis is time steps. The solid blue line indicates the global mass flow rate and the y-axis on the left show its value range. Here, we compute the relative deviation of local mass flow rate with respect to the mean local mass flow rate of the rotor. According to the expert, in a stable condition, the local mass flow for each passage remains identical due to the axisymmetry property of the rotor, so any passage that deviates significantly from the expected (mean) mass flow is abnormal. For each time step, we estimate the relative deviation of local mass flow rate $\dot{m}dev_i$ for the i^{th} passage as:

$$\dot{m}dev_i = (\dot{m}_i - \overline{\dot{m}})/\dot{m}_i \tag{3.7}$$

where $\overline{\dot{m}}$ is the mean local mass flow at a given time step, and $\dot{m}_i$ is the mass flow of the i^{th} passage. Note that, the mass flow deviations which are smaller than the mean, i.e., the negative mass flow deviation values are of our interest, since mass flow of the stalled passages drops during the stall. The relative mass flow deviation values computed per passage are shown using a heat map based visualization as seen in Figure 3.6. Each cell in this heat map reflects the relative mass flow deviation of a passage at a time step. The passage ids are depicted on the right side of the chart. Note that, each cell in this plot has a one-to-one correspondence with the above stallness chart. Laying out the mass flow information in this way was preferred by the expert since the expert could easily investigate

the mass flow deviation values of important stalled regions identified from the stallness chart in any time step. Observe that the relative mass flow deviation values convey more information of stall inception where the asymmetry in the mass flow among the passages is clearly seen from the color variation in the heat map.

3.6 Verification Through Domain Expert Evaluation

We verify the proposed method by applying it to a known data set with corrected mass flow rate (CMF) = 13.8 kg/s. The simulation was run for 4 revolutions and raw data for every 25^{th} time step was stored resulting in 576 time steps. The rotor in TURBO consists of 36 passages with a spatial resolution of $151 \times 71 \times 56$ for each passage. The Gaussian membership functions (GMFs) obtained after the training, are shown in Figure 3.7. From the GMFs, one can observe that data points coming from stable regions will find strong degree of match with rules 1 and 3 as they will have high firing strength from these rules, and similarly low degree of match with rule 2, resulting in a low stallness output from the fuzzy inference algorithm. This is because the stable regions demonstrate high and positive Uvel and low to moderate entropy and temperature values. So, in fuzzy logic, linguistic terms rule 1 can be articulated as IF Uvel is high AND entropy is low AND temperature is low THEN the predicted Stallness is low (close to 0.1 as marked by the expert). Similarly, rule 3 will be described as IF Uvel is moderate to high AND entropy is low AND temperature is moderate THEN the predicted Stallness is low. In contrast, the rule 2 can be interpreted as IF Uvel is negative and low AND entropy is high AND temperature is high THEN the predicted Stallness is high (close to 0.9 as marked by the expert). Note that, in this case, the stalled points will have high firing strength, i.e., a close degree of match with rule 2, whereas a low degree of match with rules 1 and 3 which will produce high stallness output.

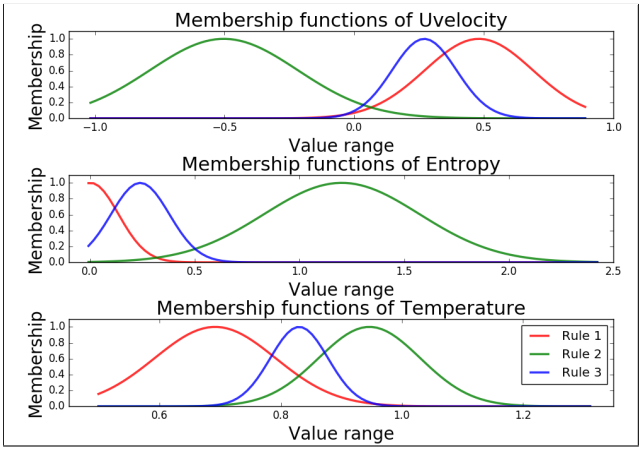
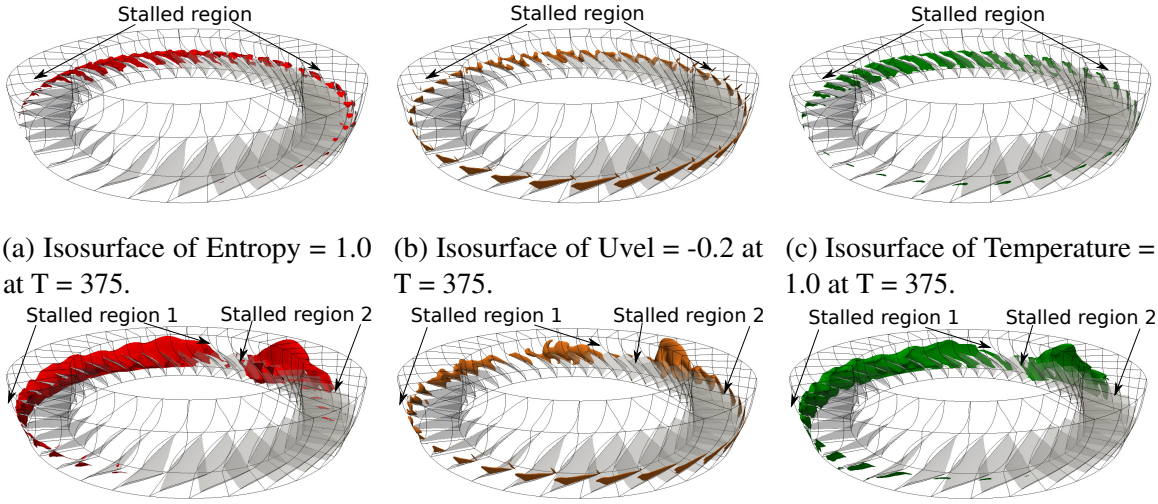


Figure 3.7: The Gaussian membership functions (GMF) generated using sample data collected from a simulation run with CMF=13.8 kg/s. Each color indicates membership functions of a rule in the image.

Accuracy study of the fuzzy system. We conducted an accuracy analysis using a cross-validation technique called the repeated random sub-sampling validation [44]. The sample set consisted of 573460 points. The points were collected from both the stalled (stallness of 0.9) and stable regions (stallness of 0.1) from several time steps of a known data set with CMF = 13.8 kg/s. In this cross-validation, we randomly divided the data into two equal groups, and one was used for training and the other for validation. We repeated this process 1000 times to obtain robust accuracy results by taking the average (expected) outcome of all the trials. We found that the stallness of 92.67% points were predicted within a small deviation of 0.1, and 97.47% points were within the deviation of 0.15 of the user marked stallness values. Besides this, we also measured the percentage of false positive points, i.e., the points predicted as stalled while considering a deviation of 0.1, but the points should not be considered as stalled from the labels in the test data. We found that the average percentage



(a) Isosurface of Entropy = 1.0 at T = 375.

(b) Isosurface of Uvel = -0.2 at T = 375.

(c) Isosurface of Temperature = 1.0 at T = 375.

(d) Isosurface of Entropy = 1.0 at T = 560.

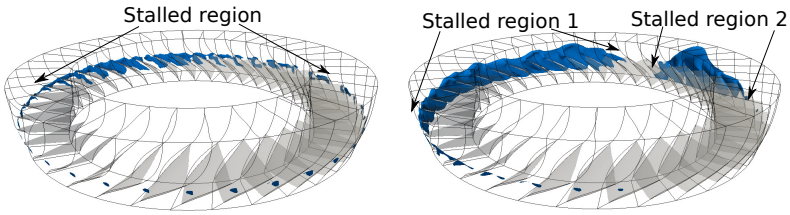
(e) Isosurface of Uvel = -0.2 at T = 560.

(f) Isosurface of Temperature = 1.0 at T = 560.

Figure 3.8: Visualization of entropy, Uvel, and temperature isosurfaces for locating the stalled regions in the CMF=13.8 kg/s data set. Figure 3.8a, 3.8b, and 3.8c show the isosurfaces at T = 375 when the global mass flow rate drops as shown in the bottom left panel of Figure 3.6 indicating stall inception. Figure 3.8d, 3.8e, and 3.8f depict the isosurfaces of the same variables at later time, T = 560 when the stall is well developed. It can be seen that two separate stalled regions are formed as marked in the images.

of false positive points is 3.95%. Furthermore, if we increase the stallness deviation to 0.15, the percentage of false positive points reduces to 1.043%.

Verification using the visual-analytics system. Figure 3.6 depicts the result of our method when applied to all the time steps of the run with CMF=13.8 kg/s. The temporal pattern of the predicted stallness, shown in the top left panel, demonstrates that our method is able to capture the overall evolution of stall. This plot also shows that passages ranging from 5-15 start showing signs of stall around time step 200 where the dark blue cells reflect passages with larger stalled regions compared to others. In the bottom left panel, the global mass flow, indicated by the solid blue line, confirms the occurrence of the stall when it starts



(a) Detected Stallness at T = 375. (b) Detected Stallness at T = 560.

Figure 3.9: Visualization of isosurfaces of the fuzzy system predicted stallness field. Figure 3.9a shows isosurface of 0.8 at T = 375, and Figure 3.9b shows isosurface of 0.8 at T = 560. The detected regions at both of these time steps correspond well with the regions identified in Figure 3.8 validating the correctness of our method.

to drop around time step 375. The heat map colored by the relative deviation of local mass flow rate in the same panel shows that several passages deviate from the expected mass flow breaking the uniform flow through all the passages. This asymmetry in mass flow, caused by lower local mass flow rates, is detected around time step 325 (the scattered reddish cells around passages 10-15) indicating a potential stall inception. The relative mass flow finally drops significantly for those passages (the yellow regions) for later time steps as the stall grows.

Next, we provide a 2D radial plot based visualization of the rotor on the right in Figure 3.6 where the spatial locations of the stalled regions can be studied. Investigation of stalled regions in this way for each time step allows the expert to verify that the stall cells are formed close to the blade tip. To verify this, we show isosurfaces of a high entropy value of 1.0, a negative Uvel value of -0.2, and a high temperature value of 1.0 in Figures 3.8a, 3.8b, and 3.8c respectively. All these surfaces indicate a common region (as marked in the figures) where the stall is happening. Figure 3.9a shows the isosurface extracted from the predicted stallness field using an expert specified high stallness value of 0.8 at T = 375. It is

Table 3.1: Different CMF values used for experimentation and their outcomes.

Simulation number	CMF (Kg/s)	#Revs. simulated	Outcome of the run
1	13.8	4	Stalled (Previously known)
2	14.0	4	Stalled (Expected to stall)
3	14.2	8	Stalled (Previously known)
4	14.5	8	Stable (Previously unknown)
5	14.8	4	Stable (Previously unknown)
6	16.0	4	Stable (Previously known)

observed that the predicted region matches with the regions obtained from the data quite well in Figures 3.8a - 3.8c. Next, Figures 3.8d, 3.8e, and 3.8f show the isosurfaces of a later time step when the simulation has transitioned into a deep stall. We observe that there are two separate stall cells, and our method in Figure 3.9b is able to capture it. These two stalled regions are also marked in the stallness and the mass flowchart in Figure 3.6. A similar observation is seen in the spatial view for a specific time step. So, by studying the local mass flowchart and the fuzzy system predicted stallness map, the expert concluded that our method provides a comprehensive view of the evolution of stall.

3.7 In Situ Stall Analysis With Various Parameter Configurations

We tested our method using 6 different runs with different CMF values as reported in Table 3.1 with their outcomes. We worked closely with an expert having more than 20 years of experience in flow simulations and is one of the developers of TURBO code. Feedback was collected through regular bi-weekly meetings with the expert. The learning algorithm took 1.027 seconds to generate the fuzzy system with 3 rules using 573460 training samples. Note that, the appropriate number of rules are dependent on the target applications [106], and since we want to distinguish points which are either stalled or not, the minimum number

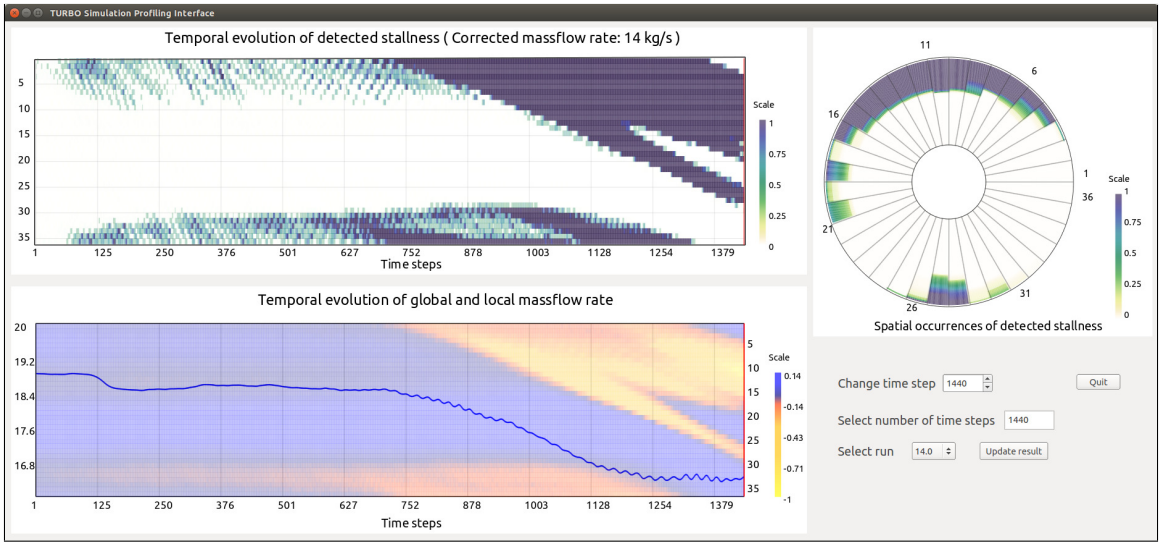


Figure 3.10: Result of simulation run with CMF=14.0 kg/s. This resulted in a stalled condition which is visible from the stallness and mass flow deviation plot.

of rules can be 2. However, that may result in high variability in each cluster. So, we tested with $3 \sim 5$ rules and found that the results are similar with minor variations without changing the overall outcome. Also, a higher number of rules increases computational cost of the algorithm. So 3 rules for all the experiments were used producing satisfactory results. The *in situ* code, developed in C++, is linked to TURBO as a new module. The *in situ* routines directly access the simulation memory minimizing any additional data copy.

3.7.1 Simulation with Stall (CMF = 14.0 kg/s)

We first tested our system using a new run with CMF = 14.0 kg/s. It was expected that this run would stall since, in a previous study [30], a run with a higher CMF of 14.2 kg/s produced stall. We simulated 4 revolutions resulting in a fully developed stall. As each revolution of TURBO runs 3600 iterations, for 4 revolutions, a total of 14400 iterations were run. *In situ* call was made at every 10^{th} iteration resulting in 1440 time steps. Here, the

time step numbers used are in the units of tenths of simulation iterations due to the sampling rate of *in situ* call. At every 10^{th} time step, we applied our fuzzy rule-based algorithm and estimated the local mass flow rates. The stallness values estimated by the fuzzy algorithm were then aggregated *in situ* as discussed earlier in Section 3.4, and finally, the task-specific aggregated information was stored.

As can be seen from the stallness and local mass flow deviation charts in Figure 3.10, the simulation produces a stalled condition. Indications of a potential stall start appearing around time step 70 in the stallness overview chart. Finally, around time step 750 the stall happens when dark blue regions become persistent. The global mass flow rate also starts dropping around the same time step, and the local mass flow becomes significantly asymmetric, reflecting the occurrence of the stall. Also, similar to the stallness pattern of CMF = 13.8 kg/s run, this run too produces two separate stalled regions within passages 1-12 and 15-20 starting around time step 1240. From these observations, the expert is able to verify the hypothesis that the CMF value of 14.0 kg/s indeed produces a stall.

3.7.2 Simulation with Stall (CMF = 14.2 kg/s)

Next, we tested our method using a run with CMF = 14.2 kg/s which is a known stalled condition. The simulation was run for 8 revolutions requiring us to process 2880 time steps. The results are presented in Figure 3.11 where the development of the stall is visible from both the stallness and mass flow deviation plots. The early indication of the stall is seen around time step 350 in the stallness plot. This grows consistently over time, and around time step 2508 the major flow breakdown happens reflected by the persistent and blue regions covering passage range 20-30. The consistent drop in global mass flow around time step 2508 validates this event. Note that, the local mass deviation chart, in this case, starts

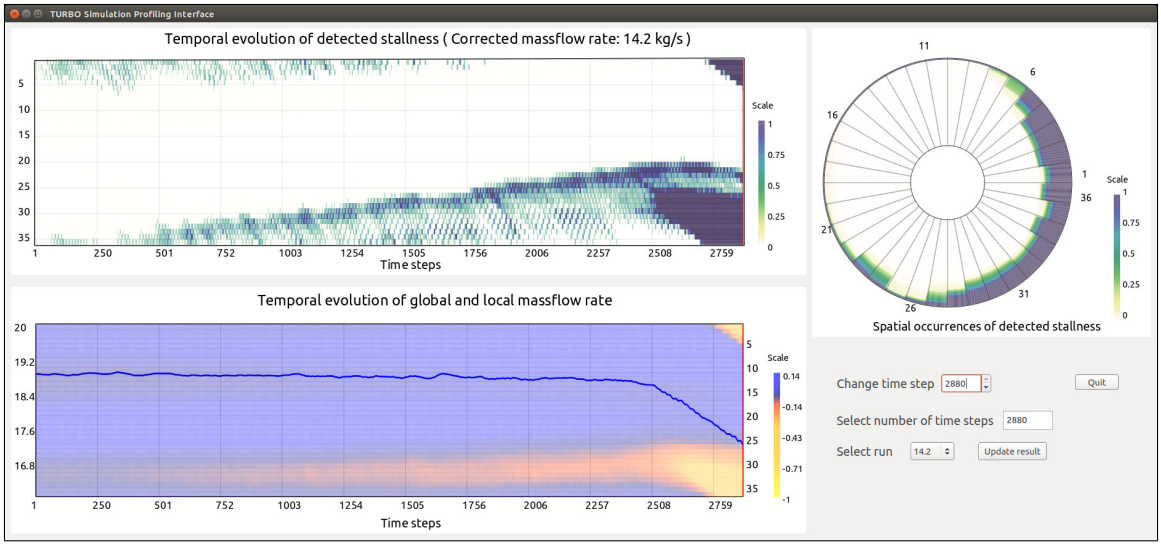


Figure 3.11: Result of simulation run with CMF=14.2 kg/s. Provided CMF value drives the simulation into a stalled state which our proposed method is able to detect correctly.

showing indications of a future stall starting around time step 500 onwards within passages 30 - 33. The deviation of mass flow gradually becomes large and persistent reflecting the evolution of flow instability. The expert explained that this deviation is caused by the drop in mass flow rate in the stalled regions where stall cells act as blockages. The spatial plot shows the stalled regions in a radial map for a specific time step confirming that the stalled regions are concentrated on the blade tips. Finally, we observe that compared to the previous run with CMF = 13.8 kg/s and CMF = 14.2 kg/s, this case produces stall in a different set of passages.

3.7.3 Simulation with Unknown Parameter Configuration (CMF = 14.8 kg/s)

Our next experiment was conducted with CMF = 14.8 kg/s, suggested by the expert whose outcome was unknown. The simulation was first run for 4 revolutions. As seen from

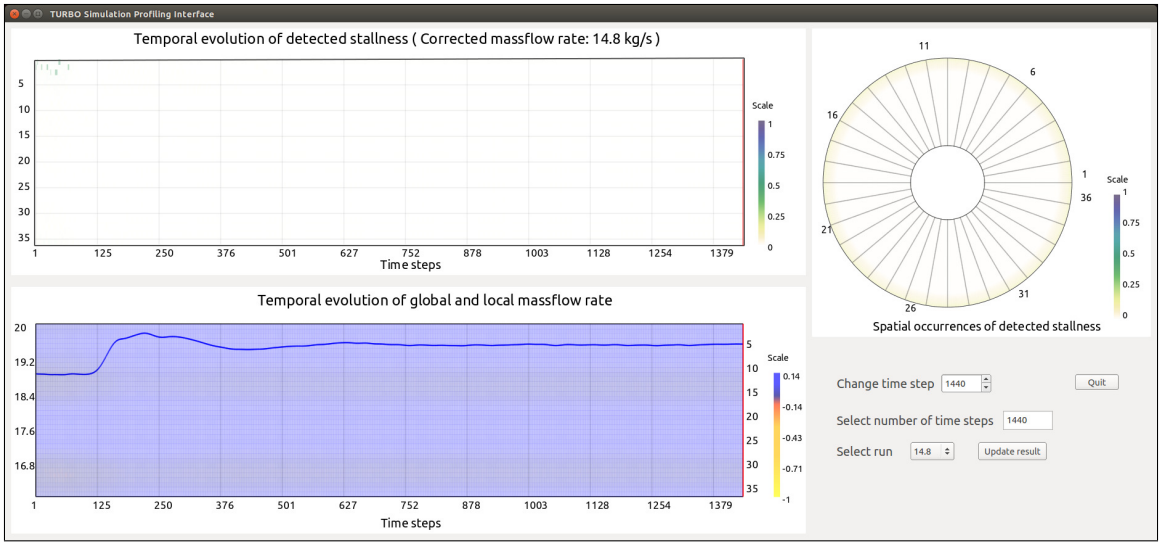


Figure 3.12: Result of simulation run with CMF=14.8 kg/s. The outcome was a stable run with uniform mass flow chart and clean stallness plot.

Figure 3.12, we find that both the overview and the spatial stallness chart is clean, and the local mass flow deviation chart is uniform as well. Since the restart files of the simulation at the beginning was obtained from a previous run with a lower CMF value, we observe an increase in the global mass flow rate initially, which saturates to a stable state. Judging by the clean stallness plot and uniform mass flow distribution, the expert concluded that this parameter setting produced a stable run. Since it was already known that CMF of 16.0 kg/s produces a stable run, this experiment further confirmed that CMF values higher than 14.8 kg/s will produce stable runs. However, as we were still unsure about the outcome of the runs between CMF 14.2 and 14.8, the expert suggested another experiment with CMF = 14.5 kg/s.

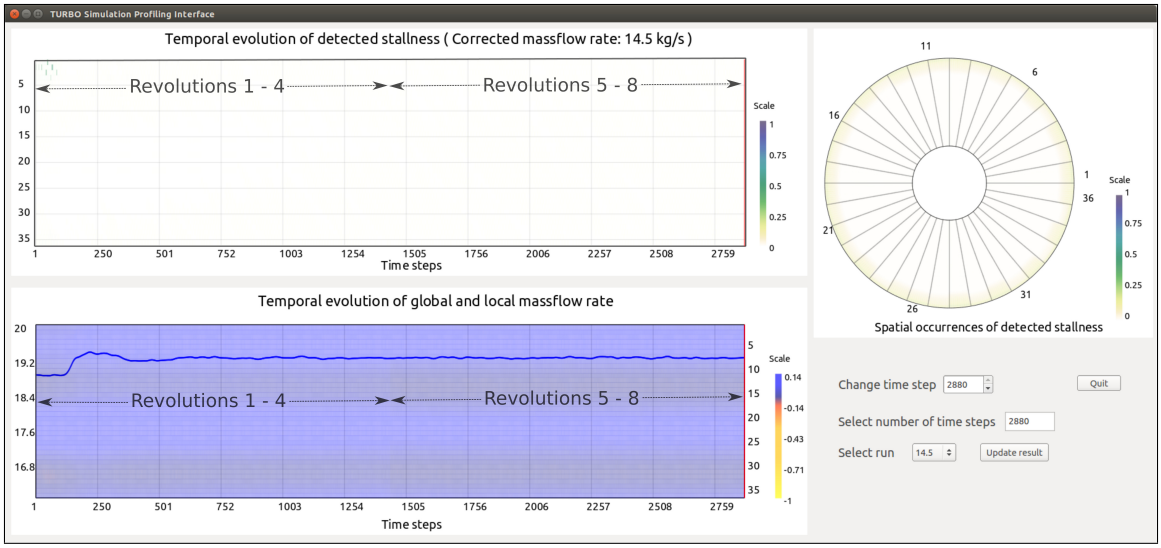


Figure 3.13: Result of simulation run with CMF=14.5 kg/s. The simulation resulted in a stable run which is observed from uniform mass flow chart and clean stallness plot.

3.7.4 Simulation with Unknown Parameter Configuration (CMF = 14.5 kg/s)

Here we present the result of the run with CMF = 14.5 kg/s where the outcome was again unknown. The simulation was initially run for 4 revolutions. By observing the result of first 4 revolutions, we found that the stallness plot was clean and the mass flow deviation plot was uniform. This can be seen in Figure 3.13 for time step ranges of first 4 revolutions as marked. To verify that this run was indeed a steady state case, the expert suggested continuing to run it for another 4 revolutions. As we can see from Figure 3.13, the revolutions 5-8 continued as a stable state.

Further verification by comparison with existing method. We computed spatial point-wise anomaly used by Chen et al. [30] for revolutions 5-8 of the 14.5 kg/s run. The point-wise anomaly shown in Figure 3.14 produced a clean plot with sporadic noises, which did not reflect stall. Hence, with this comparison and the results obtained by our method,

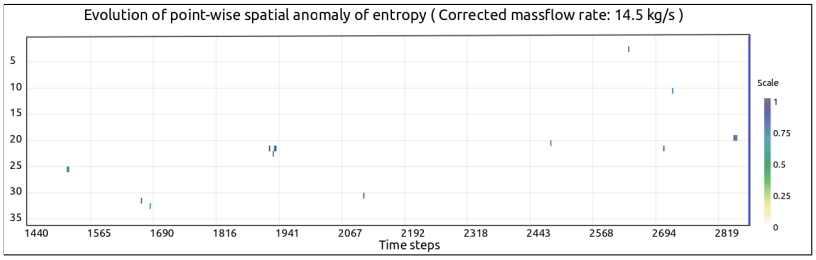


Figure 3.14: Spatial point-wise anomaly plot of entropy for the simulation run with CMF = 14.5 kg/s proposed in [30]. The anomaly computation was done for revolutions 5-8.

the expert concluded that 14.5 kg/s run was a stable run. However, this spatial anomaly method requires data from all the passages to estimate the value of anomaly. Therefore, this method is not inherently *in situ* friendly as it would require additional data communication. In contrast, the proposed method is parallel in nature which makes it a preferable choice for the expert when timely analyzable result extraction from an ensemble of simulation runs is essential.

3.7.5 Simulation without Stall (CMF = 16.0 kg/s)

Finally, for completeness, we present the results of the run with CMF = 16.0 kg/s. It was run for 4 revolutions to produce a stable data set. In Figure 3.15 we present the results from this run. As can be seen, the stallness plot is clean and the mass flow deviation plot is also uniform throughout the run with a stable global mass flow rate over all the time steps validating it as a stable run.

3.7.6 Summary of Results and Discussion

Summary of the stall analysis. The above analyses allow the expert to obtain a detailed understanding of the influence of the parameter CMF on the simulation outcome. With

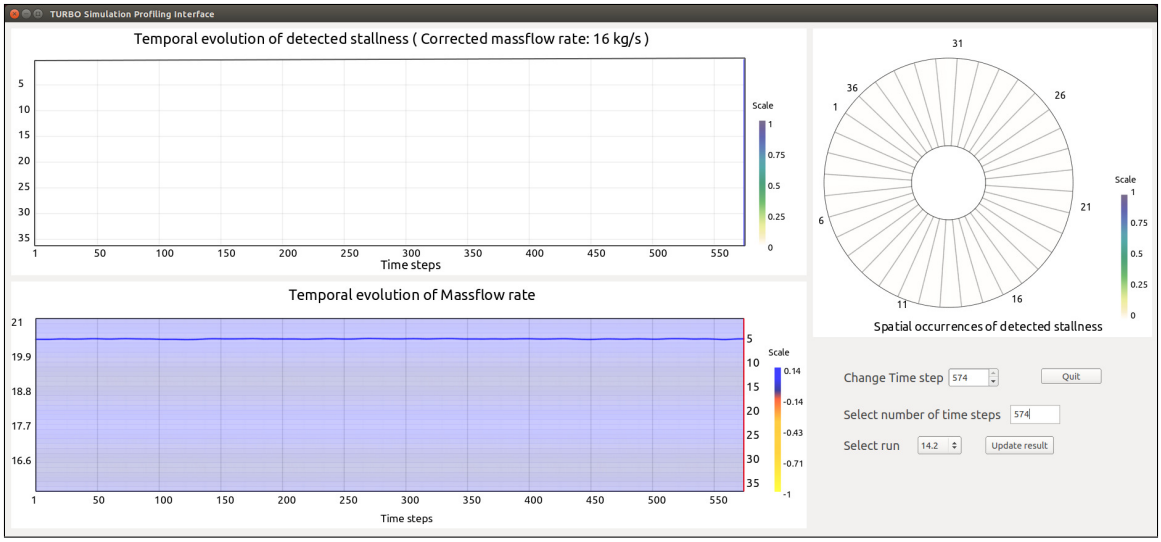


Figure 3.15: Result of simulation run with CMF=16.0 kg/s. The outcome of this run was known and resulted in a stable case which is observed from uniform mass flow chart and clean stallness plot.

our technique, the expert can now perform ensemble parameter studies in a timely manner without worrying about the expensive I/O and large-scale data management issues. By studying the stall patterns generated by CMF = 13.8 kg/s (Figure 3.6), CMF = 14.0 kg/s (Figure 3.10), and CMF = 14.2 kg/s (Figure 3.11) runs, it is observed that the local mass flow shows earlier signs of a potential stall for the 14.0 and 14.2 runs, whereas, for the 13.8 case, the indication of stall comes quite late. In contrast, the proposed fuzzy system based stall detection is found to be superior in detecting earlier signs of stall compared to the local mass flow for all of these cases. It is found that the stall can develop in different passage ranges and impact a different number of passages as well. In contrast, the stable runs produce clean stallness plots and uniform mass flow charts for 14.5, 14.8, and 16.0 cases. Through our study, the unexplored range of the CMF parameter while searching for the true stall point is reduced from 14.2 – 16.0 kg/s to a much smaller range of 14.2 – 14.5

kg/s which reduces the uncertainty in the outcome of the simulation for the expert. The expert can use this knowledge for model refinement, and more importantly, tune the throttle setting for a safer operation of the engine.

Discussion of the results. The above results with thorough expert evaluation and the performance study demonstrate the efficacy of our method in robust stall analysis in large-scale flow data sets. Here, we propose an off-line learning and *in situ* prediction based analysis pathway. In the off-line phase, the fuzzy system learns the feature specific multivariate patterns from the expert identified regions. After learning from a known data set, the inference algorithm is employed *in situ* on the unknown cases when the simulation is run using different parameter settings. By performing *in situ* feature detection and task-specific information aggregation, we overcome the expensive I/O bottleneck, and output only reduced information. By comparing and contrasting the feature-specific information, obtained from multiple runs, our technique reveals important details about the evolution of stall and its spatial spread. The local passage-wise mass flow also emerges as a more effective measure to locate the early flow asymmetry compared to the traditional global mass flow. Together with the spatial-temporal stallness charts, and local mass flow plots, our method is able to provide a comprehensive view of the simulation to the expert. We also derive new knowledge about the impact of the CMF parameter in stall inception that can potentially help in refining stall preventive technologies.

3.8 Limitations

The proposed fuzzy rule-based feature detection technique discussed above is easily generalizable to other multi-field data sets when a precise descriptor is unavailable, and a predictive algorithm is required. In this study, we have used the TURBO simulation

as a specific use case to show the effectiveness of the method. Furthermore, the *in situ* friendly nature makes this method suitable for feature extraction from very large-scale data sets in a timely manner which reduces the effort of data analysts significantly during the post-hoc analysis. However, this strategy inherently assumes that the target feature, which the scientists want to study from their simulation data, is predetermined beforehand. This assumption does not hold for many scientific applications where the scientists do not have the precise set of questions that they want to answer from their simulation data. Furthermore, a wide range of exploration tasks requires a significant amount of user interaction, hypothesis generation and verification which can only be done in a post-processing phase. In these cases, a pure *in situ* feature extraction and summarization scheme is not ideal since such techniques only store details about the predetermined features and cannot answer new queries. To remedy this, in the following chapter, we propose a more general purpose statistical data summarization scheme where the data is reduced using compact probability distribution functions in the *in situ* environment. We also show that such statistical data summaries can be used in the post-hoc analysis phase for performing various data analysis and visualization tasks with sufficient accuracy and can be used as a replacement for the raw simulation data.

3.9 Conclusion

We demonstrate the efficacy of an *in situ* fuzzy learning based exploration approach for robust feature analysis in large-scale simulations. Our method learns multivariate relations from the expert-highlighted regions in the data and generates several fuzzy rules which are then applied to the simulation with unknown parameter settings. Detection of the target feature for new simulations is done *in situ* which bypasses the expensive I/O and allows

exploration of data in a timely manner. The effectiveness of our approach is shown by applying it for the detection of the stall in large flow simulations.

Chapter 4: *In Situ* Distribution Guided Data Summarization for Flexible Post-hoc Analysis

4.1 Distribution-driven Data Summarization

In this Chapter, we describe the effectiveness of probabilistic data models for analyzing scientific data sets and present the distribution-based data modeling schemes proposed in this dissertation. Distributions of data can be estimated by following different modeling techniques, and each of these methods has their benefits and limitations. Therefore, choice of appropriate distribution modeling scheme is important and is usually done based on the requirements of the scientific applications. Given a scientific data set, a global distribution can capture the overall statistical properties of the data and such a global distribution modeling has been used successfully in the past for many visualization applications. However, often scientists want to analyze local regions in the data for identifying salient features and regions of interest (ROI). In this context, a local region-based distribution model is more suitable where the data domain is first decomposed into smaller sub-regions, and then for each region, the data is summarized using a distribution. While the storage of global distribution-based data summaries is significantly smaller compared to the raw data, the memory requirement of local region-wise distribution-based data summaries can be high

as each for each region, each data variable will have its own distribution. Therefore, it is essential to use appropriate distribution modeling schemes for each of these methods.

Distribution modeling techniques can be broadly classified into two categories: (a) non-parametric distribution models; and (b) parametric distribution models. Histogram and Kernel Density Estimators (KDE) are popular non-parametric distribution models used extensively in visualization community, whereas, parametric distributions such as Gaussian distributions, Gaussian Mixture Models (GMM) are also found to be effective in data analysis. In our work, we aim at a compact representation of data distributions with (1) small memory footprint and (2) fast computation time. The efficient computation capability enables the distribution estimation method to be employed *in situ*, while the simulation runs. This will ensure that the *in situ* analysis time is only a small fraction of the actual simulation time which will not cause additional burden to the simulation. Furthermore, the size of the output data from these extreme-scale simulations makes it prohibitive to store all the raw data into the disks. The compact representation of the distribution-based data summaries, in this case, helps to reduce the size of the output data significantly while preserving the important statistical data properties. Our study finds that, when the size of the raw data is too big, the local region-wise distribution-based data summaries can be used as a replacement of the raw data, and can be used efficiently for solving important visualization problems.

4.2 Distribution-based Data Modeling Schemes

4.2.1 Non-parametric Distribution Models

Given a set of discrete samples $\{s_i\}$, a non-parametric distribution in the form of a histogram can be formally defined as:

$$H(s) = \sum_i \delta(s - s_i) \tag{4.1}$$

where δ is the Dirac delta function defined as follows:

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

The area under a histogram is normalized, and such histograms are used for estimating probabilities of data values. It is to be noted, that the computation time for histograms is very fast, as it only requires a scan of data values and counting the frequencies of discretized data values by converting them into bins. However, the storage cost of histograms is not small, since the frequency of each bin needs to be stored. Another non-parametric distribution model KDE formally defined as:

$$F(s) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{s - s_i}{h}) \tag{4.3}$$

where n is the number of samples, h is the bandwidth, and $K(\cdot)$ is the non-negative kernel function. A range of kernel functions such as uniform, triangular, Gaussian, Epanechnikov kernels can be used for estimating data density. The computation cost of KDE is comparatively much higher than histograms, and hence, is not suitable for applications which require fast computation time.

4.2.2 Parametric Distribution Models

Parametric distribution models, on the other hand, offer a more compact distribution representation, since, only the parameters of the models are stored. Use of Gaussian distributions for data modeling is widely known across various scientific domains. However, the assumption of normality of data is not always true and can introduce modeling errors. In

contrast, Gaussian mixture models (GMM), removes this normality assumption by modeling data as a combination of several Gaussian distributions. The storage for a GMM is also smaller compared to the non-parametric models, since, only the parameters of the Gaussian distributions and their weights are stored. Formally, the probability density $p(X)$ of a GMM for a random variable X is expressed as:

$$p(X) = \sum_{i=1}^{K} \omega_i * \mathcal{N}(X|\mu_i, \sigma_i) \tag{4.4}$$

where K is the number of Gaussian components. ω_i, μ_i and, σ_i are the weight, mean, and standard deviation for the i^{th} Gaussian component respectively. It is to be noted that the sum of weights in the mixture, $\sum_{i=1}^{K} \omega_i$ is always equal to 1. Computation of parameters for the GMMs is typically done by Expectation Maximization (EM), which uses an iterative approach to maximize a likelihood function [13]. For an approximate and computationally efficient estimation of parameters of a GMM, an alternative incremental estimation scheme is available [127, 132]. Below we discuss these two methods of GMM estimation.

4.2.2.1 GMM Estimation using Expectation Maximization

Estimation of parameters of a GMM from the given sample points can be done using the Expectation Maximization (EM) algorithm [13]. This algorithm computes the parameters of a GMM by maximizing a likelihood function using the sample data points. Let us assume that $\chi = \{x_1, x_2, ...x_n\}$ are a set of i.i.d. samples, and θ is the set of parameters. Therefore, the resulting density for the samples $p(\chi|\theta)$ can be expressed as:

$$p(\chi|\theta) = \Pi_{i=1}^{n} p(x_i|\theta) = L(\theta|\chi) \tag{4.5}$$

Here, $L(\theta|\chi)$ is called the likelihood function, i.e., the likelihood of parameter set θ given the sample data χ. The EM algorithm maximizes this likelihood function and finds $\theta^{\star}$ where,

$$\theta^{\star} = argmax_{\theta} \; L(\theta|\chi) \tag{4.6}$$

The initial parameters of the EM algorithm can be randomly selected. However, several previous works have used k-means algorithm to find out the initial parameters for the EM algorithm which reduced the overall convergence time improving the performance.

4.2.2.2 GMM Estimation using Incremental Update Scheme

An approximate scheme of parameter learning for GMM estimation was proposed in [132]. This computes the GMM parameters from an initial state by modifying the old parameters based on the sample data points. Such a modeling scheme increases the computational efficacy of the distribution estimation and makes it suitable for applications with large-scale data sets. Hence, for estimating GMMs of large-scale data sets, the initial estimation of the parameters is done using the traditional EM algorithm for the initial time step only. From next time step, the incremental scheme is applied for updating the parameters of the existing GMMs to obtain new GMMs.

While updating, every new data point is checked against the existing K Gaussians. A positive match is found if a data point lies within the 2.5 standard deviation of a Gaussian. If multiple matches are found, then the best matched Gaussian is selected, which is the Gaussian with the minimum matched value. If none of the K Gaussians match the current data value, then the least probable distribution in the model is replaced with a new Gaussian with the current data value as the mean, an initial high standard deviation, and a low weight.

The weight at time t for the i^{th} mixture is adjusted as:

$$\omega_{i,t} = (1-\beta)\omega_{i,t-1} + \beta(I_{i,t}), \;\; i \in \{1,2...K\} \tag{4.7}$$

β is called the learning rate and the value of $I_{i,t}$ is 1 for the distribution with the best match and 0 otherwise. After the adjustment, all the weights are normalized again for maintaining consistency. The μ and σ parameters for the unmatched distributions remain the same, however, for the matched distribution they are updated as:

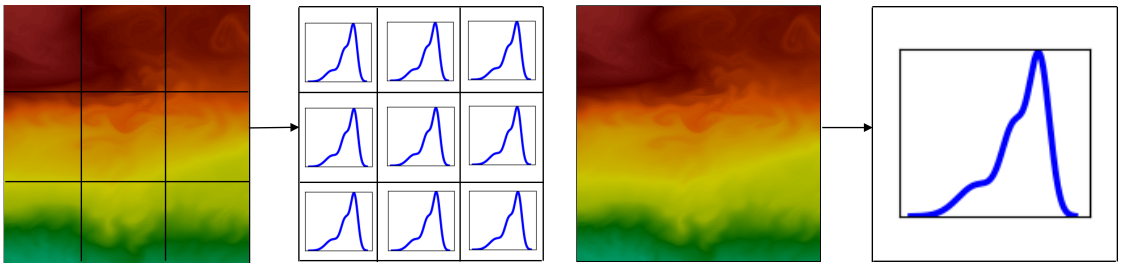
$$\mu_{i,t} = (1-\beta)\mu_{i,t-1} + \beta x_{i,t} \tag{4.8}$$

$$\sigma^2_{i,t} = (1-\beta)\sigma^2_{i,t-1} + \beta(\mu_{i,t} - x_{i,t})^2 \tag{4.9}$$

Once we have observed all the points, the GMM will give us the updated distribution. It is evident that the model adapts with the new observed data since it adds or removes Gaussians from the existing model as required.

4.3 Local Region-based Probabilistic Data Models

Local region-based probabilistic data models convert raw data into a statistical representation, which preserves the local properties of the data via storing their distributions (see Figure 4.1a). Keeping the large size of time-varying data in mind, a block-wise partitioning of data for analysis is adopted. The whole data space is partitioned into smaller non-overlapping blocks. Then the data points in each block/partition are represented by a probability distribution function. More specifically, each data variable is represented by its own separate probability distribution (either parametric or non-parametric). Such a block-wise approach is often employed in computer vision and video processing applications for exploiting the spatial coherency in data at a reduced computational complexity. Instead of analyzing at individual point level, the approach uses data blocks as analysis units



(a) Local distribution-based data model. Data values for all the points inside each block is represented by a probability distribution. For each data variable and for each local block, a separate distribution is created.

(b) Global distribution-based data model. Data values for all the points is represented by a probability distribution. For each data variable, a separate distribution is created.

Figure 4.1: Local and Global distribution-based data modeling schemes.

resulting in computation cost reduction without losing too much information, particularly when the size of the data is very big [146]. Oftentimes, scientific visualization tasks involve exploration of specific phenomena, defined as features, which are found to be spatially connected regions in the data set. In this case, using local region-based analysis allows efficient identification and isolation of such features. Furthermore, in some domain-specific applications, scientists specifically desire to look at the characteristics of local regions for temporal event discovery, instead of individual points. This is because the behavior of individual points cannot display the phenomenon robustly, and may capture false positives during automatic event detection.

The benefits of local region-based probabilistic data models can be fourfold (1) Representing a block of data with a probability distribution can preserve the block's statistical properties well, which allow efficient feature analysis; (2) A compact distribution-based data model is able to reduce the size of the data significantly which enables flexible and

scalable exploration of extreme-scale data sets; (3) Uncertainty quantification during analysis becomes possible which enriches verifiable visualization [135]; (4) By sampling the distributions, a statistical realization of the raw data can be constructed and visualized for exploration [29, 55, 89]. An ideal local region-based statistical data summarization scheme aims at preserving the statistical properties of the data as much as possible with a compact representation. Therefore, for achieving a compact-yet-accurate data representation, a region partitioning scheme produces partitions with coherent data values, such that, efficient distribution-driven summarization is possible. In this context, note that, while obtaining a regular partitioning of the data is trivial, it does not consider any inherent spatial data coherency. As a result, many data blocks will have high data value variation resulting lower accuracy in sampling and higher uncertainty during visualization. Furthermore, as regular partitioning does not consider data homogeneity during decomposition, visualization will introduce artifacts and discontinuities on block boundaries, making the visualization less effective. Hence, there is a growing need for more accurate and efficient statistical data summarization techniques.

In this dissertation, we propose several local region-based statistical data summarization technique using distributions. A simple data partitioning scheme would be the regular partitioning of data where the data domain is divided into regular data blocks. Another data partitioning scheme we employ is based on K-d tree partitioning. A more accurate third data partitioning technique, we propose, partitions the data by its spatial coherency and aims to reduce the uncertainty of all partitions. In this scheme, to partition the data into local regions, we make use of SLIC (Simple Linear Iterative Clustering) algorithm [1], which was used for generating super-pixels and super-voxels [1, 161]. This minimizes the data value variance in each spatial partition and hence, each region/partition can be compactly summarized

using a probability distribution function which preserves the statistical properties of the data efficiently. To achieve this, we propose a hybrid scheme of distribution-based summarization by using either a single Gaussian or a mixture of Gaussians (GMM) per partition. Advantages of using GMMs as a compact parametric distribution representation over other alternatives such as histograms and Kernel Density Estimators (KDE) have been discussed previously in [89]. Furthermore, GMMs also have been shown to be effective for probabilistic data classification [89, 149].

For evaluating the efficacy of the proposed technique, we conduct extensive quantitative and qualitative studies among (a) The SLIC-based method; (b) Regular partitioning; and (c) K-d tree partitioning. For each of these partitioning methods, the data summarization is done using the aforementioned hybrid summarization scheme. The results demonstrate that: (a) Both quantitatively and qualitatively, the SLIC-based summarization produces superior statistical sampling-based data reconstruction with best storage-to-quality trade-off; (b) More precise distribution similarity-based feature matching, where identified features are free from boundary discontinuities and other artifacts, which often arise from regular or k-d partitioning scheme. We also demonstrate that the SLIC-based method is also suitable for *in situ* summarization, by running the proposed scheme directly with a large-scale CFD simulation, resulting in an improved distribution-based data summarization.

4.3.1 Different Data Partitioning Schemes for Local-region based Data Summarization

4.3.1.1 Regular Block-wise Partitioning

Regular block-wise partitioning decomposes the domain into equal sized blocks of predetermined dimensions. Due to the simplicity and computational efficacy, this scheme has been widely adopted in many of the previous works involving local region-based data

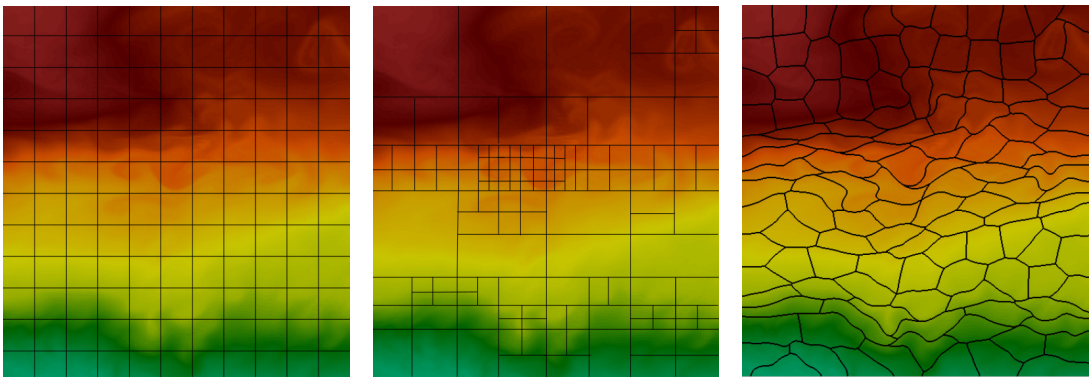
analysis. A regular partitioning of a 2D data is shown in Figure 4.2a as an illustrative example. Observe that, this scheme does not consider data properties while dividing the domain into smaller sub-regions. As a result, some partitions will contain data points with a high data value variation. Consequently, statistical summaries estimated from the data points of such blocks will have a higher value spread. Analysis using these summaries will lead to higher uncertainty since the samples drawn from those block distributions will contain large errors.

4.3.1.2 K-d Tree Partitioning

To obtain a more homogeneous decomposition of data, partitioning using k-d trees can be performed. As was discussed in [102], a recursive partitioning of the domain can be obtained by following a top-down subdivision scheme with appropriate termination criterion. Here, we use the information theoretic measure entropy [38] to measure the randomness of a partition. In information theory, the value of Shannon entropy is regarded as a measure of the information content of a probability distribution of a random variable. Formally, information entropy $H(X)$ is measured as:

$$H(X) = -\sum_{i=1}^{n} prob(x_i) log(prob(x_i)) \tag{4.10}$$

where n is the total number of data points in the partition, $prob(x_i)$ is the probability of data value x_i. It is observed that Shannon entropy increases when the spread of a distribution is higher, i.e., the distribution contains a wide range of data values. In the k-d partitioning scheme, the entropy of the data values of each partition is checked against a predefined threshold value. If the entropy of the partition is higher than the threshold, then the partition is divided into smaller sub-regions to refine the region and reduce the variation. As a result,



(a) Regular block-wise partitioning. (b) K-d tree based decomposition. (c) SLIC-based partitioning.

Figure 4.2: Different types of data partitioning schemes.

it guarantees that all the partitions satisfy the predefined homogeneity criterion. An example of this scheme is demonstrated in Figure 4.2b using a 2D data set. It can be seen that the regions with higher variation have been refined more.

4.3.1.3 SLIC-based Partitioning

In this work, we employ a clustering-based data partitioning scheme using a variant of k-means algorithm, called SLIC (Simple Linear Iterative Clustering) [1]. SLIC originally was designed for the generation of superpixels in images and was also used successfully for the generation of supervoxels in 3D data sets [1, 161]. The fast execution time and state-of-the-art clustering quality make SLIC a suitable choice in our work. Each cluster/supervoxel generated by SLIC is treated as a partition in this work.

Motivation for using SLIC. Compared to traditional k-means clustering, SLIC adopts a local neighborhood-based approach, where similar data points within a local neighborhood are grouped into one cluster. During the optimization stage, from each cluster center, distances only to the points in the predefined neighborhood are compared. This reduces the

total number of distance computations significantly by limiting search in a local window. As a result, the algorithm performance is boosted significantly. Furthermore, SLIC uses a weighted distance measure that provides contributions from both the spatial locality of the data points and their scalar value similarities. Due to these properties, SLIC partitions the data domain into smaller sub-regions where each partition contains points which are: (a) spatially as contiguous as possible; and (b) homogeneous in value domain. In Figure 4.2c, we show an illustrative example of SLIC algorithm applied on a 2D image. As shown, SLIC partitions similar valued data points along non-axis aligned boundaries compared to the methods shown in Figure 4.2a and 4.2b. We later demonstrate that summarization using distributions of SLIC partitions achieves superior quality than the previously described partitioning schemes. Below, we briefly discuss the SLIC algorithm.

SLIC algorithm. SLIC requires the user to only provide the expected approximate size of the spatial clusters/partitions. Assuming that the user has provided the spatial size of the partitions as $p \times q \times r$, and if the dimension of the data is $X \times Y \times Z$, then the number of partitions K can be estimated as $K = (X \times Y \times Z)/(p \times q \times r)$. For finding the K initial cluster centers, the entire data domain is divided into $p \times q \times r$ sized blocks, and the center of each block is selected as its initial cluster center. In the cluster assignment step, each voxel is associated with the nearest cluster center whose search region overlaps with the voxel's spatial location. Since the expected size of a cluster is $p \times q \times r$, the search for similar voxels is done within a volume $2p \times 2q \times 2r$ around each cluster center. This local region-based search during the clustering reduces the total number of distance computations significantly compared to the traditional k-means algorithm, resulting in an overall speed up. Similar to the K-means algorithm, SLIC is an iterative clustering algorithm. During each iteration (a) Each voxel is associated to its nearest most similar cluster; (b) The cluster

centers are recalculated with the updated cluster assignments. For each iteration of SLIC, the difference δ, between the current cluster centers and the previous cluster centers are computed using the L_2 norm between all the cluster centers. If the value of δ is higher than a predefined threshold value, the algorithm moves to its next iteration, otherwise, when δ becomes lower than the threshold, the algorithm terminates.

It is to be noted that, by restricting the search window into a local region for every cluster center, the time complexity of SLIC is significantly reduced compared to a traditional k-means algorithm. The complexity of a k-means algorithm scales with $O(kN)$, whereas, SLIC scales with $O(N)$ [1] for each iteration of the algorithm. This improved computation complexity makes SLIC applicable to large data sets [161] and also attractive for *in situ* environments, where performance is an important factor.

Distance measure. The distance measure used in this algorithm is similar to as was used in [161], and is defined as:

$$dist(i,j) = \alpha \cdot ||C_i - P_j||_2 + (1-\alpha) \cdot |val_i - val_j| \tag{4.11}$$

Here, C_i is the location of the cluster center i and P_j is the location of point j. val_i and val_j are the scalar values at i^{th} cluster center and j^{th} data point respectively. The mixing weight α is configured based on the importance of spatial vs value components, such that $0 <= \alpha <= 1$, and $\alpha + (1-\alpha) = 1$. Smaller values of alpha will produce higher weightage on the difference of data values than their spatial locations. In Equation 4.11, as data values and spatial locations can be scaled inconsistently, we normalize the data and normalize spatial distances using the block length to achieve a consistent distance measure.

4.3.2 Distribution-Driven Data Summarization

The goal of the summarization is to achieve a compact and storage efficient statistical data representation which preserves important data properties. For a large number data partitions created by techniques presented in the previous section, a single Gaussian may be a sufficiently accurate representation. Hence, to reduce the storage cost of the distribution-based data summary, we advocate for a hybrid distribution-based data representation scheme. We perform a statistical normality test, D'Agostino's K-squared test [40], on each of the partitions. This test provides a goodness-of-fit measure of departure from normality given the set of data points in a partition. The method uses both kurtosis and skewness to detect the deviation from normality. If a partition satisfies the normality criteria, only a single Gaussian is used to summarize it, otherwise, a GMM is estimated for modeling the data in the partition. By using this hybrid distribution summarization scheme, (i.e., Gaussians and GMMs), we achieve a compact statistical summarization of the data without sacrificing the information content of the data. Another advantage is the potential reduction in computation cost in creating the distributions for each partition because the Expectation Maximization algorithm for calculation of a GMM is computationally costlier compared to estimating the parameters of a single Gaussian distribution. Therefore, if more partitions satisfy the normality test, then the overall computational cost will be reduced since fewer partitions will employ the EM algorithm. So, by generating coherent and homogeneous partitions, that have low variance via the SLIC method, we can reduce the computation cost and storage of partitions via our hybrid GMM and Gaussian representations.

Algorithm 1 presents the proposed method of statistical data summarization. As discussed above, instead of just a regular partitioning scheme, we employ SLIC for generating the data-driven partitions. Then each partition is tested for normality. If the partition satisfies

Algorithm 1 SLIC-based Statistical Data Summarization

```
1: Input: Raw data, user specified initial partition dimensions.
2: Output: Local distribution-based compact summary data.
3: Initialize cluster/partition centers uniformly over data domain.
4: Compute SLIC for partition generation.
5: for all p in Partitions do
6:     Perform D'Agostino's K-squared normality test.
7:     if (p satisfies normality test) then
8:         summarize p using a single Gaussian distribution.
9:     else
10:        summarize p using a GMM.
11:    end if
12: end for
```

the test, a single Gaussian is used for representing the partition, otherwise, a GMM is computed for summarizing it. The final output is reduced and compact hybrid distribution-based data summary. For our following comparisons, we change the step 4 of the above algorithm with a different (regular or k-d tree) partitioning scheme.

4.3.3 Comparative Study among Different Partitioning-based Data Summarization Techniques

We provide a comprehensive comparative study of the three partitioning methods and demonstrate the efficacy of our proposed method. We consider both storage cost and quality of statistical summarization while comparing these methods. For comparing the quality of statistical summarization, we use sampling-based data reconstruction and visualization as one of our tasks via stochastic sampling-based methodologies for data analysis [92]. We perform sampling on the distribution-based summary data for creating a statistical realization of the raw data. It follows that, with a better quality of the statistical summarization, it will result in a more accurate realization of data with better quality of samples [80]. We employ Monte Carlo sampling for generating a realization of the raw data, as was used in [55, 89].

To estimate the quality of this sampling-based reconstruction, we use Signal-to-Noise Ratio (SNR) for quality comparison. SNR is defined as the dimensionless ratio of the power of a signal to the power of noise in the signal. Hence, higher values of SNR signify better quality. Formally, SNR is defined as:

$$SNR = \frac{P_{signal}}{P_{noise}} \tag{4.12}$$

where the power of noise is measured by the variance of the error in the reconstructed data. A higher variance of error will decrease the value of SNR. We use the logarithmic decibel scale for SNR:

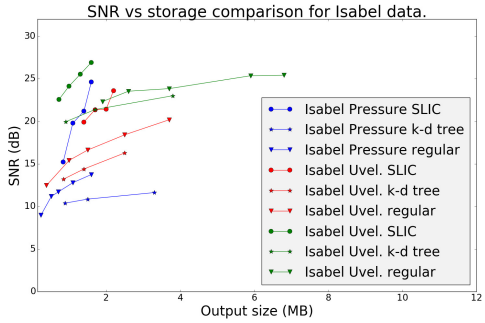
$$SNR_{dB} = 10 \cdot log_{10}(SNR) \tag{4.13}$$

Storage format for different partitioning schemes. For the regular block-wise partitioning scheme, we only store the estimated distribution parameters, i.e., the parameters of the Gaussian distributions (mean and standard deviation) and the parameters of GMMs (means, standard deviations, and weights). Each GMM consists of 3 Gaussian distributions in these experiments. For a partition with a single Gaussian, we keep two floating points for its parameters, and for a partition with a GMM, we use 9 floating points for storing the parameters. Also, we keep a GMM/Gaussian flag for each partition. In case of the k-d tree partitioning scheme, we additionally need to store the ids of two corner point locations of the bounding box for each partition which is stored as integers.

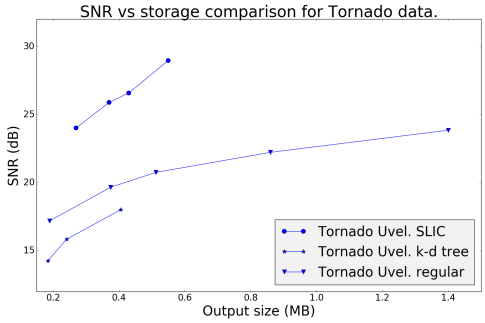
The proposed SLIC-based method generates irregular partitions and we keep the cluster ids per point as the additional information. Our method is designed for a distributed memory environment and the general assumption is that each node will only process a small subset of data. Furthermore, as we create large homogeneous partitions using SLIC, the range of the cluster ids for each processing node is relatively small. So, we use unsigned shorts for storing

Table 4.1: Experimental results of storage vs SNR (quality) for regular partitioning, k-d tree partitioning, and the proposed SLIC-based partitioning scheme with different parameter configurations. A specific parameter configuration is highlighted in bold from each of the three methods. By observing these three storage vs SNR results, it can be seen that our proposed method achieves superior storage-vs-quality trade-off.

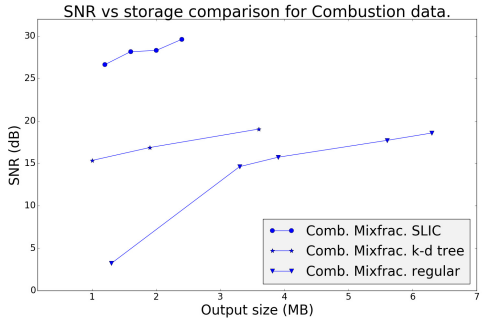
		Regular block partitioning scheme						**K-d tree partitioning scheme**						**SLIC-based partitioning scheme**					
Data set	Raw data size (MB)	**Storage (block size = 3x3x3) (MB)**	**SNR (dB)**	Storage (block size = 4x4x4) (MB)	SNR (dB)	Storage (block size = 5x5x5) (MB)	SNR (dB)	**Storage (entropy th=1.2) (MB)**	**SNR (dB)**	Storage (entropy th=1.5) (MB)	SNR (dB)	Storage (entropy th=1.8) (MB)	SNR (dB)	Storage (Aprx. 5x5x5 points per cluster) (MB)	SNR (dB)	**Storage (Aprx. 6x6x6 points per cluster) (MB)**	**SNR (dB)**	Storage (Aprx. 7x7x7 points per cluster) (MB)	SNR (dB)
Isabel Pressure	12.5	**3.2**	**11.63**	1.5	10.87	0.89	10.39	**1.6**	**13.75**	1.1	12.79	0.71	11.72	1.6	24.63	**1.4**	**21.20**	1.1	19.78
Isabel Uvel	12.5	**2.5**	**16.29**	1.4	14.40	0.85	13.21	**3.7**	**20.18**	2.5	18.42	1.5	16.63	2.2	23.61	**2**	**23.11**	1.7	21.37
Isabel Temperature	12.5	**3.8**	**23.01**	1.7	21.39	0.91	19.96	**6.8**	**25.42**	5.9	25.37	3.7	23.84	1.6	26.92	**1.3**	**25.52**	1	24.14
Tornado Uvel	3.5	**0.41**	**18.01**	0.24	15.84	0.18	14.27	**1.4**	**23.85**	0.86	22.22	0.51	20.76	0.55	28.97	**0.43**	**26.58**	0.37	25.89
Combustion	20.7	**3.6**	**19.05**	1.9	16.87	1.0	15.35	**6.3**	**18.58**	5.6	17.71	3.9	15.73	2.4	29.63	**2**	**28.31**	1.6	28.17
Vortex	8.4	**1.2**	**11.91**	0.90	9.87	0.57	6.37	**7.7**	**19.79**	6.8	17.95	4.7	17.95	1.9	22.68	**1.6**	**21.73**	1.4	20.70



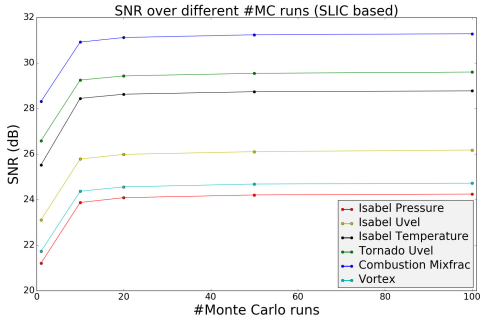
(a) Storage vs SNR comparison of Isabel data.



(b) Storage vs SNR comparison of Tornado data.



(c) Storage vs SNR comparison of Combustion data.



(d) SNR vs #MC runs for SLIC-based summarization scheme.

Figure 4.3: Figures 4.3a-4.3c present storage vs SNR comparison for different data sets. It is observed that using equal or lower storage, proposed SLIC-based method is able to produce better Monte Carlo sampling-based data reconstruction. Figure 4.3d shows the trend of SNR values with different number of Monte Carlo runs. It can be seen that the SNR values saturate after around 20 Monte Carlo runs. This trend is similar for all the summarization schemes discussed.

the cluster ids which reduces the storage overhead. The point ids for k-d tree partitioning and the cluster ids for the SLIC-based scheme are both stored using Zlib compression for further storage reduction.

Storage vs SNR results. The performance of storage vs quality of statistical summarization of (a) Regular partition based scheme; (b) K-d tree-based scheme; and (c) The proposed SLIC-based scheme are presented in Table 4.1. We have tested several data sets, described

later in our case studies, for conducting these experiments. It is to be noted that, when the partitions are smaller, they are more likely to become more homogeneous compared to bigger partitions. This results in a higher quality of statistical summarization using smaller partitions, but the storage increases. This trend is common for all the 3 methods. The SNR is higher when the size of the partitions are smaller, while the storage is also higher.

The quantitative results of the experiments for all the methods with different parameter configurations are provided in Table 4.1. As can be seen, we change the block size for regular partitioning scheme to vary the number partitions and measure the quality of reconstruction in each case by measuring the SNR. In case of the k-d partitioning scheme, we vary the entropy threshold value to obtain a different number of partitions. It is to be noted that, making the entropy threshold higher will result in a decrease of the number of partitions, as well as the storages. However, the SNR will also decrease. Finally, for our proposed scheme, we are able to use bigger partitions (smaller storage) and yet achieve better sampling-based reconstruction quality. The number of clusters is varied by changing the number of points in each cluster. In Table 4.1, we have highlighted a pair of storage and SNR columns in bold from each of the methods. By comparing these three selected configurations, it can be easily observed that the proposed method achieves superior storage-to-quality trade-off by producing higher SNR values for all the data sets while using less or comparable storage.

Comparative study among different methods. By using the data presented in Table 4.1, a line chart based comparison of these three partitioning schemes are presented in Figures 4.3a-4.3c. Results of different data sets are shown in separate charts. By studying these 3 charts, a common observation can be made that the proposed SLIC-based technique produces better sampling-based reconstruction of data while using comparable or less storage. It can be seen that by increasing the value of entropy threshold, the storage of k-d

tree-based scheme can be reduced, however, as mentioned above, it will reduce the SNR, i.e., the reconstruction quality as well. Similarly, we can also create bigger partitions for achieving a better storage in regular block partitioning by sacrificing accuracy. Hence, from Figures 4.3a-4.3c, we find that, for similar output storage cost, the proposed method gives superior analysis accuracy among the three methods.

Effect of different numbers of Monte Carlo runs on sampling quality. Since we employ Monte Carlo sampling for generating a realization of data from the distribution-based summaries, we further study the effect of different numbers of Monte Carlo runs on the quality of sampling. Ideally, as was shown in [55], a higher number of Monte Carlo runs would make the reconstruction smoother and the reconstruction quality will increase and eventually will saturate. In Figure 4.3d, we show that by increasing the number of Monte Carlo runs, and taking an average over all the runs while reconstructing, the reconstruction quality indeed improves. Also, after around 20 Monte Carlo runs, the increase in quality saturates. This trend is observed for all the three summarization schemes.

Comparison of summarization using: (a) Gaussian only; (b) GMM only; and (c) Hybrid scheme. For the same number of partitions, using only Gaussians for summarization will result in the smallest storage, and using only GMMs will need the highest storage. It is expected that the reconstruction quality will be similar for hybrid scheme vs only GMMs. In Table 4.2, we show the results of the SLIC-based method (with approximately $6 \times 6 \times 6$ points per cluster) when it is: (a) Gaussian only; (b) GMMs only; and (c) Hybrid distribution-based scheme; using equal number of partitions for each. We find that the sampling-based reconstruction using only GMMs is very similar to the Hybrid scheme. However, when only Gaussians are used, the quality decreases slightly, but it is still superior when compared to the regular partitioning and k-d tree partitioning schemes. This shows that

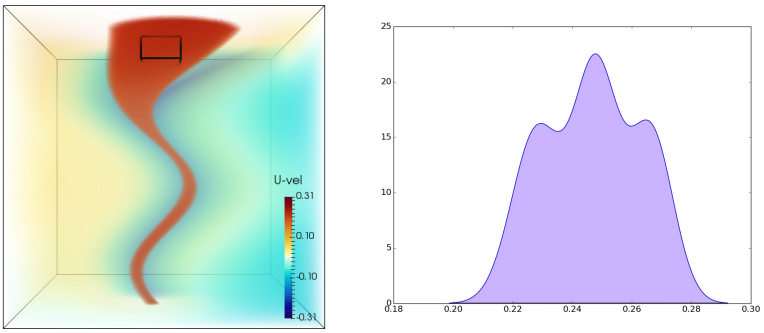
Table 4.2: Comparison of SNR using fixed number of partitions (approx. $6 \times 6 \times 6$ points per cluster) for the SLIC-based scheme when: (a) only Gaussian distributions; (b) only GMM; and (c) Hybrid (Gaussian + GMM) distribution scheme are used for summarization.

Data set	Raw Size (MB)	Gaussian only SNR (dB)	GMM only SNR (dB)	Hybrid (Gaussian + GMM) SNR (dB)
Isabel Pressure	12.5	21.15	21.35	21.20
Isabel Uvelocity	12.5	22.98	23.09	23.11
Isabel Temperature	12.5	25.51	25.86	25.52
Tornado Uvelocity	3.5	26.16	26.85	26.58
Combustion	20.7	28.01	28.42	28.32
Vortex	8.4	21.22	21.72	21.73

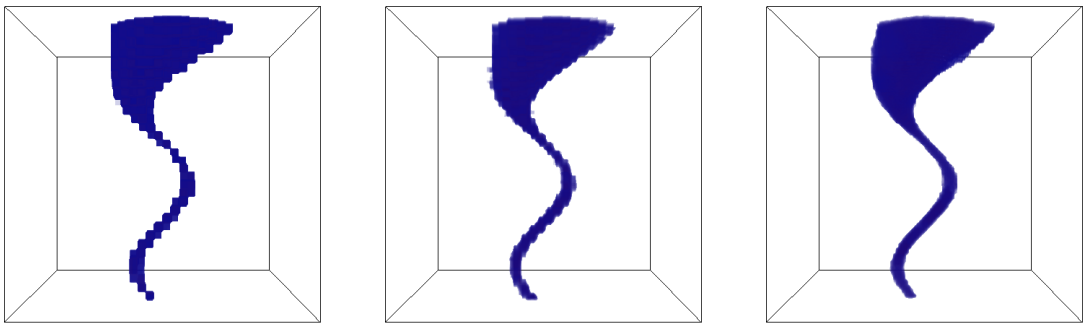
the SLIC partitions are largely homogeneous and a single Gaussian-based summarization can be used per partition when further storage reduction is desired, without compromising the quality much.

4.3.4 Comparative Visual Study among Different Partitioning-driven Summarization Schemes

Distribution-based summary data can be used in two ways for analyzing scientific data sets. The first technique is by directly exploiting the local statistical properties of the data for distribution-driven classification and feature search. This method does not require any sampling and analyzes the distributions of the local regions directly for classification. The second method is using Monte Carlo sampling-based reconstruction of a statistical realization of the data for visual analytics. For all the visualizations ParaView [6] was used for rendering the results, and we used $3 \times 3 \times 3$ block-size for the regular-partitioning scheme, entropy threshold of 1.2 for k-d partitioning, and approximate partition size of $6 \times 6 \times 6$ points per partition for our SLIC-based method. In the experiments, we show



(a) Feature selection in Tornado data set. The estimated target feature distribution is shown on the right. The feature is modeled using a mixture of Gaussians.



(b) Distribution similarity-based identified feature using regular block partitioning.

(c) Distribution similarity-based identified feature using k-d tree partitioning.

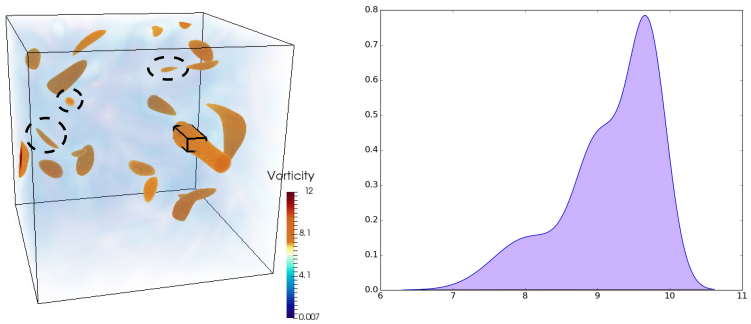
(d) Distribution similarity-based identified feature using our SLIC-based partitioning.

Figure 4.4: Distribution data driven probabilistic feature search in Tornado data set.

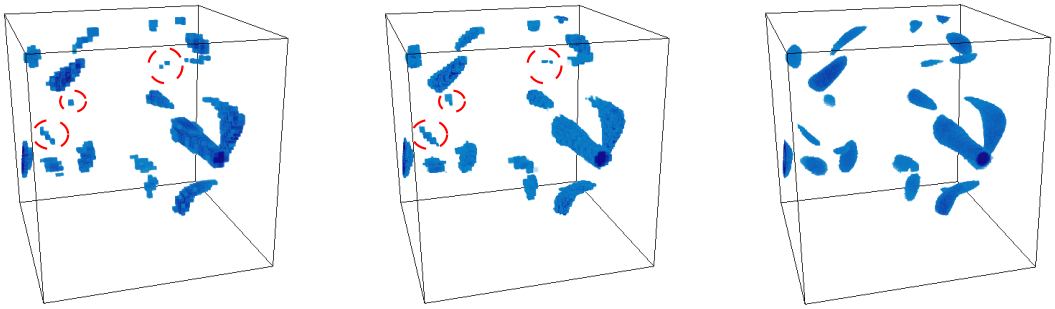
that, with lower or comparable storage cost, as reported previously in Table 4.1, our method produces better and more accurate visual quality.

4.3.4.1 Distribution-driven Feature Search

Scientific data sets contain features that are not well defined in the value domain and it is difficult to define such features using precise threshold values. Several previous works [93, 151] have shown the use of distributions to represent such features probabilistically. In the absence of a precise value range-based feature descriptor, stochastically-defined features



(a) Feature selection in Vortex data set. The estimated target feature distribution is shown on the right. The feature is modeled using a mixture of Gaussians.



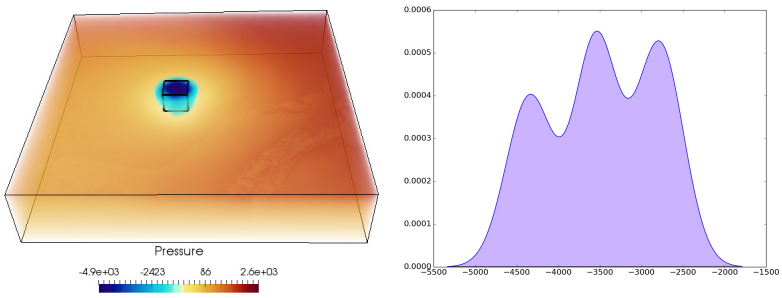
(b) Distribution similarity-based identified feature using regular block partitioning.

(c) Distribution similarity-based identified feature using k-d tree partitioning.

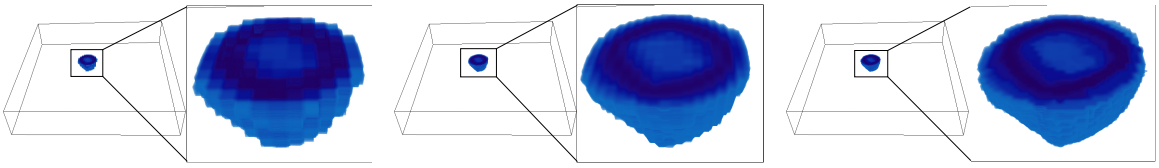
(d) Distribution similarity-based identified feature using our SLIC-based partitioning.

Figure 4.5: Distribution data driven probabilistic feature search in Vortex data set.

can be searched by using local distribution-based data summaries. Local regions (partitions) containing similar distributions that of the target distribution will be detected. Here we show that by using our distribution-based summarization, a more accurate and refined feature searching can be performed. For measuring the similarity between distributions, we use the Earth Mover's Distance (EMD), defined by the minimal transport effort to match two distribution shapes. To compute the EMD for 1D distributions, we use the *match distance* as



(a) Feature selection in Isabel data set. The estimated target feature distribution is shown on the right. The feature is modeled using a mixture of Gaussians.



(b) Distribution similarity-based identified feature using regular block partitioning.

(c) Distribution similarity-based identified feature using k-d tree partitioning.

(d) Distribution similarity-based identified feature using our SLIC-based partitioning.

Figure 4.6: Distribution data driven probabilistic feature search in Hurricane Isabel data set.

the ground distance [120], since, the EMD then can be estimated by the absolute difference between the cumulative distribution functions (CDF) of the distributions [154].

Feature Search in Tornado Data Set. Our first experiment studies feature searching in a Tornado data set with spatial dimensions of $96 \times 96 \times 96$, and velocity vectors at each grid point, generated by an analytical equation [39]. The data set has 50 time steps simulating a tornado-like vortex structure. For this case study, we use the U-velocity field.

As seen from Figure 4.4a, the high values of U-velocity provides a structure of the tornado, which is the feature of interest. For highlighting the region of interest, the users select a small 3D box in this region, as shown in Figure 4.4a to easily select their region of interest. We collect the data points in this box for defining the target feature distribution as a

GMM. The estimated feature GMM is shown on the right of Figure 4.4a. Using an user-specified fixed threshold of 0.1 on the normalized EMD-based distance field, the detected region is extracted and visualized from the statistical summary data. Figure 4.4b and 4.4c show the results obtained from regular partitioning scheme (block size = $3 \times 3 \times 3$), and k-d tree partitioning scheme (entropy threshold = 1.2). The identified regions contain block-like artifacts on the boundaries as observed from the results. In contrast, our SLIC-based method partitions data by its local homogeneity which preserves the feature boundaries more accurately.

Feature Search in Vortex Data Set. Our second case study shows the result of distribution-driven feature search in a Vortex data set, which is a pseudo-spectral simulation of coherence vortex structures. The spatial dimensions of this data set is $128 \times 128 \times 128$. The scalar field used in the data is vorticity magnitude containing several tubular vortex cores, which are the features of interest.

The high vorticity values roughly correspond to the vortex features, which can be seen in the Figure 4.5a. By following a similar technique as discussed in Section 4.3.4.1, the target feature GMM is obtained and shown on the right of Figure 4.5a. Compared to the previous Tornado case study, identifying features in this data set is not easy since there are many features which show similar data properties, and such features are located separately across the spatial domain as seen in Figure 4.5a. Furthermore, there are several vortex features, as shown by black dotted lines in Figure 4.5a, that are very small and hence, challenging to be detected. From the results presented in Figures 4.5b, 4.5c, and 4.5d, we see that our SLIC-based partitioning method is the best in extracting those vortex features among the three techniques. The EMD threshold of 0.23 was used for the extraction of the matched regions. Also, from Figures 4.5b and 4.5c, it is observed that both the regular block-wise

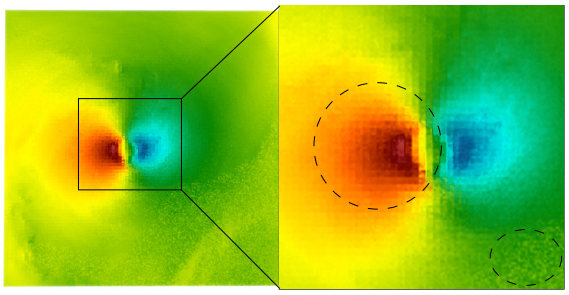
scheme and k-d tree-based scheme detect the small features less accurately compared to our proposed method. The shape of those small features gets distorted as highlighted by red dotted lines in Figures 4.5b and 4.5c, whereas the proposed method is able to identify these fine features with high accuracy.

Feature Search in Hurricane Isabel Data Set. Hurricane Isabel data is a multivariate time-varying data consisting of 13 scalar fields. The data set is a courtesy of NCAR and the U.S. National Science Foundation (NSF), and was created using the Weather Research and Forecast (WRF) model. The resolution of the grid for each time step is $250 \times 250 \times 50$ and there are total 48 time steps. In this study, we use the Pressure field of the data set.

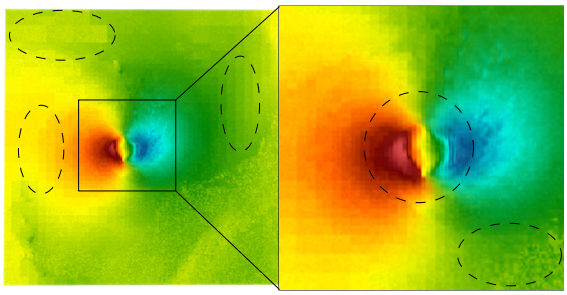
We use the low-pressure region which is known as the eye of the hurricane and is an important feature in the data. We show the selected region using a small 3D box and the estimated target distribution in Figure 4.6a. The results of the detected feature using different schemes are depicted in Figures 4.6b, 4.6c, and 4.6d. The EMD threshold of 0.25 was used for this experiment. From the zoomed view of the detected regions, we see that the block-like artifacts due to axis-aligned partitioning is visible in both Figures 4.6b and 4.6c on the boundaries. However, we obtain a much smoother and refined feature matching using our proposed summarization technique as observed from Figure 4.6d.

4.3.4.2 Sampling-based Data Visualization

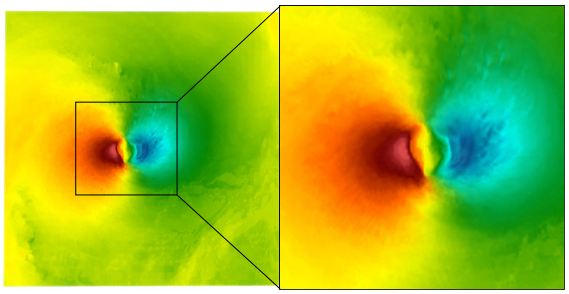
Monte Carlo sampling-based reconstruction for visualizing distribution-based data sets was used previously in [55, 89]. In this section, we demonstrate that by summarizing the data using our proposed scheme, a more accurate sampling-based visualization can be achieved compared to the other discussed methods. We use Zlib compression for storing the point ids in k-d partitioning, and cluster ids in SLIC-based partitioning. Decompression is



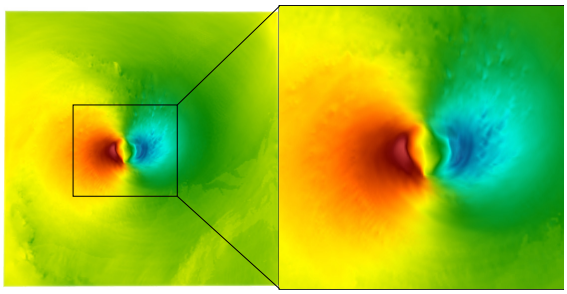
(a) Reconstruction using regular block partitioning scheme. The block size used is $3 \times 3 \times 3$. A zoomed view is shown on the right.



(b) Reconstruction using k-d tree based scheme. Entropy threshold of 1.2 is used for this experiment. A zoomed view is shown on the right.



(c) Reconstruction using proposed SLIC-based scheme. Relatively large cluster size (approx. $6 \times 6 \times 6$ points per cluster) is used. A zoomed view is shown on the right.

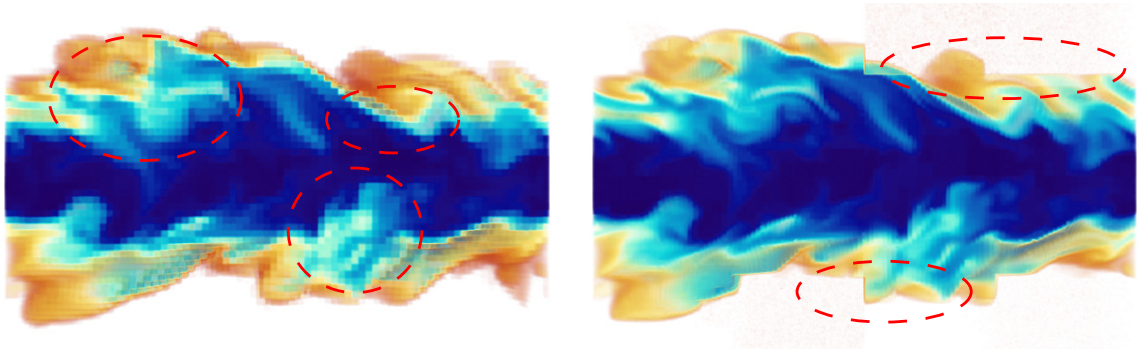


(d) Raw data (Ground truth). A zoomed view is shown on the right for better visual comparison.

Figure 4.7: Visual comparison of U-velocity of Hurricane Isabel data. The reconstructed fields are generated using Monte Carlo sampling of distribution-based summarized data.

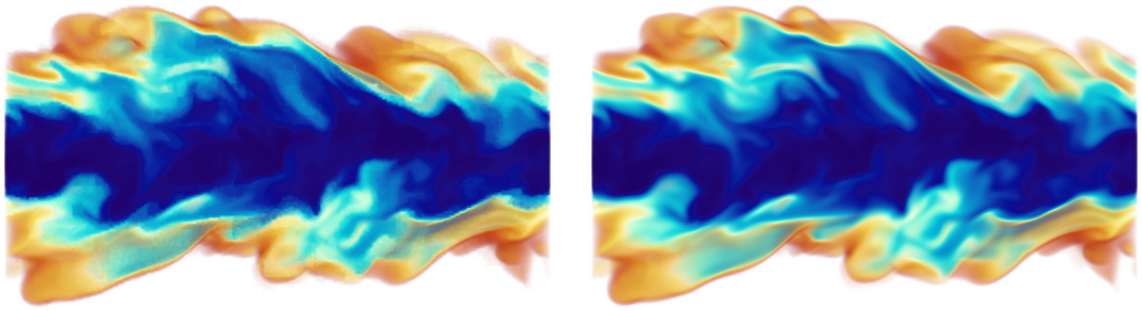
done in memory during runtime when cluster information for reconstruction is required for visualization.

Visual Analysis using Hurricane Isabel Data Set. This case study uses the U-velocity field of Hurricane Isabel data set, which was used earlier in Section 4.3.4.1. Figure 4.7 shows the volume rendered images from the reconstructed data using different methods. In Figure 4.7d we present the result of raw data and a zoomed view of the core region of the storm showing the high and low wind speed. This is an important region for U-velocity, since the wind velocity can reflect the power of the storm. As seen from the zoomed view on the right



(a) Reconstruction using regular block partitioning scheme. The block size used is $3 \times 3 \times 3$.

(b) Reconstruction using k-d tree based scheme. Entropy threshold of 1.2 is used for this experiment.



(c) Reconstruction using proposed SLIC-based scheme. Relatively large cluster size (approx. $6 \times 6 \times 6$ points per cluster) is used.

(d) Raw data (Ground truth).

Figure 4.8: Visual comparison of Mixture Fraction of Combustion data. The reconstructed fields are generated using Monte Carlo sampling of distribution-based summarized data.

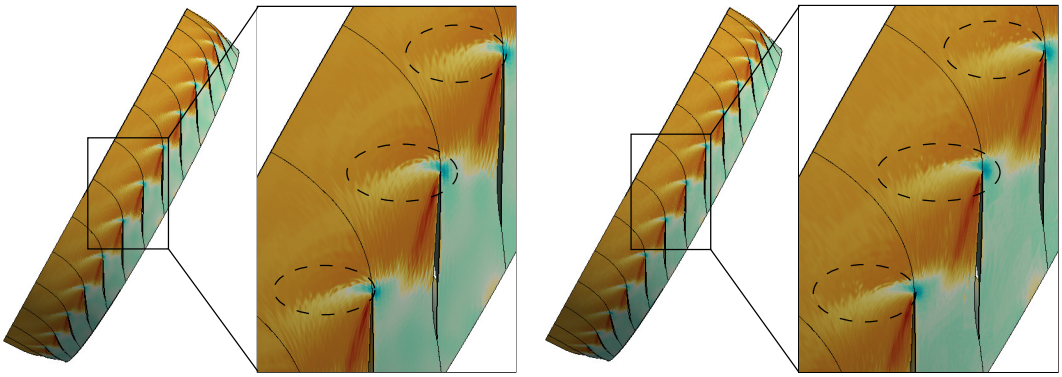
of Figure 4.7a (regular block scheme), the image produces checker-box-like artifacts (as shown by black circle in Figure 4.7a). Note that, this image is produced using a block size of $3 \times 3 \times 3$. If we increase the block size, these artifacts will become even more prominent which further reduces the visual quality. In comparison, the k-d tree based reconstructed data generates a comparatively smoother result with fewer artifacts (highlighted with black dotted lines in Figure 4.7b), however, a low entropy threshold of 1.2 was used to achieve it which led to increased storage (see column 9 of Table 4.1). Finally, Figure 4.7c depicts the

result of our proposed SLIC-based partitioning scheme (uses approximately $6 \times 6 \times 6$ points per cluster), which produces the closest visual quality to the raw data.

Visual Analysis using Turbulent Combustion Data Set. The Combustion data set is a time-varying turbulent simulation data set containing 5 chemical variables. The spatial resolution of each variable is $240 \times 360 \times 60$. The data set was made available by Dr. Jacqueline Chen at Sandia Laboratories through US Department of Energy's SciDAC Institute for Ultra-scale Visualization. We used the mixture fraction variable, which represents the proportion of oxidizer mass and fuel in the data.

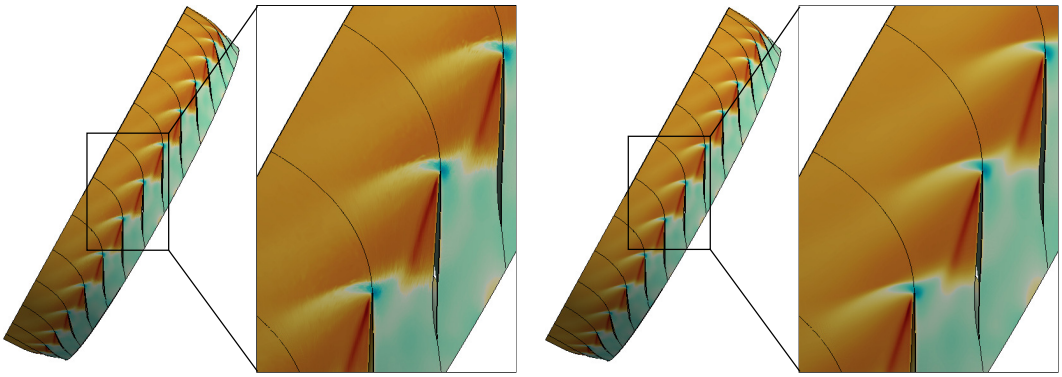
Visualizations generated by different summarization methods using the mixture fraction variable of Combustion data set are shown in Figure 4.8. From the reconstructed image using regular partitioning scheme using block size $3 \times 3 \times 3$, we see checker-box-like discontinuities in Figure 4.8a marked with red dotted lines. The k-d tree based reconstruction technique is able to reduce this checker-box-artifact. However, the result still shows some differences on the boundary of the flame structures (highlighted with red dotted lines in Figure 4.8b) when compared to the raw data in Figure 4.8d. In particular, the k-d tree algorithm produces a partition (the red dotted region in the top right in Figure 4.8b) which only covers a small portion of the flame structure. The rest of the region is considered the background, containing homogeneous values. As a result, this region was not partitioned further by the k-d tree since it satisfied the entropy-based termination criterion. During reconstruction, this causes higher error making it impossible to recover the correct boundary of the flame structure. Note that, a smaller entropy threshold will divide this region into smaller partitions, thus reducing this artifact, but consequently the storage cost will increase.

The visualization produced by the proposed SLIC-based partitioning scheme, depicted in Figure 4.8c, was produced using approximately $6 \times 6 \times 6$ points per cluster. With the



(a) Reconstruction using regular partitioning scheme. The block size is $3 \times 3 \times 3$. A zoomed view is shown on the right.

(b) Reconstruction using k-d tree based scheme. Entropy threshold of 1.2 is used. A zoomed view is shown on the right.



(c) Reconstruction using SLIC-based scheme. Approx. $6 \times 6 \times 6$ points per cluster is used. A zoomed view is shown on the right.

(d) Raw data (Ground truth). A zoomed view is shown on the right for better visual comparison.

Figure 4.9: Figures 4.9a-4.9d: Visual comparison of Pressure field of Turbine data set. The reconstructed fields are generated using Monte Carlo sampling of summarized data.

smallest storage, the SLIC-based visualization matches the raw data the best, preserving the overall flame structures on both the sides.

4.3.5 In Situ Application Study and Performance Evaluation

In this section, we present a domain study using a large-scale computational fluid dynamics (CFD) simulation code, TURBO [32, 33], and demonstrate the applicability of

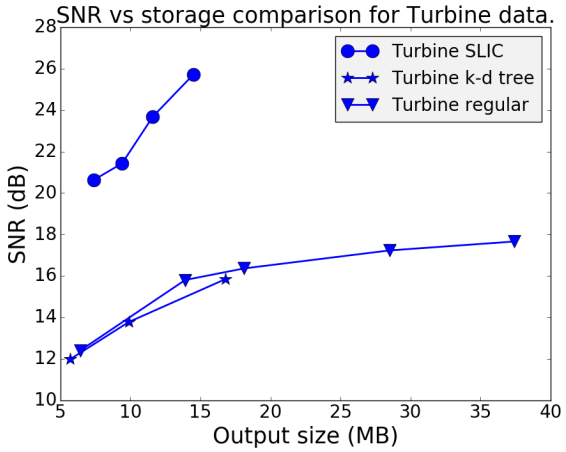


Figure 4.10: Storage vs SNR comparison of Turbine data. It is observed that, with similar storage, the SLIC-based method produces more accurate visual quality compared to other techniques.

the SLIC-based method for *in situ* environments. TURBO is a high-resolution, Navier-Stokes based, time-accurate CFD code, developed at NASA, and is used to study the flow instability in transonic jet engine compressors. Our domain expert studies the characteristics of Pressure values for detecting the inception of flow instability. Figures 4.9a-4.9d show the sampling-based reconstructed results of Pressure variable, where the proposed method produces higher quality reconstruction compared to the other methods. The image generated using regular partitioning and k-d partitioning contain artifacts highlighted with black dotted lines in Figure 4.9a and 4.9b. Furthermore, we study the storage vs SNR of the three methods and present the results in Figure 4.10. We observe that with equal storage, the SLIC-based method achieves much higher quality than the other methods. However, the size of the raw data output of a single simulation is quite large which makes the analysis cumbersome and overwhelming for the expert.

Table 4.3: *In situ* performance of the proposed SLIC-based method.

Simulation (hrs)	Simulation raw I/O (hrs)	*In situ* analysis (hrs)	*In situ* I/O (hrs)
13.217	2.06	1.822	0.015

We applied the SLIC-based method for summarizing Pressure variable *in situ* which only stored the distribution-based summary data. Our *in situ* study was done using a cluster, Oakley [28], at the Ohio Supercomputer Center, which contains 694 nodes with Intel Xeon x5650 CPUs (12 cores per node), and 48 GB of memory per node. The simulation was run with 328 cores for the study, and we ran it for 2 revolutions, resulting in 7200 time steps. *In situ* call was made at every 10^{th} time step which required us to process 1.008 TBs of data for 720 time steps. Note that, the expert used to only write out raw data at 25-30 time steps without the *in situ* capability. We summarized the data of the rotor section of the model, by directly accessing the simulation memory without any additional data copy. During *in situ* processing, we generated the partitioning using SLIC and summarized the partitions with our hybrid distribution-based scheme. The raw simulation outputs 5 variables in multi-block plot3d format, and the raw data size for the rotor section is 690 MB per time step, i.e., 496.8 GB for just 2 revolutions. The domain of the compressor consists of 36 blocks (blade passages), and the spatial resolution of each block is $151 \times 71 \times 56$. In contrast, the size of the SLIC-based summarized data for Pressure variable is only 10.8 GB, i.e., around 54 GB for all 5 variables, using SLIC-based summarization with approximately $6 \times 6 \times 6$ points per partition, resulting in a significantly smaller data. Table 4.3 shows the timings of the *in situ* run for this study. We see that the SLIC-based method takes about 13.5% of the simulation time for analyzing the data and summarizing it. In contrast,

post-hoc SLIC-based partitioning and summarization on a standard Linux machine with an Intel core i7-2600 CPU, 16 GB of RAM, and 1 TB HDD using OpenMP parallelization, takes 73.5 secs on average per time step, i.e., about 14.7 hrs for processing 720 time steps. Furthermore, the computation time for Monte Carlo sampling for all the 720 time steps took 2.96 hrs including the I/O time.

We presented the results to the domain scientist. The expert agreed that the reduced summary data accelerates the post-hoc analysis, which can be used as a replacement of the raw data for exploration. The expert was particularly impressed by the reconstruction quality that we achieved with the SLIC-based method over the other partitioning techniques. Also, the expert acknowledged that, by transforming the data into a distribution-based representation, a wide range of visual-analytics can be done using it. With the *in situ* data triage and summarization, the expert now can keep a higher temporal resolution of data by storing more time steps than before, which will help in a more precise temporal event detection. Hence, the expert feels that the additional computational time spent for *in situ* analysis is well justified, given the benefits it offers during exploratory post-hoc analysis. Finally, the domain expert also suggested extending the SLIC-based method for multivariate and vector fields which will increase the usefulness of the method.

4.3.6 Discussion about Different Partitioning Schemes

The regular partitioning requires minimum storage for the same number of partitions among these techniques because the partition bounds are implicit, so, no additional storage is necessary. However, the quality of sampling-based data reconstruction and probabilistic feature analysis using regular block-based summary data is found to be less effective, since this method does not consider data value coherency, resulting in partitions with high-value

variance. The key difference between regular partitioning, k-d tree partitioning, and the SLIC-based scheme is that the last two methods aim at reducing value variation while creating the spatial partitions. Nevertheless, for finding spatially homogeneous regions, k-d partitioning often divides the data into many smaller axis-aligned regions which cause higher storage. Note that, we can reduce the storage of k-d based scheme by changing the k-d tree decomposition termination criterion, but, that will also reduce the summarization quality.

The SLIC-based partitioning works by generating irregular partition shapes. With this, SLIC captures data homogeneity better than the other two methods. It minimizes data value variation inside each partition and enables more accurate statistical data summarization. From the three selected parameter configurations highlighted in Table 4.1, we find that SLIC-based summarization preserves the statistical data properties more accurately, reflected by the best SNR-to-storage ratio. Using larger partitions, we effectively summarize a smaller number of partitions when compared to the other two methods, which is the primary reason that we have the best SNR-to-storage ratio. We achieve superior partitioning through irregularly shaped cluster representation and compact distribution-based summarization with compressed cluster id information.

However, the introduction of irregularly shaped partitions in the SLIC-based method has resulted in a storage overhead of cluster ids per point, which can be regarded as a potential limitation. An improved method for storing cluster information will make this method even more storage efficient. Finally, we have successfully applied the method to a large-scale CFD data set to demonstrate the *in situ* capability of our method. Positive feedback from the domain expert further shows the effectiveness of the SLIC-based method for summarizing large-scale data sets for flexible post-hoc analysis.

4.4 Global Probabilistic Data Models

Global probabilistic data models transform the data into a single global distribution which represents the overall statistical properties of the data (see Figure 4.1b). In this case, for each data variable, a separate probability distrbution is estimated. Use of global data models has found many applications in visualization such as statistical volume visualization, transfer function design of time-varying data sets, statistical analysis of isosurfaces [4, 79, 113]. Furthermore, the global data distributions can be efficiently used for computing various information theoretic measures, which have a significant impact on data exploration and visualization. Even in the presence of the full-resolution raw data, information theoretic measures have been shown effective for correlation-based multivariate analysis, automatic informative view selection for rendering, important variable identification, salient isosurface detection, uncertainty quantification in visualization etc. [15, 17, 19, 24, 152]. For creating a global distribution-based data model, storage of the distribution is not critical since the size of a single global distribution is very small. However, as histograms are fast to compute compared to the other distribution models, it has been used widely for modeling global data distributions.

4.5 Conclusion

In this chapter, we have presented various types of distribution-based data summarization schemes. Three different local region-based data partitioning techniques are proposed for summarizing data using distributions. A comprehensive study (both quantitative and qualitative) among them reveals that the SLIC-based irregular shaped partitioning based summarization technique is able to preserve the highest quality of data using distribution-based summarizations. Since the partitions are created based on data homogeneity, distribution

in each partition contained lower data value variation which resulted in a higher quality of analysis. Furthermore, we also conducted *in situ* estimation of such distribution-based data summarization using a real-world large-scale flow simulation to show the computation performance of the summarization scheme and measure the amount of data reduction. However, while analyzing a time-varying phenomenon where comparison of data properties in a fixed local region is necessary over time, this irregular data-driven partitioning may not be readily applicable since the partition shapes over time will change. This will require additional data processing for finding a best possible correspondence among partitions over consecutive time steps. In such cases, a regular or k-D partitioning based data summarization scheme may still be used where the partitions do not change shapes and boundaries over time. Therefore, the choice of partitioning schemes should be done based on the application requirements. Next, in the following chapters, we show the usefulness and applicability of the various proposed data summarization schemes using several scientific applications.

Chapter 5: Distribution Data Guided Flow Instability Analysis for Rotating Stall Detection and Visualization

Recent advancements in parallel computing capabilities have enabled aerodynamics scientists to study the phenomenon of rotating stall in great detail by performing high-resolution numerical simulations. Rotating stall initiates from local airflow disturbances among the engine compressor blades, but grows rapidly to become fatal to the engine. Numerous efforts have been made in the past to understand this phenomenon in detail. Recently, a computational fluid dynamics (CFD) simulator TURBO [32, 33] has been developed in NASA which is capable of accurately modeling the stall behavior. However, due to the computational cost and the amount of data produced, traditional post-processing analysis utilizing raw data is not ideal, since storing all the data is not a viable option. This is because of the bottleneck comes from I/O compared to the ever-increasing computing speed.

The goal of this work is to analyze and visualize the spatiotemporal evolution of rotating stall in a jet engine simulation. More specifically, the expert wants to identify **(a)** the blade passages; and **(b)** time step ranges when stall is imminent. Detection of early signs of the stall is critical for the analysis as it enables the expert to perform exploration with a focus on the relevant data. In addition, the expert wants to concentrate on the transition of the simulation from a stable condition to an unsteady state which leads to engine stall. An

important point to note in this context is that, for some operational conditions, it is unclear whether the simulation is going to stall condition. Therefore, it is often necessary to run the simulation for long times in supercomputers which produces a very large data set. Thus, the expert wants to have a significant storage reduction in their data so that scalable post-hoc analysis is achievable. The expert also hopes to take a multifaceted approach requiring several variables for stall analysis. Finally, visualization techniques are required which can be used to validate the hypotheses and formulate new reasoning for a better understanding of rotating stall.

To address the aforementioned issues, we use our local region-based distribution data summarizations to allow the expert for detailed, flexible, and scalable stall analysis. As discussed in previously Chapter 4, we triage the extreme-scale simulation data *in situ* and reduce the data size significantly by storing the GMM based block-wise distribution data. Using only the distribution data, we demonstrate a scalable method for rotating stall analysis which exploits the advantages of *in situ* analysis and facilitates exploratory post-hoc analysis. Since rotating stall is in general characterized by local disturbances in airflow, it is non-trivial to have a precise descriptor for their detection. Hence, use of the proposed distribution guided methods in such scenario gives us promising results. We use probability distributions in the form of GMMs to model local statistical properties of the data and analyze the spatiotemporal distribution variations to identify stall impacted regions. The post-hoc analysis using GMM based data is done through leveraging the existing and new visualization algorithms with the much-needed capability of uncertainty quantification [23, 89, 94, 112–114]. We employ statistical anomaly analysis over space and time for identifying the stalled regions. To validate the suspected locations in the spatial domain, we utilize uncertain isocontour algorithms. Positive feedback from the

expert confirms the efficacy and benefits of the proposed method, and also demonstrates the capability of *in situ* processing in analyzing extreme-scale data sets in an effective way. Therefore, Our contributions in this work are as follows:

1. We present a novel distribution guided rotating stall exploration technique which exploits both spatial and temporal nature of stall by analyzing the statistical information of data stored during *in situ* processing.

2. We use a comparative visualization technique to conduct anomaly pattern analysis using multi-variables to obtain new insights about rotating stall.

3. Finally, we make use of uncertain isocontouring and other spatial visualization techniques to allow the expert to explore the stall analysis results in the spatial domain for validating the hypotheses.

5.1 Motivation, Requirements, and Overview

The scientists have always sought after techniques that can detect the sign of rotating stall as early as possible so that corrective control methods can be applied. Rotating stall initiates as intermittent local flow separation on the turbine blade surface, often caused primarily by fluid instabilities around the tip region of a blade. These regions initially contain small bubble-like blockages which hinder normal airflow through the passages. If the blockages increase over time and become persistent, they are characterized as stall cells. By detecting and studying the local flow abnormalities in the early stages, further understanding of rotating stall formation can be gleaned. To study the rotating stall phenomenon in detail and understand how it develops over time, in this work, we have used a high-resolution CFD simulator TURBO [33]. It has been validated by previous works [33, 60] that TURBO is

able to model the rotating stall with high resolution and hence provide detailed knowledge about the phenomenon of rotating stall.

5.1.1 Limitations of Current Stall Analysis Approaches and Motivation

The motivation of our work stems from the limitations of existing stall analysis approaches which are twofold in nature: **(1)** The limitations that arise from excessive storage requirements, I/O bottleneck, and prolonged post-processing time; **(2)** The limitations of existing conventional stall analysis techniques. Next, we discuss each of these points briefly and justify the necessity of our work.

A full annulus simulation of TURBO consisting of 4 revolutions of the compressor stage takes around a day to finish and generates around 20 TBs of raw data. Processing, handling, and analyzing such scale of data creates a significant obstacle to the domain scientist. Therefore, a growing need for scalable and efficient analysis methods has become prominent. The bottleneck coming from I/O, and the cost of moving a huge amount of data from supercomputers to the local processing machines become prohibitive as the data size grows. Furthermore, traditional post-hoc stall analysis methods require a longer time for such large data to produce any useful results.

Among the most utilized existing stall detection techniques are mass flow rate plots and pressure probe observations. **Mass flow rate** is a measure of the air mass that flows through the compressor stage per unit time [60] and is defined as $\dot{m} = \rho \vec{v} \cdot \vec{A}$, where ρ is the density, $\vec{A}$ is the area vector of a flow path cross-section of the inlet or exit of the compressor, and $\vec{v}$ is the flow velocity. Mass flow rate in stable condition remains constant over time, however, when rotating stall happens, it drops rapidly. Unfortunately, the event is only observable as rotating stall is occurring and thereby cannot be used as a precursor for the stall.

Analysis using **pressure probe** readings [41, 99] to capture pressure variations at fixed locations over time has a better potential of detecting earlier signs of the stall. This technique employs pressure probes at the engine casing circumferentially. When a rotating instability passes through the probed locations, the pressure reading of the probes fluctuate and by observing the time-varying patterns of such pressure probes, rotating stall can be detected. However, it is non-trivial to find appropriate probing locations and usually, the pressure probe-based methods only use a few probes to detect the stall phenomenon which does not provide detailed spatial information.

The aforementioned problems of current stall analysis approaches have led us to design a new approach with a list of domain-specific requirements. Below we list those requirements first and then present an overview of our approach for solving them.

5.1.2 Domain Specific Requirements

Following are the requirements that have been identified:

1. Since the expert wants to have a significant reduction in data size and yet preserve the important information which can be analyzed flexibly, an *in situ* analysis seems suitable.

2. Since the stall phenomenon is characterized by locally unstable regions, a spatial region-based analysis will be more suitable. This will reduce the processing cost and yet detect the desired regions. Furthermore, apart from the spatial anomaly, the expert is also interested in finding anomalous airflow behavior in temporal domain and hopes to verify both analysis methods in capturing the early signs of the stall.

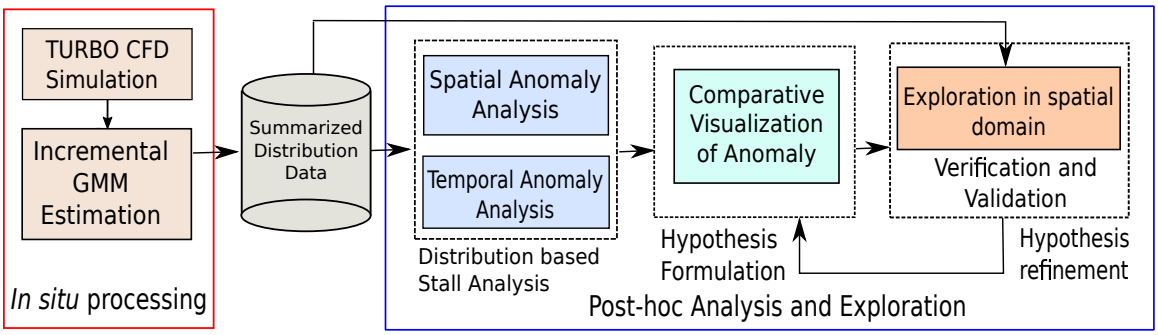


Figure 5.1: A schematic diagram of the proposed analysis method.

3. A majority of the previous works have focused on observing the variation of pressure for stall analysis. The expert desires to look at other variables as well. It is hypothesized that the entropy values are also closely related to the stall cells, so the domain expert wants to compare the effectiveness of pressure and entropy as stall indicator variables.

4. The reduced data output should be capable of rendering the results in the spatial domain for exploration, verification, and validation.

5.1.3 Overview of Our Approach

To fulfill the above requirements set by the expert, we provide a new rotating stall analysis approach which integrates *in situ* data summarization with distribution-based analysis and visualization techniques. Figure 5.1 shows a schematic diagram of our proposed method. To tame the extreme size of the simulation output, during the simulation we take advantage of *in situ* processing to summarize the important data into GMM distributions. The details of the *in situ* GMM estimation has been described previously in Chapter 4. In this chapter, we demonstrate, how we exploit the GMM based data summaries for identifying the stall

suspected regions efficiently. We employ spatial and temporal anomaly detection methods on the stored GMM data, which measure statistical variations among GMMs over space and time. By studying the spatial and temporal anomaly results of multi-variables through comparative visualization, the expert can analyze the evolution of rotating stall and identify the locations where the stall is initiated. Finally, to investigate the detected phenomena in the spatial domain for validation and verification of the hypotheses, we allow the scientist to visualize the spatial distribution data with uncertain isosurfaces.

5.2 Rotating Stall Analysis Using GMM based Distribution Data

In this section, we present the stall analysis techniques which exploit the statistical information summarized in the form of GMMs. Since the domain expert describes *stall cells* as local instability in the airflow, we have focused on identifying such instability by characterizing them as a local statistical anomaly in the data. In a stable condition, all the blade passages of the compressor are expected to be *axisymmetric*. Hence, a specific local region (i.e. a block) relative to all the passages would behave symmetrically. Furthermore, the observed values of simulation variables such as pressure, and entropy at a local region in a specific passage would be similar in future time steps as well. Based on these two key observations, a classification for anomalous regions over both space and time can be achieved; where quantification of a region being anomalous is defined by the amount of statistical dissimilarity in the same region of other passages over space and time. In the following, we first describe our spatial anomaly based analysis and then introduce a temporal anomaly measure for rotating stall analysis.

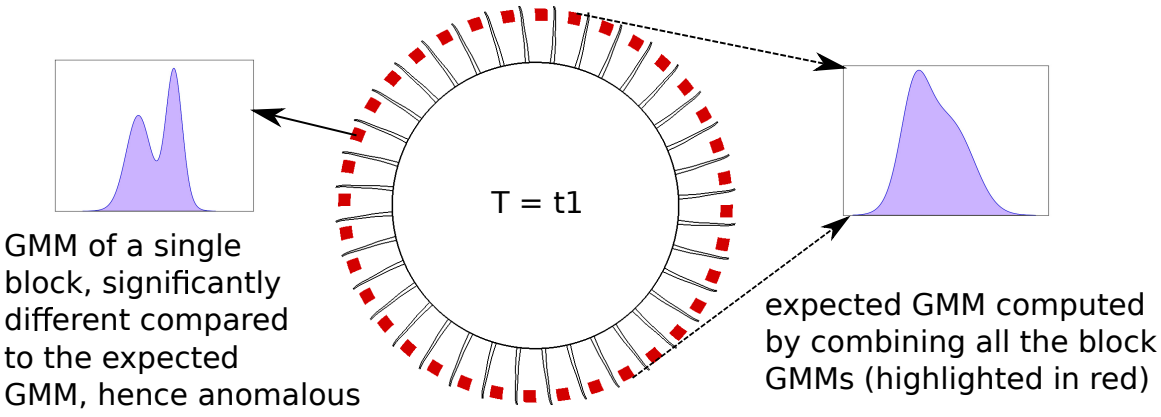


Figure 5.2: Illustration of spatial anomaly detection method using GMM distributions over space.

5.2.1 Spatial Anomaly Guided Stall Analysis

In our *in situ* distribution guided analysis, we aim at detecting local anomalous regions i.e. the data blocks which are expected to contain stall cells. In a recent work, a point-based spatial anomaly detection method was proposed by Chen et al. [30] and was shown to have a high potential for identifying instability which indicated the early formation of stall cells. The method used the pressure variable for the analysis and exploited the property of axisymmetry among blade passages.

In our work, we advocate for a local region based analysis for the detection of stall cells rather than a point based analysis because the point-based method requires a whole domain analysis on the raw data, which is too expensive to perform post-hoc or compute *in situ*. Furthermore, the expert believes that, since the stall cells form a spatial region, a local region based analysis is more suitable. Following these thoughts, in this work, block-wise local distributions are modeled and stored in the form of GMMs for efficient post-hoc analysis. To detect the spatial anomaly, we first group the GMMs coming from the same relative location

in each passage. As illustrated in Figure 5.2, each group contains 36 blocks (highlighted in red), i.e. 36 GMMs, coming from 36 blade passages in the rotor. Thus, our goal of identifying the spatial anomaly among axisymmetric regions is to find the outliers among a group of GMM distributions.

To identify a GMM that consists of abnormal values among a group of GMMs, our approach first estimates the *expected* distribution as a basis for all the GMMs in the group to compare with. If any of the 36 GMMs is sufficiently different from the *expected* distribution, it is regarded as an outlier and the corresponding block in the physical space is reported anomalous. We define the *expected* distribution as the average of the probability density functions of the GMMs in the group. This is equivalent to computing the combined distribution from all samples in the group of GMMs. In this way, the major value distribution is attained and the effect of outlier values to the *expected* distribution, if present, is reduced. The *expected* GMM is a new GMM distribution consisting of the Gaussians from all the input GMMs with normalized weights.

After the *expected* GMM is formed for a group, we compare it with each GMM in the group using the Earth Mover's Distance (EMD). The EMD is a distance measure defined by the minimal ground transport effort to match two distribution shapes. EMD has been widely used in pattern matching and image analysis [77, 124], as well as to compare probability distributions in uncertain data [121]. Besides its robustness against noise [88], EMD's measuring of ground transportation is able to capture an outlier if its value deviates from the majority of the distribution, which is particularly desired in our anomaly detection study. To compute the EMD for 1D distributions, we use the *match distance* as the ground distance [120] since the EMD can thus be efficiently computed by the absolute difference

between the cumulative distribution functions (CDF) of the distributions [154]:

$$EMD(X,Y) = \int_{-\infty}^{\infty} |F_X(x) - F_Y(x)|\, dx \tag{5.1}$$

Here $F_X(x)$ is the CDF of the distribution X at x. The CDF of a GMM can be simply computed by the weighted sum of the CDFs of the individual Gaussians. We numerically approximate the integral of CDF difference by the trapezoidal rule. After the EMD is computed, a user-specified fixed threshold is used to extract the blocks with outlier GMMs. We apply the above method for all groups of blocks per time step and mark the blocks identified as a spatial anomaly.

5.2.2 Temporal Anomaly Guided Stall Analysis

GMM based spatial anomaly measure presented above can have a potential limitation. During a stall developing phase the anomalous regions might propagate to the majority of the blade passages and in that case, the outlier values will dominate the *expected* GMM and the spatial anomaly test may not be able to detect them as anomaly anymore. Therefore, in this section, we extend the anomaly based analysis to the temporal domain to remedy such a situation. Since the GMMs for each local region (i.e. the data block) gets updated as data from new time step is observed, it is hypothesized that in a stable state, the GMMs for a block will not change significantly over time. Therefore, by observing the temporal dissimilarity between the GMMs for the same block over time, the temporal anomaly can be computed for each block. Since temporal anomaly is computed by looking at each block individually over time, even if the majority of the blocks get affected by the stall, temporal anomaly method will still detect them as anomalous.

Another motivation to investigate the rotating stall using temporal anomaly comes from a domain-specific hypothesis. The expert thinks that when the stall has fully developed, the

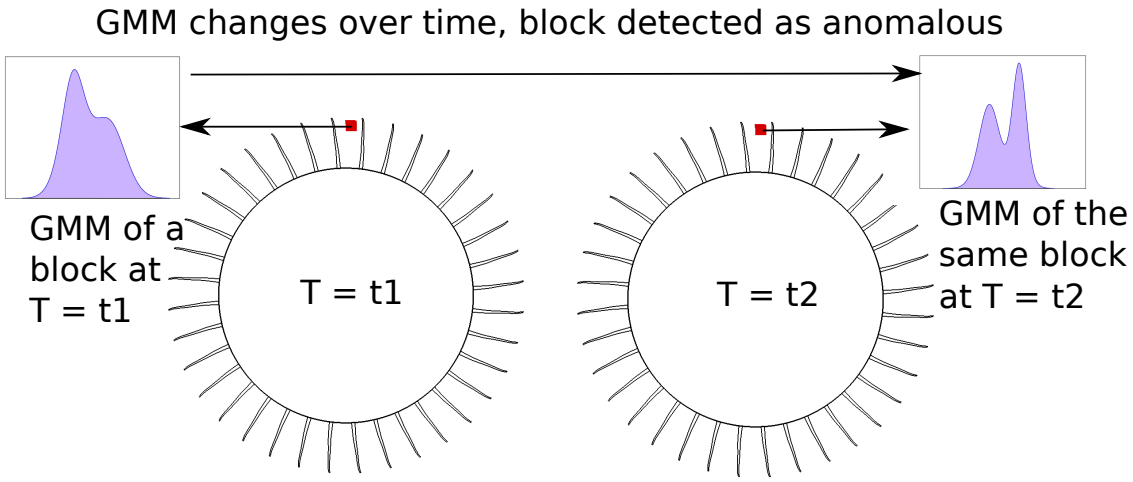


Figure 5.3: Illustration of temporal anomaly detection method using GMM distributions over time.

temporal variation of data values inside the fully grown stall cells may become less apparent compared to that of the stall developing phase. Since temporal anomaly will measure the instability of data values inside a block by comparing the block GMMs over time, it is expected that the degree of the temporal anomaly for a block contained in a developed stall cell will be small. Such a behavior of stall cells can be investigated by observing the pattern of the detected temporal anomaly in an appropriate time window.

In Figure 5.3, we present the idea of temporal anomaly using an illustrative diagram. On the left side at time t1, we show that a specific block (highlighted in red) is selected for analysis and its GMM is estimated. Next, the GMM of the same block coming from the same blade passage in the next time step is observed. If the similarity between the two GMMs over time for this block is less than a preferred degree, then the block is classified as anomalous. As shown in Figure 5.3, the selected block at time t2 has a GMM which is quite different from its previous state and therefore the block is detected to be anomalous at time t2. To measure the similarity between the GMMs of a block over consecutive time

steps, we have used the Earth Mover's Distance (EMD) which is introduced in Section 5.2.1. Therefore, applying this similarity based measurement to all the blocks over time, we estimate the chance of a block containing a temporal anomaly.

5.3 Visualization Techniques for Exploration and Verification of Detected Regions

In this section, we present the visualization techniques employed for analyzing the identified regions and hypotheses verification. Based on the requirements of the expert, we present a comparative chart as shown in Figure 5.4 that effectively shows spatiotemporal evolution of anomalies detected by different methods in an overview. Interesting time steps and regions are then selected and visualized in the physical space for verification of rotating stall and further exploration. It is to be noted that, in post-hoc analysis we do not have access to the raw data anymore, instead local distributions in the form of GMMs are available. Therefore, spatial visualization techniques that analyze probability distributions are employed in our work. Below we first describe the comparative anomaly chart as an overview visualization, followed by the spatial visualization techniques used for validation and exploration.

5.3.1 Comparative Visualization for Anomaly Pattern Study

As the anomaly based stall analysis methods described in Section 5.2 estimate the chance of instability for all regions over time, it is important to provide such information to the expert through an overview showing the pattern and evolution of the detected anomalous regions over time. By investigating the trends of detected regions from the global view, the expert can quickly identify the blade passages and time step ranges for further examination

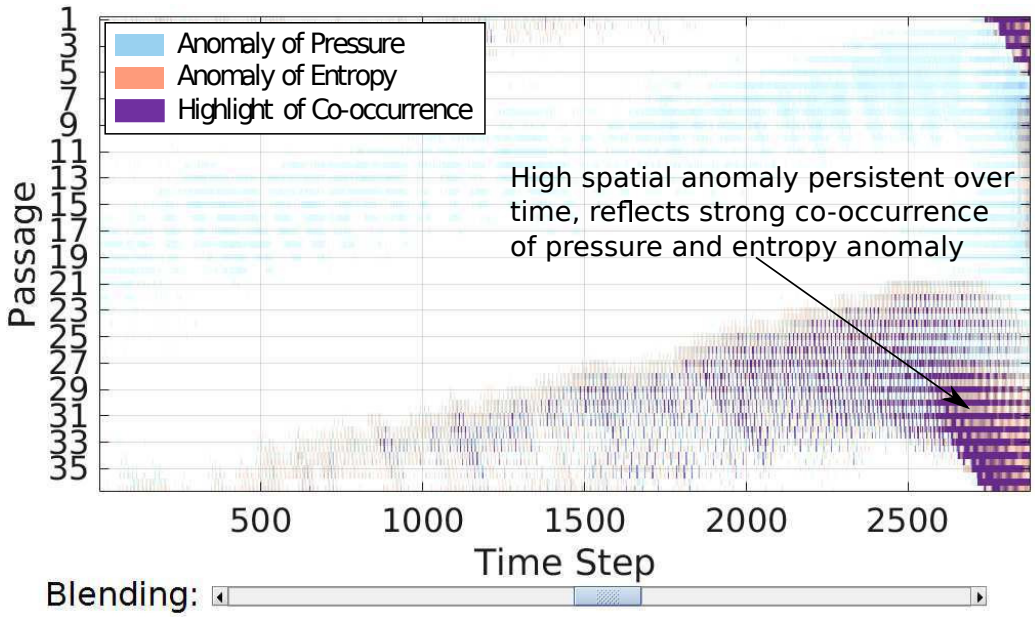
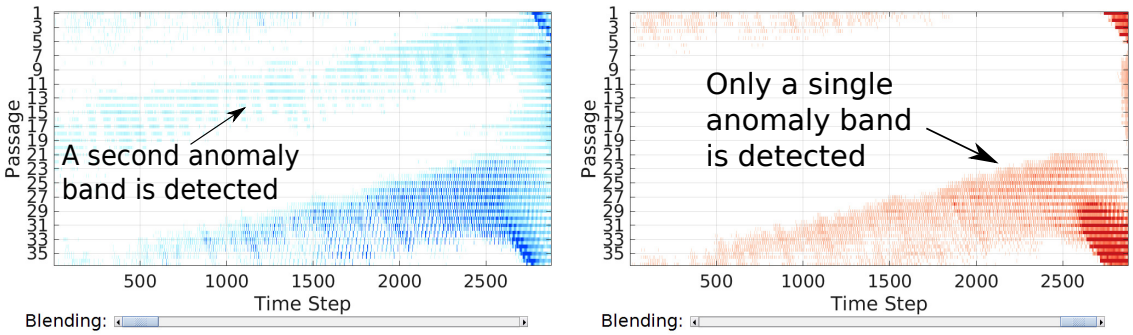


Figure 5.4: Showing spatial anomaly of pressure and entropy where the co-occurrence regions are highlighted in blended purple color.

in data domain. Moreover, a comparative visualization technique is applied for the expert to compare with anomalies detected in different variables for hypothesis verification.

The anomaly chart. According to the domain expert, stall cells that cause engine stall generally have the following properties: **(1)** They can exist and propagate across passages for a long time, and **(2)** They can grow in size to hinder normal airflow through the compressor. In the overview visualization, the expert is interested in how the detected flow instability propagates among passages instead of how it moves inside a passage. Therefore, a 2D heat map is used to visualize the anomaly detection results, where the Y axis on the chart represents the passage number and the X axis represents the time step, as shown in Figure 5.4. Each point on the chart is color coded by the size of the detected anomalous region in



(a) Spatial anomaly pattern of Pressure.

(b) Spatial anomaly pattern of Entropy.

Figure 5.5: Spatial anomaly study of Pressure and Entropy for simulation run with CMF = 14.20 kg/s.

the corresponding passage and time. As a result, points with higher counts and connected with other points across several time steps are more salient.

Superimposition. In addition to visualizing the anomaly pattern of a single variable, the expert is interested in how the anomalies detected from different variables are correlated and whether the combination of them generates a more confident indication of stall. Therefore, comparing and contrasting anomalies from different variables are required. We provide superimposed views [69] which composite two anomaly charts into one chart with transparency. Compared to juxtaposition views which place visualizations side-by-side, superimposition does not split the user's attention into different parts of the screen, and also makes the comparison intuitive [69]. Since superimposition may cause visual clutter in complex co-occurrence regions, we provide an interaction tool to overcome this problem as discussed below.

Alpha blending and interaction. To superimpose two selected charts, we overlay them with alpha blending, where the blending coefficient α is adjusted by the user. As shown in Figure 5.4, α is adjusted using a slider in the bottom, where moving the slider to an end

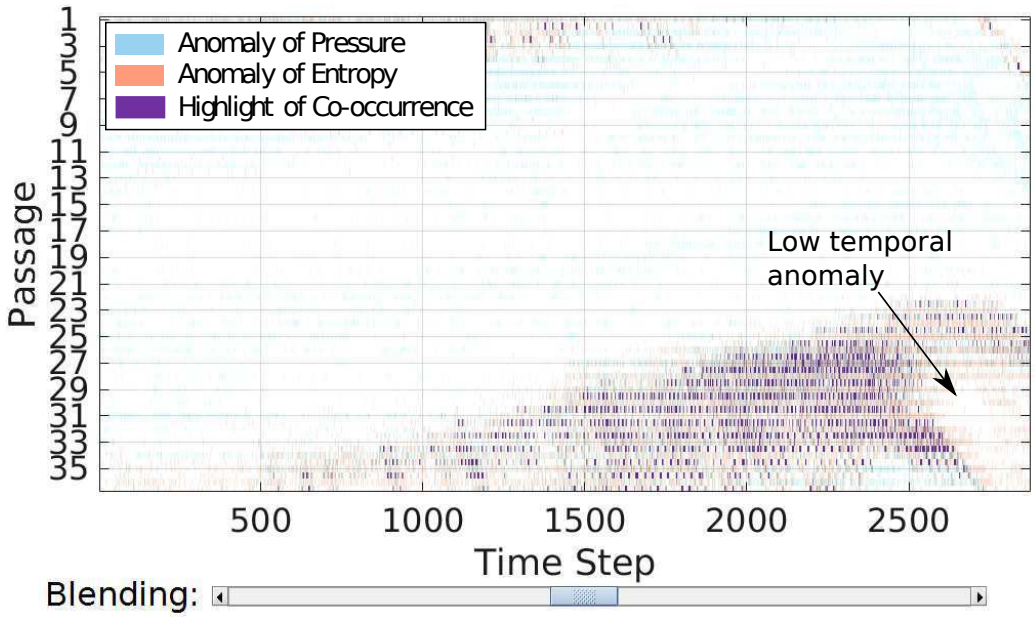
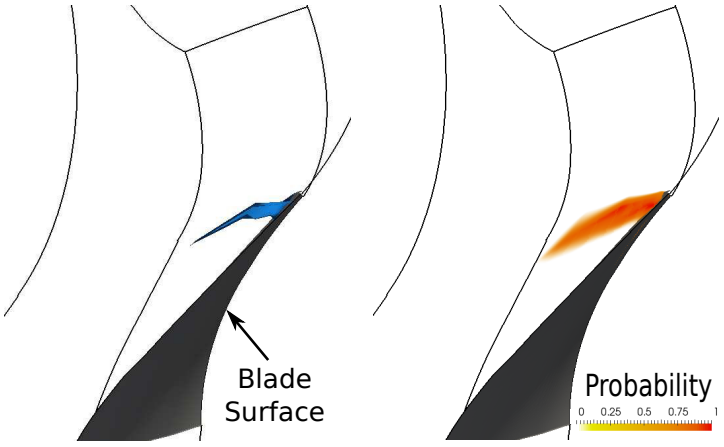


Figure 5.6: Showing temporal anomaly of pressure and entropy where the co-occurrence regions are highlighted in blended purple color.

shows the anomalies of a single variable. Therefore, the user can move the slider to contrast different anomaly detection results in position with both contents. Since it is hypothesized that the co-occurrence of anomaly from different detection criteria can indicate the existence of stall cells with more confidence, it is required to highlight these regions on the chart. Therefore, we increase the saturation of co-occurrence regions on the chart when the slider is closer to the middle, as shown in Figure 5.4. This enhances the focus on co-occurrence regions and keeps the regions of individual occurrence in context.

Spatial rendering. Since the spatial and temporal anomaly analysis technique quantifies the possibility of containing an anomalous region for the whole data domain, a new scalar field using the anomaly values is constructed. For detailed exploration on the detected anomalous regions in the spatial domain, we allow the expert to render the anomaly field



(a) Mean isocontour of pressure = 0.42.

(b) Uncertain isocontour of pressure = 0.42.

Figure 5.7: Visualization of GMM distribution-based data using surface renderings and uncertain isocontours.

using isosurfaces. The user can either inspect the detected anomalous regions per time step or animate through time to observe the growth of the anomalies. By selecting highly anomalous regions on the anomaly chart, the expert can investigate the location of potential stall impacted regions in data domain and then verify its correlation to the stall inception.

5.3.2 Hypotheses Verification using Uncertain Isocontour Visualization

In order to help the expert to verify the detected anomalous regions as potential stall cells and further understand the data, it is necessary to provide a flexible exploration tool for spatial data visualization which can utilize the distribution type data. It has been previously shown that with the data represented in the form of spatial local distributions, probabilistic visualization algorithms can be employed for analyzing and visualizing the data with uncertainty quantification [94, 112–114]. As isocontouring is commonly used by the

domain expert to visualize the data variables and verify the stall phenomenon with domain knowledge, we make use of an existing uncertain isocontouring algorithm proposed by Pöthkow et al. [110, 112] to generate the level crossing probability field. For each cubic cell with probability distributions modeled at the eight vertices $(X_1, X_2, \cdots X_8)$, the level-crossing probability of isovalue ϑ is defined as:

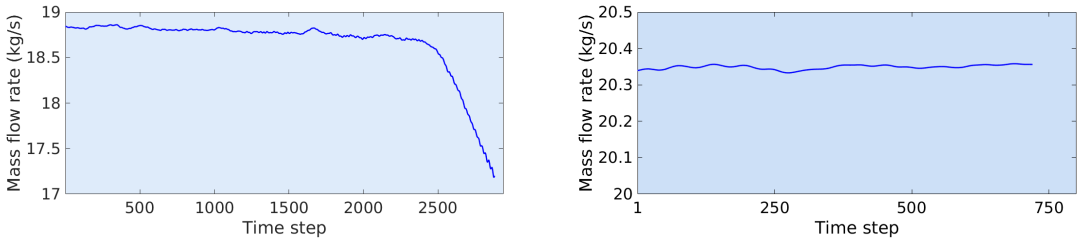
$$\begin{aligned} Pr(\vartheta\text{-crossing}) &= 1 - Pr(\vartheta\text{-non-crossing}) \\ &= 1 - Pr(X_1 > \vartheta, X_2 > \vartheta, \cdots X_8 > \vartheta) \\ &\quad - Pr(X_1 < \vartheta, X_2 < \vartheta, \cdots X_8 < \vartheta) \end{aligned} \tag{5.2}$$

We use the stored GMMs to represent the probability distributions on the vertices of each cell and assign them to X_i. The computation result is a probability field of level crossing, which is then visualized by volume rendering. Figure 5.7a shows a distribution mean isosurface of a low pressure value (pressure = 0.42) and in Figure 5.7b the uncertain isocontour of same pressure value is displayed. It can be observed that the uncertain isosurface is able to provide a better estimation with uncertainty information presented in the form of level crossing probability. In this example we have extracted a single passage and applied the aforementioned algorithm for computing the level crossing probability given isovalue of pressure = 0.42.

5.4 Case Studies and Expert Feedback

Here we present the case studies for demonstrating the efficacy of our distribution guided stall analysis method and discuss the domain expert's feedback. we used two simulation runs in this work to verify the effectiveness of the method. The simulation parameter *corrected mass flow rate* (CMF) was varied to produce two distinct cases where one led to a stall and

the other without a stall. The simulation with CMF = 14.2 kg/s produced stall and the run with CMF = 16.0 kg/s was stable. We verified our method on both the cases and found that the proposed method worked accurately in each of the test cases in detecting stall or the absence of it. Using 4 Gaussians per GMM for estimating the distributions of a data block gave us stable results as was observed in earlier work [89]. Furthermore, since we aimed at capturing local data properties, a smaller block-size was preferable in our case to achieve better accuracy. However, a smaller block-size could lead to higher storage and hence there should be a trade-off between allowed storage and the block-size. We used a block-size of $5 \times 5 \times 5$ throughout all the experiments which obtained stable results with good data reduction.



(a) Simulation with CMF = 14.2 kg/s

(b) Simulation with CMF = 16.0 kg/s

Figure 5.8: The mass flow rate plot of simulations in a stall condition (CMF = 14.2 kg/s) and a stable condition (CMF = 16.0 kg/s).

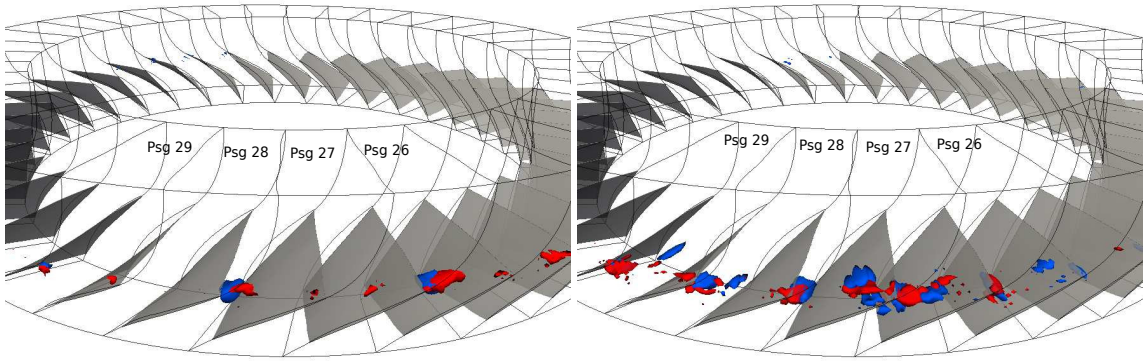
5.4.1 Simulation Run with Stall (CMF = 14.2 kg/s)

Exploration using spatiotemporal anomaly chart. Figure 5.8a shows the mass flow rate plot of this simulation. We ran the simulation for 8 revolutions to obtain a fully developed stall condition. Figure 5.4 and 5.6 show the superimposed chart of spatial and temporal anomaly respectively where each of the charts demonstrates the evolution of

anomalous regions by combining both pressure and entropy. From Figure 5.4 it is observed that around time step 2540, the anomalous regions show strong co-occurrence, and become persistent. This pattern is visible from the consistent purple color. By inspecting the mass flow rate in Figure 5.8a, we observe that the mass flow rate drops rapidly around the same time step 2540. This sudden drop in mass flow rate confirms the occurrence of the stall and verifies that our combined spatial anomaly chart is able to capture this phenomenon. Another important observation from Figures 5.4 and 5.6 is that both of these combined anomaly charts show the existence of anomalous regions starting around time step 500 which is much earlier than the final occurrence of the stall at time step 2540 detected by traditional stall indicator mass flow rate. Furthermore, since the mass flow rate, as shown in Figure 5.8a, presents an almost flat pattern up to time step 2540, which does not indicate any imminent stall, the expert agreed that the proposed method is able to detect the signs of stall much earlier than the traditional technique using mass flow rate.

By studying the spatial anomaly chart of pressure (Figure 5.5a), the expert found that there are two anomaly bands. However, entropy anomaly chart showed only one band (Figure 5.5b) which is located at the bottom of the chart including passages from 24 - 32. Since only the passages in this second band eventually led to a stall, it was concluded that entropy identified the stall impacted regions more accurately than pressure. The expert noted that entropy was not a very commonly used variable in stall detection and with this new finding a more accurate and refined stall indicator measure could be devised using entropy along with pressure.

A further inspection of Figure 5.4 showed that when the persistent purple regions appear from time step around 2540 reflecting a strong co-occurrence of spatial anomaly of both both pressure and entropy, as annotated in Figure 5.4, the temporal anomaly becomes low

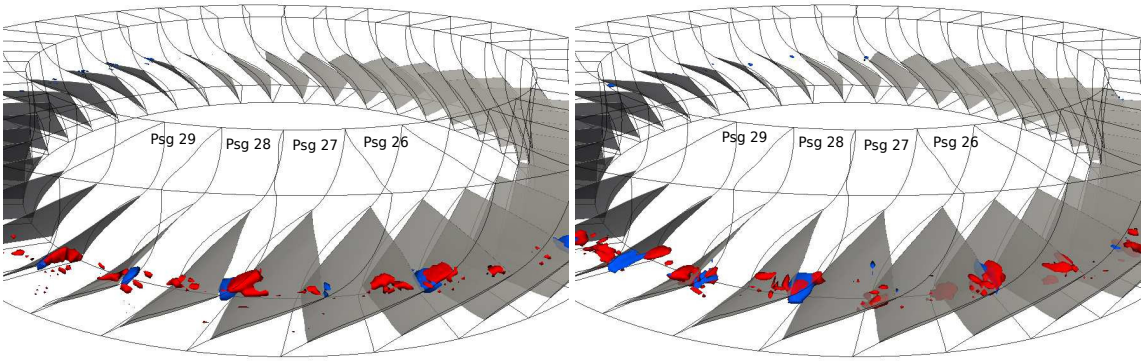


(a) Spatial anomalies at time step 2200. (b) Temporal anomalies at time step 2200.

Figure 5.9: Visualization of detected anomalous regions with the stall condition (CMF=14.2) at time step 2200. Spatial and temporal anomalous regions of pressure (in blue surfaces) and entropy (in red surfaces) are detected near the blade tip regions of several rotor passages.

in such stalled regions as marked in Figure 5.6. This pattern of temporal anomaly chart helped the expert to confirm the hypothesis that when the rotating stall is fully developed, the variation of values (pressure and entropy in our study) inside the stall affected regions become less since value distributions of variables do not change significantly with time in fully developed stall cells. Therefore, the temporal anomaly is low between the GMMs over time. Furthermore, since such developed stall impacted regions only cover a subset of passages, spatial anomaly becomes high due to the increased asymmetry among blade passages. However, since the expert is primarily interested in detecting signs of stall inception at earlier time steps, it was concluded that both spatial and temporal anomaly methods are capable of capturing the early signs of rotating stall.

Visualization of detected anomalous locations in spatial domain. To study the anomalous regions detected within the blade passage range 24 - 32, we render the surfaces which contain the detected anomalies of both pressure and entropy. Figure 5.9 and 5.10 depicts the spatial and temporal anomalous regions of pressure (blue) and entropy

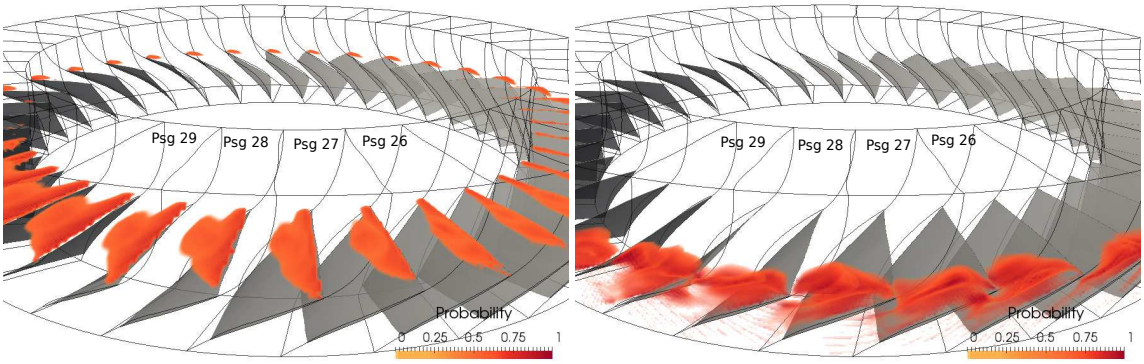


(a) Spatial anomalies at time step 2540. (b) Temporal anomalies at time step 2540.

Figure 5.10: Visualization of detected anomalous regions with the stall condition (CMF=14.2) at time step 2540. Spatial and temporal anomalous regions of pressure (in blue surfaces) and entropy (in red surfaces) are detected near the blade tip regions of several rotor passages. These regions act as blockage to the regular airflow and create flow instability which eventually leads to stall.

(red) detected by the proposed method respectively. Figure 5.10a and 5.10b present the anomalous regions at time step 2540 when the sharp drop of mass flow rate is initiated. To investigate the anomalous regions at an earlier time step, results of spatial and temporal anomalous regions from an earlier time step 2200 is shown in 5.9a and 5.9b. Two important observations about the detected regions emerged from Figures 5.9 and 5.10: **(1)** the anomalous regions of pressure and entropy demonstrate high spatial co-occurrence within the blade passage range 24 - 32 which is also visible from the anomaly chart in Figure 5.4 and 5.6, and **(2)** the detected anomalous regions appear near the blade tips.

Verification and Expert Feedback. According to the expert, the stall cells are generally located around the blade tip regions. The detected anomalous regions also appear on the tips as seen in Figures 5.9a, 5.9b, 5.10a, and 5.10b respectively. The expert further explains that in a stable state, the tip regions contain a vortex, known as *tip clearance vortex*, and pressure in the vortex core is low. Due to the axisymmetry property, all the tips are expected



(a) Uncertain isocontour of pressure = 0.38. (b) Uncertain isocontour of entropy = 1.0.

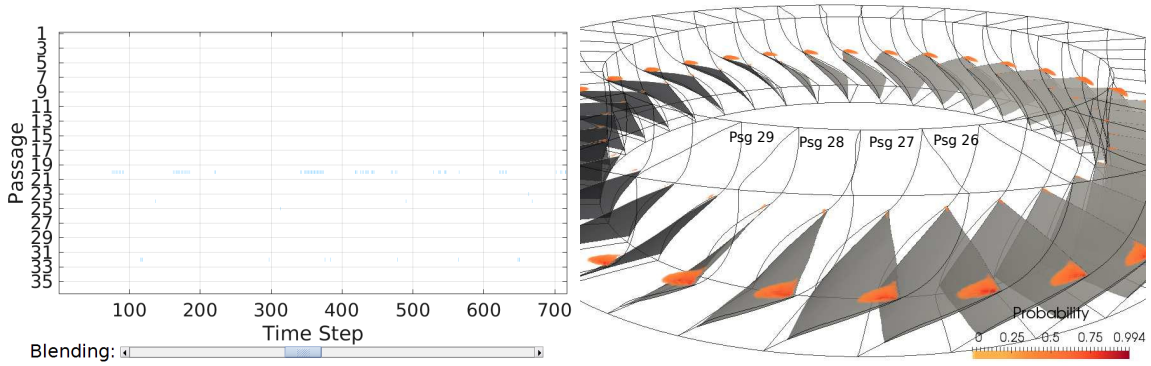
Figure 5.11: Uncertain isocontour visualization at time step 2540 for visual exploration and verification of stall impacted regions.

to have similar low-pressure region indicating the tip vortex. However, as the compressor approaches stall, the passages affected by the formation of stall cells tend to show more fluctuations in pressure values around the tip region. Finally, when the stall occurs the axisymmetry observed in the tip vortices is broken and the region is classified as anomalous. Entropy values in the stall impacted regions also increase significantly compared to the blade passages which do not contain the stall cells. To visualize these phenomena, the expert used uncertain isocontours of low pressure (pressure=0.38) and high entropy (entropy=1.0) as shown in Figure 5.11. As can be seen from Figure 5.11a, the uncertain isocontour of pressure at time step 2540 is distinctly different in the stall affected passages, while on the other side of the rotor, the contours are well organized and symmetric. Also, in Figure 5.11b it is observed that high entropy contour is located in the similar passages close to tip regions which further confirms the locations of the stall. These observations conform well with the blade passages detected using our anomaly based analysis and validate the efficacy of the proposed method.

To further confirm whether the anomalous regions are indeed stall cells, the expert studied the formation and evolution of anomalous regions for all the time steps. It was observed that the anomalous regions actually propagate from passage to passage in the opposite direction to the rotation of blades. During the formation of these regions, they pop up and gradually move to the neighboring passages. This behavior is consistent to the transportation of stall cells and a domain explanation is as follows: when a stall cell grows in a passage, it forms a blockage to the incoming flow which redirects a portion of the flow to the neighboring blades. This increases the angle of attack and causes stall for the proceeding blade, as well as, decrease the angle of attack on the preceding blade increasing stability. However, as the proceeding blade stalls, the currently stalled blade experiences a decrease in the angle of attack and begins to resume normal operation. The cycle of stall cell passing continues and thereby causes the counter-rotating motion observed in the simulation. With these explanations, the expert finally concluded that the detected anomalous regions are stall cells.

5.4.2 Simulation Run without Stall (CMF = 16.0 kg/s)

The simulation run with CMF = 16.0 kg/s is a known configuration where the simulation runs consistently and is considered to be stable. It demonstrates high axisymmetry across all the passages. For verification, we ran this configuration for 2 revolutions. In Figure 5.8b we show the mass flow rate plot of this simulation and observe a flat trend. As can be seen in Figure 5.12a, the combined spatial anomaly chart of pressure and entropy barely detect any anomalous regions and hence the chart is almost clean. This confirms the effectiveness of the proposed distribution-based anomaly analysis methods in differentiating a stable and unstable operating condition. To verify the data in spatial domain, in Figure 5.12b the



(a) Combined spatial anomaly plot. Almost no anomalous regions are detected.

(b) Uncertain isocontour of pressure = 0.38. Well structured symmetric isocontour of pressure is observed.

Figure 5.12: Anomaly analysis and spatial visualization of the stable condition (CMF = 16.0 kg/s).

uncertain isocontour of pressure = 0.38 is depicted. From Figure 5.12b it is observed that all the passages have similar pressure value distributions and hence they produce similar isocontours following the axisymmetry property.

5.5 Discussions

The above results on different parameter conditions demonstrate the capability of the proposed *in situ* distribution guided approach for rotating stall analysis. Our technique shows the benefits of distribution-based analysis when the target feature, i.e. the stall cell, has no precise descriptor. Furthermore, the local region based spatial and temporal anomaly analysis also demonstrates the efficacy of our method in detecting earlier signs of the stall as depicted in anomaly charts in Figure 5.4 and 5.6, whereas the traditional approach using mass flow rate does not work well. From our comparative visualization of anomaly plots, the expert also discovered that the entropy anomaly indicates the potential

stall impacted regions more accurately than pressure anomaly alone. He concluded that by finding the co-occurrence of both pressure and entropy anomalies, a more refined stall detection technique can be obtained. Another hypothesis of the expert that the variation of data values inside a fully developed stall cell becomes less compared to the stall inception phase is also confirmed by analyzing the temporal anomaly chart where the degree of measured temporal anomaly diminishes as marked in Figure 5.6. Finally, by rendering uncertain isocontours the expert is able to visualize the data properties in the spatial domain and validate the results of the proposed approach.

5.6 In Situ Performance Study

The performance study was done using a cluster, Oakley [28], at the Ohio Supercomputer Center, which contains 694 nodes with Intel Xeon x5650 CPUs (12 cores per node) and 48 GB of memory per node. A parallel high-performance, and shared disk space Lustre was used for I/O during the simulation runs with *in situ* processing.

5.6.1 Storage Savings

A full annulus run of TURBO with 1 rotor revolution generates 5.04 TBs of raw data. In our two test cases, we ran the simulation for 8 revolutions for the first test case with CMF = 14.2 to capture the stall phenomenon and 2 revolutions for the second case with CMF =

Table 5.1: Post-hoc GMM computation time with I/O in the absence of *in situ* processing.

Component	2 revs.	4 revs.	8 revs.
Simulation raw I/O (hrs)	2.59	5.2	10.36
GMM computation (hrs)	2.38	4.82	9.52

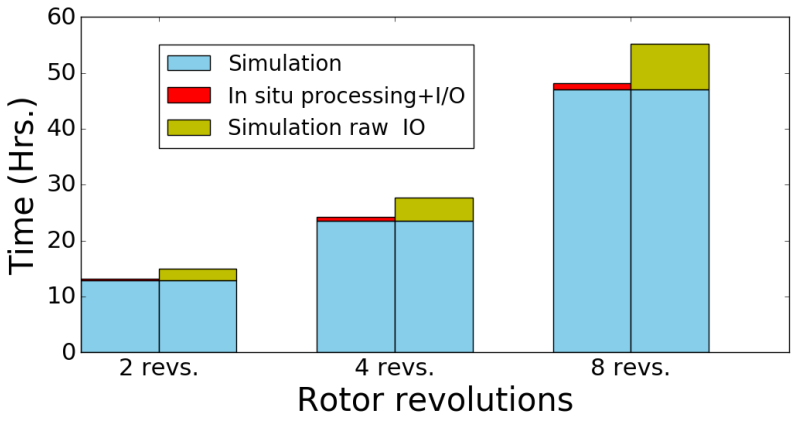


Figure 5.13: Timing comparison with and without raw output. With the *in situ* pathway, the raw I/O time can be saved.

16.0. These two runs generated raw data of 40.32 TBs and 10.08 TBs respectively. The *in situ* call was made at every 10^{th} time step which required us to process 4.032 TBs for the first and 1.008 TBs for the second test case. The simulation model has three sections: one rotor and 2 stators. In these experiments, we have stored GMMs only for the rotor which is the focused region of study, and have also stored only 2 variables. The data size for the rotor part in plot3d format is 690 MB per time step. The output of the *in situ* summarized data of two variables, in VTK multi-block format, took only 51.8 GBs for the first simulation run and 12.9 GBs for the second run, which is significantly less than the actual raw data size needed for a purely post-hoc analysis. An important point to mention is that with a different CMF condition, we would require running the simulation for a longer time and the size of raw data would be even larger.

Table 5.2: Percentage timing of *in situ* processing with half and full annulus runs. All the cases show similar percentage.

Configuration	2 revs.		4 revs.	
	Simulation	In situ	Simulation	In situ
Half annl. (164 cores)	97.3%	2.7%	97.5%	2.5%
Full annl. (328 cores)	97.63%	2.37%	97.42%	2.58%

Table 5.3: Computation time including I/O for anomaly analysis.

Anomaly type	2 revs.	4 revs.	8 revs.
Temporal anomaly analysis time (hrs)	0.56	1.11	2.19
Spatial anomaly analysis time (hrs)	0.57	1.12	2.20

5.6.2 Computation Time Savings

In Figure 5.13, we present the comparison of timings for the two scenarios: with and without *in situ* processing. The left bar in each case shows the simulation time (light blue) along with the *in situ* processing time (red), and the right bar shows simulation time (light blue) with the raw output time (green). Note that the *in situ* processing timings include the I/O time for GMM distributions, which is significantly less than the actual raw data output time. Without the proposed *in situ* processing (i.e., the GMM computation and distribution type data I/O), in addition to the mandatory raw data I/O time, extra time for estimating the GMMs post-hoc using the raw data is necessary. This extra time without the *in situ* scenario is presented in Table 5.1, where the I/O and computation time become prohibitive as the data size grows with increased rotor revolutions.

In order to study the overhead of *in situ* processing, we tested our approach using a half annulus model of TURBO which consists of 18 blades instead of 36. In this half annulus configuration, the workload for each processor was kept the same as a full annulus. In Table 5.2, we report the percentage timings of both half and full annulus *in situ* runs. From Table 5.2, we observe that the percentage time required for our *in situ* processing is only a small fraction, around 2.5% of the simulation time in both the cases. Therefore, the benefits obtained in terms of saving time in post-hoc exploration using the proposed *in situ* strategy is obvious, since we essentially bypass the simulation raw I/O, post-hoc GMM computation and I/O time completely by performing the task *in situ*. The timings of spatial and temporal anomaly analysis using the reduced GMM distribution data are shown in Table 5.3 which include the I/O time as well. Hence, by performing *in situ* processing, we have enabled a scalable and flexible post-hoc rotating stall analysis to help the expert achieve a better understanding of the phenomenon.

5.7 Conclusion

In this work, we have demonstrated the effectiveness of a distribution guided local region based rotating stall analysis. The approach that takes advantage of *in situ* processing for summarizing the important data in simulation time. Our method uses mixtures of Gaussians which facilitates flexible and scalable post-hoc analysis. By exploiting the spatiotemporal variations of distributions, statistically anomalous regions in the data are identified which have been shown to have a strong correlation to the inception of rotating stall. In the future, we would like to enable the user to steer the simulation by changing the CMF parameter and provide real-time feedback on stall analysis results to the expert. Furthermore, we would

like to extend our work to include more sophisticated uncertainty quantification capabilities and apply it to other parameter configurations.

Chapter 6: Distribution Data Driven Feature Extraction and Tracking for Time-varying Data Analysis

Effective exploration of time-varying data poses a significant challenge to the data scientists in the era of big data analytics. Since experts from diverse fields are interested in a wide range of phenomena, generally referred as *features*, efficient detection and tracking of such features is an essential task in temporal data understanding. A key component of such analysis is the ability to accurately classify the large-scale data based on the expert's interest. A visual exploration with a focus on the relevant data allows domain scientists to quickly make crucial decisions about the important scientific problems.

A majority of the tracking algorithms proposed in the past [31, 72, 104, 123, 125, 130, 131, 148, 150] have a general assumption that the definition of the feature is predetermined and hence the feature extraction process is deterministic. However, owing to the ever-increasing complexity of scientific phenomena, precise definition of a feature (i.e. the region of interest) is often unavailable. Therefore, given only a fuzzy feature description, automatic detection and tracking of such regions require novel algorithmic approaches. A key requirement of such algorithms to be considered as practical is to have the ability to quickly adapt to a refined/new feature description without going through the entire raw data again. Also, the scientific data contains features which can undergo rapid changes over both space and time and usually do not maintain any specific structure. Therefore, tracking such a region requires

robust techniques which can efficiently capture its dynamic nature and be able to detect it in consequent time steps.

In this work, we use the proposed distribution-based data summaries and show that by using distribution as a measure of feature definition given a user highlighted region in the data, such vaguely defined regions can be extracted and tracked over time. Since features in scientific data sets demonstrate properties like deformation and non-rigidity, use of distributions to represent such features adds great flexibility to our tracking algorithm. We exploit both temporal and spatial coherency of data to build a novel distribution driven feature tracking algorithm. The key observation here is that a tracking algorithm needs to account for the two key types of information:

1. possibility of the presence of motion at a specific region which might indicate the existence of a potential feature.

2. possibility of the existence of the feature at a specific region given a signature of the target feature.

Here, the term possibility reflects the *degree of belief* of a certain event. Note that the motion information can be inferred by modeling the temporal dynamics of the data while estimating the second possibility measure requires classification of data domain into spatially coherent regions that match the target definition. While none of this information mentioned above independently can give accurate results, a combination of them, however, yields an algorithm which works well for the extraction and tracking of features without a precise feature definition.

In order to efficiently capture the temporal dynamics of the feature, we estimate both the feature's location and its motion using distributions. To measure the existence of a moving

object through a local region, we employ a foreground estimation algorithm which helps us to quantify the first possibility measure stated above. Given a target region of interest, we model it as a GMM and then estimate the possibility of each local region (spatial block) containing the target which allows us to compute the second possibility measure. Finally, they are combined to generate a feature-aware classification field where high possibility valued regions are representative of the feature's location. Applying a threshold on the possibility values based on user's requirement, we are able to segment the classification field and focus on the feature. Tracking fuzzy features using the classification fields enhances the robustness of our algorithm since such fields inherently encapsulate the spatiotemporal data dynamics and allow us to analyze the feature probabilistically. Therefore our contributions in this work are as follows:

1. We take advantage a distribution driven foreground modeling scheme to detect the existence of motion in the spatial domain.

2. We propose a new algorithm which models features as a GMM and reconstructs a feature-aware classification field where the regions with high possibility values highlight the target feature.

3. Finally, we present a tracking algorithm using the classification fields and visualize the evolution of time-varying features using volume visualization techniques.

6.1 Overview of the Proposed Method

Our high-level goal in this work is to devise an efficient algorithm capable of tracking features in large-scale data when precise feature definitions are not available. We use a mixture of Gaussians (GMMs) to model the feature and employ a distribution driven

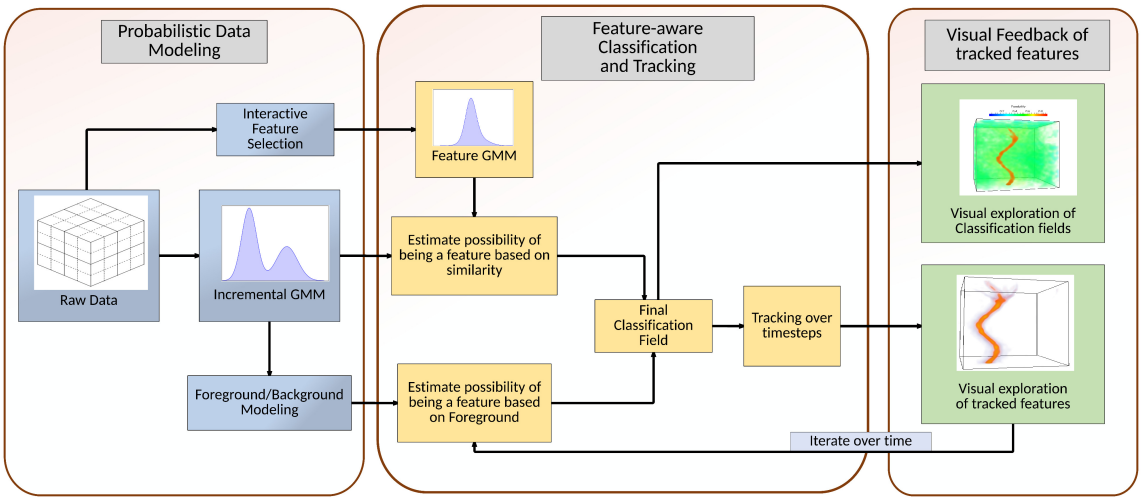


Figure 6.1: A schematic diagram of the proposed method.

technique for extraction and tracking of such features. We show that by using the local region based GMM data summaries as proposed in Chapter 4, such complex and fuzzy features can be tracked with high confidence. Figure 6.1 presents a schematic diagram of the proposed system. Initially, given a region in the data from an initial time step as the feature of interest, we construct the feature GMM using the data from the selected region. We quantify the possible existence of a moving object in the local regions (i.e. spatial blocks) by adopting a foreground estimation algorithm using the GMM based data summaries. Next, we compute the chance of a block being part of the target feature by directly comparing the GMM of all the local regions with the feature GMM. To measure the final possibility of the blocks as a part of the feature, we combine the two types of estimated information to construct a novel *feature-aware classification field* where high valued regions highlight user interested features. Finally, we demonstrate an automatic tracking technique using classification fields and explore the evolution of tracked features by interactive volume visualization techniques.

6.2 Distribution Driven Feature Classification

Our proposed tracking algorithm exploits the local region based distribution data to estimate two types of information necessary for tracking: **(a)** Estimation of moving feature by motion estimation, and **(b)** Similarity-based feature estimation. Here describe how these measures are computed and then show how they are used for tracking fuzzy features.

6.2.1 Detection of Moving Features Using Foreground Detection

In a time-varying data, if a feature has a motion, then by exploiting such motion information, the location of the feature can be identified efficiently. If a moving feature enters a region, i.e., a data block which was not present there in the previous time step, the block will encounter new data points. As described in Chapter 4, by following the incremental updating scheme of GMMs, as we update the parameters of the GMMs for this block, new Gaussians will be added to the model for accommodating the new data points in the GMM. If the new data points change the block's GMM significantly compared to its previous state, then such blocks can be characterized as containing the moving feature in the current time step. Identifying those blocks will give us the feature's possible location in space.

Data blocks containing such moving feature are often interpreted as the foreground region in the time-varying data, which demonstrate distinguishable properties compared to the relatively static background region. The possibility of a block being part of the foreground can be estimated by the number of new data points the block has encountered in the current time step. If the majority of the points are new, then the block should be classified as a foreground with high confidence. Since, we model the temporal distribution using an incremental update scheme described previously in Chapter 4, at any given instant,

the GMMs at each block represents a temporal distribution of the block created using the data observed in earlier time steps. The Gaussians with higher weights in the block GMM are the representative of the portion of the data which the block has encountered consistently in the past and the high weights reflect that. Therefore, when a new data point comes, it will not find a match with any of such Gaussians and will add a new Gaussian in the model. We aim to identify those new data points and by doing so we can quantify the possibility of the block being part of the foreground.

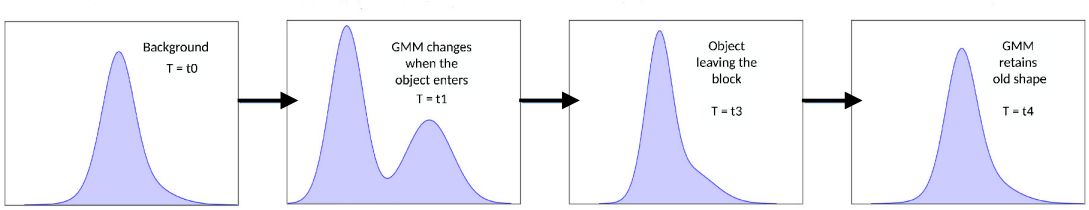


Figure 6.2: Evolution of the GMM of a block while an object moves through it.

While observing new data points during the distribution estimation, we keep track of all the data points that: (1) do not match any existing Gaussians with weight higher than a threshold T, and (2) matches with a newly created Gaussian. All such points represent the new data points that the block has observed in the current time step. As the number of such points increase, the chance of that block of being a foreground also increases. So, the possibility that a block containing a foreground object is quantified by the fraction of the new data points to the total number of observations in the block:

$$POS_{foreground,t}(b_{i,t}) = q_{i,t}/n_{i,t} \tag{6.1}$$

where $q_{i,t}$ is the number of observations that satisfies either the clause (1) or (2) stated above, and $n_{i,t}$ is the total number of observations for the i^{th} block at time t. The value of $POS_{foreground,t}(b_{i,t})$ is always between 0 and 1. As we iterate over all the time steps, we measure this possibility value for each block per time step and keep this information. Later we will use this information and combine it with another possibility measure for the final classification of each block as being part of a feature of interest.

In Figure 6.2 we show the conceptual evolution of a GMM of a block over a sequence of time steps as an object moves through it. At time $t0$ the block is considered as a part of the background. However, as the object enters the block at time $t1$, the distribution changes and the possibility value of this block is a foreground increase. From $t1$ - $t3$ the block shows evidence of being a part of a foreground object and finally when the object exits the block, the GMM returns back to its old shape, as can be seen at time $t4$.

For each time step, during the estimation of the GMMs, we store an additional possibility value for each block measured by Equation 6.1. Note that in case the GMMs are estimated *in situ*, this measure will also be estimated *in situ*. Furthermore, the estimation of the possibility value presented in this section is oblivious to the target feature. But this provides us a way to measure the chance of a block being part of a moving object which can be a potential feature. We acknowledge that for any robust feature extraction system, detection of motion component is essential for improving the tracking results [108]. However, since this information does not consider the target feature definition, the extracted regions may require further refinement based on user's need. Also, if the feature does not have a strong motion component then we can not make any definitive conclusion about the feature from only foreground information. To remedy this, we introduce another measure which estimates the

possibility of a block being part of a feature by observing the feature distribution and helps us to finally classify the blocks.

6.2.2 Distribution Driven Classification Based on Feature Similarity

Our goal is to measure the possibility of each block being a part of the target feature. This possibility measure exploits the spatial coherency and extracts the regions which contain similar distributions as the target GMM. We measure the similarity between the GMM of each block and the feature GMM. There are several techniques available for measuring the similarity between two GMMs such as the Kullback-Liebler divergence (SKL) and, Bhattacharyya-based distance measures [129, 163]. However, SKL needs Monte-Carlo approximation for its computation which makes it computationally expensive. Also, it was reported that the Bhattacharyya-based similarity measure is generally fast and leads to good results [129]. So, for measuring the similarity between GMMs, we have used the Bhattacharyya-based distance measure which can be expressed as:

$$\Psi(p,p') = \sum_{i=1}^{n}\sum_{j=i}^{m} \omega_i \omega'_j \mathscr{B}(p_i, p'_j) \tag{6.2}$$

where p and p' are the GMMs and n and m are the number of mixture components of GMM p and p' respectively. $\mathscr{B}$ is the Bhattacharyya distance between two Gaussian kernels and is defined as:

$$\mathscr{B}(p,p') = \frac{1}{8}(\mu-\mu')^T \left(\frac{\Sigma+\Sigma'}{2}\right)^{-1} (\mu-\mu')^T + \frac{1}{2} ln \left[\frac{|\frac{\Sigma+\Sigma'}{2}|}{\sqrt{|\Sigma||\Sigma'|}}\right] \tag{6.3}$$

here μ, μ' and Σ, Σ' are the mean and covariance of the Gaussian kernels p, p' respectively. After computing the values of $\Psi(\cdot)$ for all the blocks, the values are normalized. Given the feature GMM f_t at time t, the possibility of i^{th} block $b_{i,t}$ being part of the feature

at time t is computed as:

$$POS_{similarity,t}(b_{i,t}) = 1 - \Psi_{norm}(b_{i,t}, f_t) \tag{6.4}$$

Note that the value of $POS_{similarity,t}(b_{i,t})$ is always between 0 and 1 and is maximum for the block which matched best with the feature GMM f_t and as the degree of match reduces, i.e., the similarity between feature GMM and block GMM decreases, the value of $POS_{similarity,t}(b_{i,t})$ also drops.

So far, we have described two types of possibility values for each block and each of them tries to classify a block of being part of a feature. In the following, we demonstrate how this two information is combined effectively to obtain a more accurate classification of all the blocks instead of using them individually.

6.2.3 Feature-Aware Classification Fields

In this section, we present the method for combining the possibility values discussed in earlier sections to achieve a final robust classification of all the data blocks. Such a classification will assign higher values to the blocks which are more probable of being part of the target feature. Note that the possibility values defined earlier tries to analyze the block from different perspectives. For the first method, a high value of $POS_{foreground,t}(b_{i,t})$ for a block signifies that there is a high chance of the existence of a feature in that block. However, since it does not directly consider the target feature definition, we can not come to a certain conclusion by just using this measure. To complement this deficiency, we have incorporated another possibility measure $POS_{similarity,t}(b_{i,t})$ which calculates the possibility by taking into account the similarity between a block GMM and the user interested feature GMM. However, when the feature distribution is not clearly separable from the background or the feature size is sufficiently small, the performance of this approach deteriorates. So,

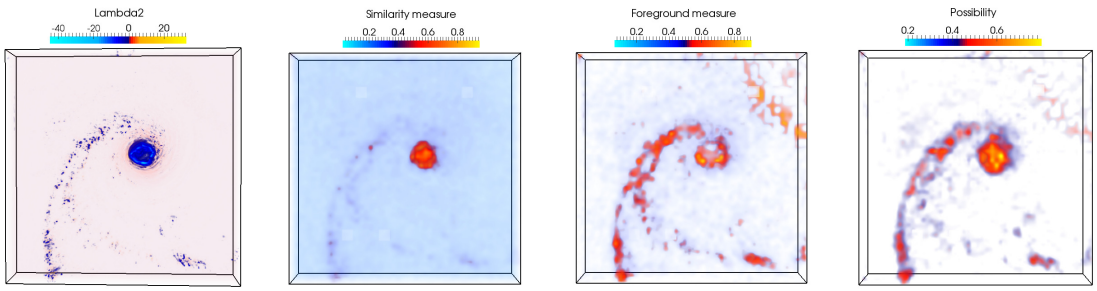
we can not always completely rely on the similarity based measure for the classification. Therefore, we seek a consensus between the two measures to classify all the data blocks with high confidence.

In statistical theory, there exist several techniques for combining multiple hypotheses for inference. In our case, we have two hypotheses (possibilities of being a feature) and they can be combined either linearly or non-linearly. A popular and effective technique for a linear combination of hypotheses is presented in [36], called the *linear opinion pool*. This technique is fast to compute and suitable for interactive algorithms such as ours. Hence, following this strategy, the two possibility values are combined as:

$$POS_{feature}(b_i) = \gamma * POS_{similarity}(b_i) + (1-\gamma) * POS_{foreground}(b_i) \tag{6.5}$$

Here γ acts as the mixing parameter which plays an important role. The value of γ always chosen between 0 and 1 which determines how much contribution each of the possibility measures will have in the final classification. Based on the knowledge of experts, this parameter can be selected carefully to enhance the robustness of classification. If a data set contains a moving feature and scientists are interested in tracking such feature then the value of γ is chosen accordingly such that the contribution of $POS_{foreground}(b_i)$ is more in the classification and similarly if the target feature does not show a strong movement over space, we can set a high γ values to increase contribution of $POS_{similarity}(b_i)$. In the absence of specific knowledge about the feature dynamics, we can set $\gamma = 0.5$, which accounts for the equal contribution of both the measures in the final classification.

Once all the data blocks are classified and a possibility value is assigned to them, a scalar field can be constructed using the possibility values for all the points in the block. Such a field is called the *feature-aware classification field* and Algorithm 2 presents the pseudo-code for constructing such field. Direct visualization of such a field can convey the



(a) Lambda2 field with target feature.

(b) Similarity measure of vortex feature.

(c) Foregorund possibility of the feature.

(d) Classified field.

Figure 6.3: Feature estimation exploiting spatial and temporal coherency using hurricane Isabel data at T=34.

Algorithm 2 Construct Feature-Aware Classification Field

1: Input: $GMM(feature), timestep$
2: **for all** block b_i **do**
3: Compute $POS_{similarity}(b_i)$. (Equation 6.4)
4: Compute $POS_{feature}(b_i)$. (Equation 6.5)
5: Use $POS_{feature}(b_i)$ for construction of classification field.
6: **end for**

information regarding the likelihood of the feature's existence at current time step. Note that this field is generated by combining the two possibility measures which are derived directly by exploiting the spatial and temporal coherence of the time-varying data. In the absence of a precise feature definition, such a classification field allows scientists to observe the evolution of the features in a time-varying data.

Figure 6.3 demonstrates the usefulness of having two key possibility measures, used in this work and shows why just a single measure is not sufficient. In Figure 6.3a the λ_2 field of hurricane Isabel data is shown for time step 34 where the vortex region and its spread is highlighted. Figure 6.3b depicts the $POS_{similarity}(b_i)$ field where it is clearly visible that

the vortex core is identified only and the smaller band of vortices are mostly missing or identified with low confidence. However, in Figure 6.3c we see the $POS_{foreground}(b_i)$ field which captures the smaller bands of vortices with higher accuracy, but the detected core region is not as accurate as in Figure 6.3b. Finally, Figure 6.3d presents the combined feature-aware classification field which is able to preserve both the core and the small vortex bands with high accuracy. This means that tracking using classification fields will yield robust results since it is able to capture the target feature in detail.

For any tracking algorithm, the accuracy of extraction of features is an important step. If the extracted features are not reliable then the tracking may not give a meaningful result to the scientists. Since we are dealing with an uncertain feature definition, the proposed technique solves an important problem in automatically detecting the feature evolution over time. In the next section, we present a technique for tracking features using the feature-aware classification fields.

6.3 Tracking Using Feature-Aware Classification Fields

Feature tracking in visualization is an important task and researchers have looked into this problem in the past [72, 104, 123, 125, 130, 131, 148]. Even though above techniques achieve stable tracking results, the feature extraction part of those methods rely on the precise feature description. In this work, we extend the capability of the feature tracking techniques by introducing a new distribution driven method, which is able to track volume features that are selected directly from raw data interactively, therefore without any precise description. We have used GMM of the selected region to model the target feature. To make the feature extraction more accurate and robust, we first generate a feature-aware classification field using GMM based summary data as described in the earlier section. Such

a field allows us to classify the data by their relevance to the user interested feature. In the classification field, regions with high possibility values represent the existence of the feature of interest and they can be easily visualized and explored. We perform tracking in this feature-aware space because the classification field allows to easily extract the feature by applying a suitable user-specified threshold on the possibility values.

Algorithm 3 Tracking In Feature-Aware Classification Field

1: Input: $GMM(b_i)$, $GMM(feature) : \forall i \in 1,2,..n$
2: Initialize $f_{target} := GMM(feature)$
3: **for all** t in T **do**
4: Generate Feature-Aware Classification Field($GMM(feature)$, t) (Algorithm 2)
5: Thresholding ($>= poss_{th}$) on the Classification field.
6: Apply connected component algorithm on the thresholded results.
7: Compute distance between centers of target feature and all the detected regions from the current time step.
8: Find the best match l with the minimum distance to the target feature f_{target}.
9: Set $f_{target} := l$
10: **end for**

A visual inspection of the classification field using interactive volume visualization techniques allows scientists to easily locate their feature of interest by focusing on the high valued regions in the field. Our method allows the users to inspect the classified field of an initial time step and provide a suitable threshold possibility value ($poss_{th}$), which we apply to the classification fields of later time steps for automatic extraction of the target feature. After the threshold ($poss_{th}$) is applied to the classification fields, a connected component based region growing algorithm is employed to the result of the thresholding to extract all the connected features. Each such detected region is treated as a separate feature. A match with the given target feature is found by using a distance-based method as was described earlier in [123], where the Euclidean distances between the centers of the target feature

f_{target} at time t with all the other detected regions at time $t+1$ are computed and the region with the minimum distance is tagged as the target feature f_{target} in time $t+1$. This process is repeated for consecutive time steps to continue the tracking process. Algorithm 3 sums up the steps of our tracking algorithm which implicitly calls Algorithm 2 for generating the classification fields for each time step and tracks the feature of interest using it. In Algorithm 3, T represents the final time step. As can be seen, at the end of calculation of every time step, the f_{target} is updated with the best-matched feature l from the current time step which is used in the next time step as the reference feature.

In some complex scenarios, apart from changing the position and size, the target feature may undergo several evolutionary events such as birth, split, merge, and dissipation etc. Unlike traditional automatic feature tracking systems where all the existing features are extracted based on a predefined feature definition and tracked over time, we are more concerned with tracking a specific region of interest which has been identified vaguely from a region directly specified from the raw data. Therefore, we do not require to focus on a feature birth process explicitly. To detect the dissipation of a feature, we set an upper limit to the matched minimum distance value. The motivation is that given sufficient temporal resolution, the evolutionary change of a time-varying feature happens gradually and if a sudden anomaly is detected during the tracking in terms of the matched minimum distance, the event needs the further attention of experts. Therefore, during tracking, we compare the value of the matched distance at each time step with the predefined upper limit and if the value is greater than the limit, we finish tracking and report the time step back to the user for further investigation. Events like feature split or merge can be detected in our system by keeping track of the feature mass. In our tracking algorithm, we measure the mass of the identified feature as was described in [123] at each time step and compare it with the previous

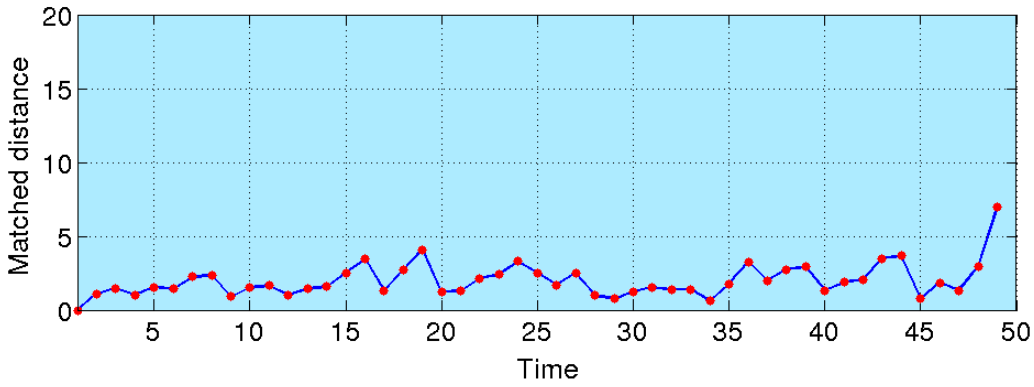
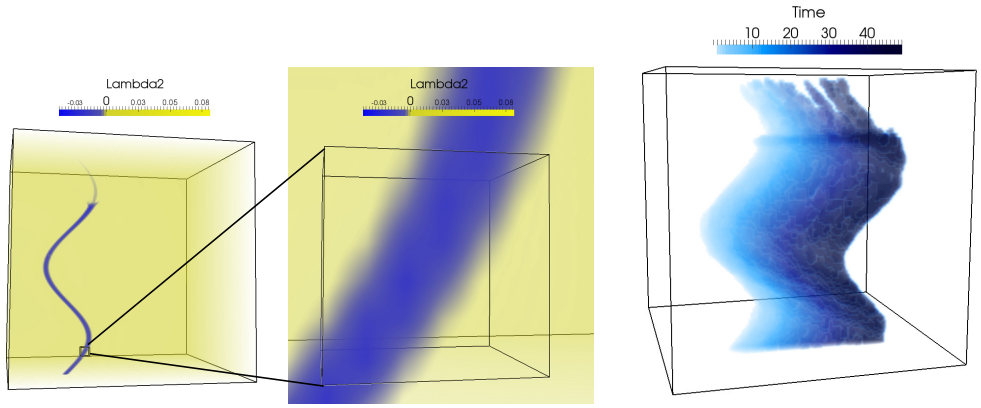


Figure 6.4: Matched distance values over time for Tornado data set.

time step. A sudden and large change of mass indicates a potential feature split or merge event, where an increase in mass indicates feature merge and decrease in mass signifies feature split. Even though a big drop in the mass may also reflect a shrinking/disappearing event, however, by keeping track of such event, our system is able to detect those time steps and they are reported back to the users for further investigation. The proposed method allows users to set a predefined threshold value based on their domain knowledge for the change of mass between two consecutive time steps and if a change of more than the threshold is detected, the time step is marked and is reported back for further exploration.

In Figure 6.4, we show one example plot of the minimum matched distance values for all the time steps for the Tornado data set. The upper limit to the matched distance was set to 15 for this experiment. As we can see that, for all the time steps the proposed method was able to extract and track the target feature with high consistency which is reflected by the low matched distance values throughout all the time steps.



(a) Selected feature in the Tornado data set and a zoomed in view of the selected region.

(b) Temporal trace volume of the tracked feature.

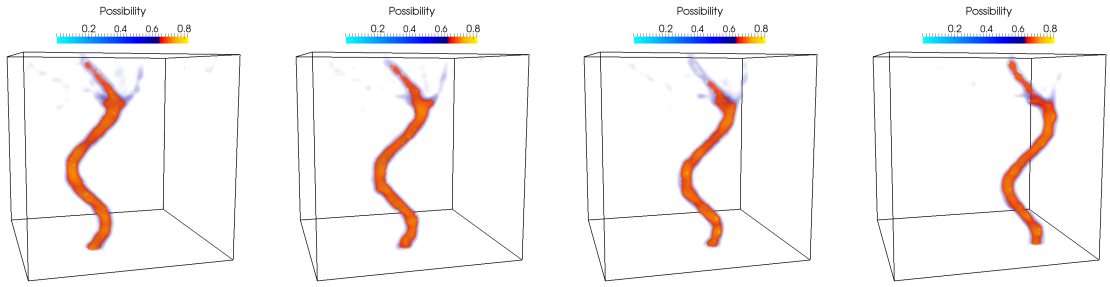
Figure 6.5: Feature tracking in Tornado data set.

6.4 Case Studies and Science Applications

In this section we demonstrate the effectiveness of the proposed method in extracting and tracking features with fuzzy definition, using several scientific data sets. In all the case studies, using a maximum of 3 Gaussians produce stable results and so the maximum number of Gaussians per GMM is set to 3 for the experimentation.

6.4.1 Tracking Vortex Core in an Analytical Tornado Data Set

The first experiment is to study a Tornado data set of dimension $128 \times 128 \times 128$, containing velocity vectors at each grid point, generated by an analytical function [39]. The data set has 50 time steps and simulates a tornado-like vortex structure. For this case study, we have modified the analytical equation so that the center of the tornado changes position with time. The block-size of $4X4X4$ is used for the experimentation. The goal is to track the vortex core of the tornado.

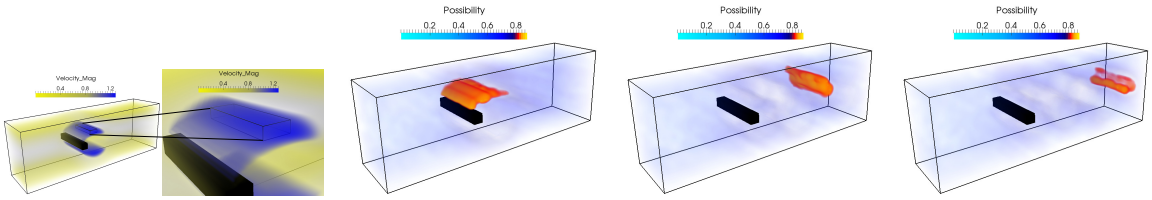


(a) T=10. (b) T=20. (c) T=30. (d) T=49.

Figure 6.6: Extraction and tracking in Tornado data set. The vortex core is tracked over time and the results of 4 selected time steps are shown.

Figure 6.5a presents the selection of the fuzzy vortex region from the first time step of the data. We have computed the lambda2 (λ_2) vortex criterion using the velocity field for measuring *vortexness* at each spatial point. Even though theoretically negative values of λ_2 criterion represent vortex region, deciding a precise threshold using a λ_2 value is difficult and often needs manual tuning. Nevertheless, visualizing the lambda2 field allows experts to mark the region in the data where the vortex exists. From the highlighted region, we extract all the points and fit a GMM on those data points to represent the region (*feature of interest*) using its distribution. After that, the proposed extraction and tracking method is applied to all the other time steps of the data for automatically extracting and tracking the vortex region.

Figure 6.6a, 6.6b, 6.6c, and 6.6d depict the tracked feature of interest at 4 time steps. Even though we have estimated the feature distribution by only using the sample points from the initial highlighted region as shown in Figure 6.5a, our extraction algorithm is able to recover the complete connected vortex core region from the data accurately. For the construction of feature-aware classification fields, the value of γ is set to 0.5 which



(a) Selected feature. (b) T=25. (c) T=35. (d) T=40.

Figure 6.7: Extraction and tracking of the selected feature in 3D Flow around a cylinder data set. High velocity feature at 3 selected time steps have been shown.

means the final classification will have 50% contribution from the foreground component (the motion component) and rest of the 50% contribution comes from the similarity based measure. In this work, we perform tracking in the classified fields and focus on the high valued regions. For extracting the target region which strongly represents the feature of interest, regions with a possibility value higher than 0.65 are considered in this study. The results reflect that our method is able to extract and track automatically the feature over time.

To provide a comprehensive view of the tracked feature, we also create a temporal trace of the feature by constructing a scalar volume using time steps as the scalar value at each grid point. At the end of tracking, the temporal trace volume is obtained which shows the dynamic transition of the tracked feature over space. In Figure 6.5b we show such a feature trace volume of the Tornado data set. The movement of the tracked feature from left to right is evident from the gradual transition of color. When the feature has a continuous motion, then the trace volume allows the experts to visualize the overall temporal evolution of the feature efficiently.

6.4.2 Tracking Vortices in a 3D Flow Around a Cylinder Data Set

This case study demonstrates feature extraction and tracking in the 3D Flow around a cylinder data Set. This is a 3D time-dependent incompressible flow data with a Reynolds number of 200 and a square cylinder has been positioned symmetrically between the two parallel walls. The data set consists of velocity vectors and simulates a complex periodic vortex shedding phenomena which are well known as the *von Kármán vortex street*. This is a direct numerical Navier Stokes simulation by Simone Camarri and Maria-Vittoria Salvetti, Marcelo Buffoni, and Angelo Iollo [26] which is made publicly available [64]. We have used a uniformly re-sampled version which has been provided by Tino Weinkauf and used in von Funck et al. [143]. The data is represented by a grid of $192 \times 64 \times 48$ and there are total 102 time steps. For experimentation, $4X4X4$ block-size is used.

In order to explore the periodic flow pattern and the vortex shedding which are produced by the von Kármán vortex street, study of the velocity field is useful. The high-velocity waves show the periodic patterns exist in the data and help scientists to understand the phenomena in greater detail. For tracking the rapidly moving high-velocity vortex street, we have used velocity magnitude field in this case study.

In Figure 6.7a, the region with high-velocity magnitude is selected as the region of interest which moves periodically through the simulation grid. Tracking of such region is non-trivial as the feature is hard to define by a hard threshold value. Furthermore, isolation of such feature consistently over the time range poses significant challenges. We have applied our extraction and tracking algorithm in this data set and Figure 6.7 demonstrates the results we have obtained. In Figure 6.7b, 6.7c, and 6.7d we show the tracked feature which moves forward over time. For creating the classification fields, the value of γ is set to 0.7 which means that the foreground measure contributes more than the similarity measure

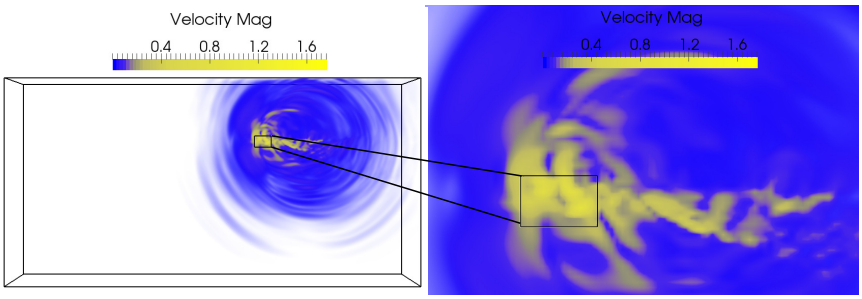
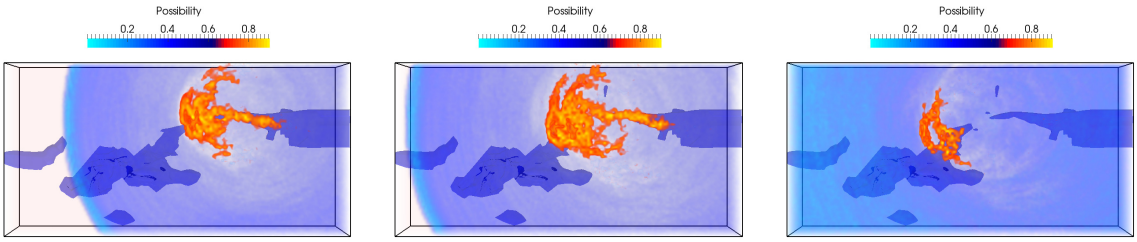


Figure 6.8: Selected feature in Earthquake data set and a zoomed in view of the selected region.

in this experiment and for isolating the feature, we have used possibility value larger than 0.8. From the results depicted in Figure 6.7, it is evident that the proposed method is able to isolate and track the selected feature over time effectively.

6.4.3 Earthquake Shock-wave Tracking

Our next case study uses an Earthquake data set which is a time-varying data consisting of wave velocity vectors. The dimensions of the 3D volume data is $750 \times 375 \times 100$ and we have used 100 time steps to perform the experiment. The data set describes a simulation of the earthquake of magnitude 7.7 on the Southern San Andreas Fault and was generated using TeraShake 2.1. The TeraShake 2.1 simulation was performed by scientists at the Southern California Earthquake Center (SCEC) and researchers at San Diego Supercomputer Center (SDSC). It records the velocity vectors of the earthquake waves spreading over time. For this case study, we have considered the magnitude of the velocity field since studying the high-velocity waves, the direction and intensity of the earthquake can be understood in detail. The data is divided into blocks of $5X5X10$ for experimentation. In Figure 6.8, the region with high-velocity magnitude is selected as the region of interest. Figure 6.9 depicts



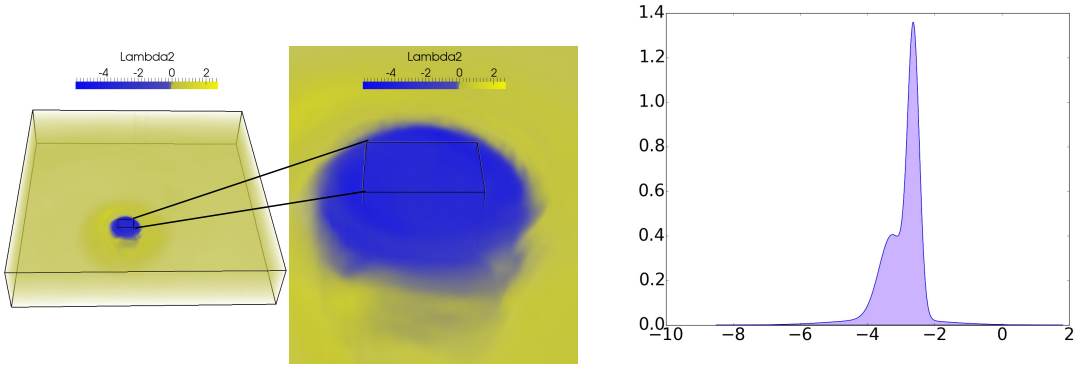
(a) T=10. (b) T=20. (c) T=30.

Figure 6.9: Extraction and tracking of the propagation of high-velocity shock waves in Earthquake data set. Results of 3 selected time steps are presented.

the result of tracking such high-velocity region over the selected time range. In Figure 6.9a, 6.9b, and 6.9c we present the tracked target region at 3 selected time steps 10, 20, and 30 respectively. The images show how the high-velocity waves propagate over time. We also visualize the land and the basin regions with the feature to reflect the areas affected by the strong wave. While creating the feature-aware classification fields, the value of γ is set to 0.3 and in tracking phase, we have considered possibility values higher than 0.72 for isolating the feature of interest at each time step. Results presented in Figure 6.9 show that the proposed method is able to detect and track the strong wavefront feature.

6.4.4 Extraction, Tracking and Comparative Analysis of Hurricane Eye Using Isabel Data Set

Next, we present the fourth case study using Hurricane Isabel data which is a time-varying data set containing a vector field of wind velocity. The data set is a courtesy of NCAR and the U.S. National Science Foundation (NSF), and was created using the Weather Research and Forecast (WRF) model. The data set corresponds to an actual physical space of 2139km (east-west) x 2004km (north-south) x 19.8km (vertical), which is represented



(a) Feature selection in Hurricane Isabel data.

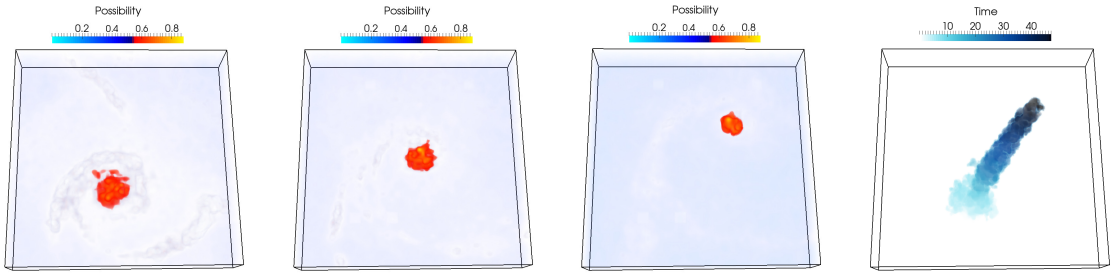
(b) Estimated GMM of the feature.

Figure 6.10: Selected feature in Hurricane Isabel data set, a zoomed in view and the GMM of the selected region.

by a grid of $250 \times 250 \times 50$ and there are total 48 time steps. For experimentation, $5X5X5$ block-size is used.

An important task in this data set is to extract and track the temporal evolution of the low-pressure eye (core) of the storm system where a strong vortical flow exists. As discussed earlier in [25], accurate tracking of the location and spread of the eye is critical to understanding the strength of the storm. We have computed the lambda2 (λ_2) field using the velocity field for the initial selection of the vortex region. Note that, the use of a hard thresholding on the λ_2 value is not always robust and often requires user intervention. Also, the dynamic nature of the feature makes the task of tracking challenging.

In Figure 6.10a, the feature selection is demonstrated. It is evident that the feature boundary is fuzzy and separating it using a hard threshold is cumbersome. Figure 6.10b displays the GMM which is estimated from the selected region and it is treated as the feature definition in the proposed tracking algorithm. The tracking algorithm, described in Algorithm 3 is applied on the entire data set over all the time steps using the estimated

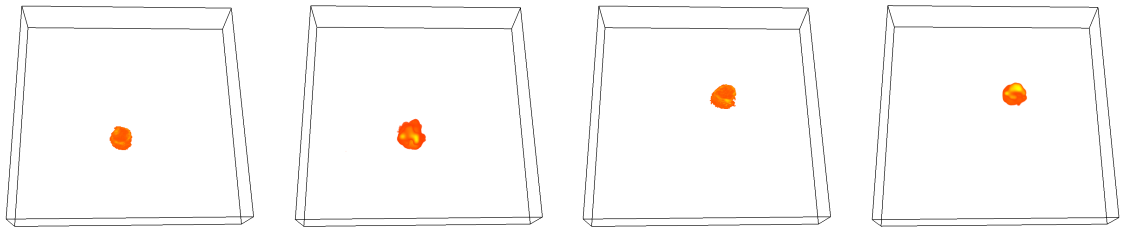


(a) T=11. (b) T=28. (c) T=41. (d) Temporal trace of the feature.

Figure 6.11: Extraction and tracking of the vortex at Hurricane eye in Isabel data set. Results of tracking of 3 selected time steps are shown.

GMM as the feature of interest. At every step of the Algorithm 3, it internally calls the Algorithm 2 for the construction of the feature-aware classification field. Since the feature of interest has a motion in the space, we use $\gamma = 0.3$ for capturing such motion information while computing the classification fields. Final tracking is done on the classification fields and the feature is extracted at each time step and visualized using volume visualization techniques. In Figure 6.11, the detected vortex region (the Hurricane eye) is presented for 3 selected time steps to show the effectiveness of our method. For isolating the feature in the final classified possibility fields, possibility value of greater than 0.55 is considered. From Figures 6.11a, 6.11b, and 6.11c it is evident that the proposed method is able to detect and track the eye of the storm with high accuracy. Also in Figure 6.11d we show the temporal trace volume of the feature to present the overall evolution of the target feature.

Figure 6.12 shows a comparison between our method and the correspondence based volume tracking method [123]. We have implemented the volume tracking algorithm for this comparative study. Since the volume tracking method requires a predefined precise feature description for tracking, we have used $\lambda_2 < -0.001$ as our feature definition in this



(a) Feature extracted using volume tracking method at T=15.

(b) Feature extracted by the proposed method at T=15.

(c) Feature extracted using volume tracking method at T=35.

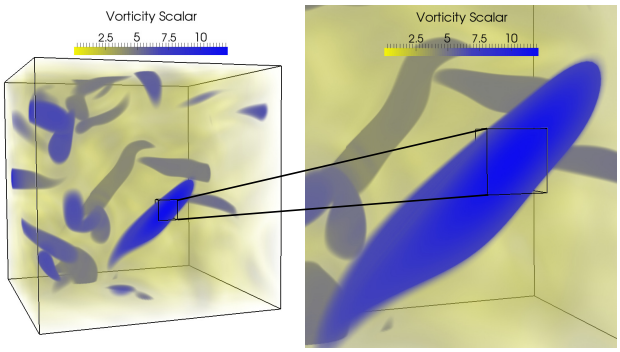
(d) Feature extracted by the proposed method at T=35.

Figure 6.12: A comparison between the volume tracking method and the proposed algorithm. The proposed method is able to produce comparable results with a fuzzy feature descriptor.

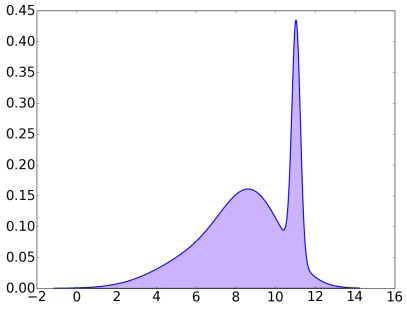
study. Figure 6.12a and 6.12c show the results obtained from the volume tracking method for time steps 15 and 35 respectively, and Figure 6.12b and 6.12d show the extracted feature for the same time steps with the fuzzy feature description. It can be seen that the results obtained by the proposed method are very similar to that of the volume tracking method with only minor differences. It shows that the proposed method is capable of extraction and tracking of features which are only vaguely defined. Therefore, this method enhances the capability of the existing feature tracking algorithms by providing a novel way of dealing with fuzzy volume features. Furthermore, the temporal trace volume depicted in Figure 6.11d also confirms that our method is robust and can track the time-varying features with high accuracy consistently.

6.4.5 Feature Tracking in Vortex Data Set

Our final case study shows the result on a Vortex data set which is a pseudo-spectral simulation of coherence vortex structures. The dimension of this data set is $128 \times 128 \times 128$ and is divided into blocks of $4X4X4$. The scalar variable in the data is vorticity magnitude.



(a) Feature selection in Vortex data set with a zoomed in view.

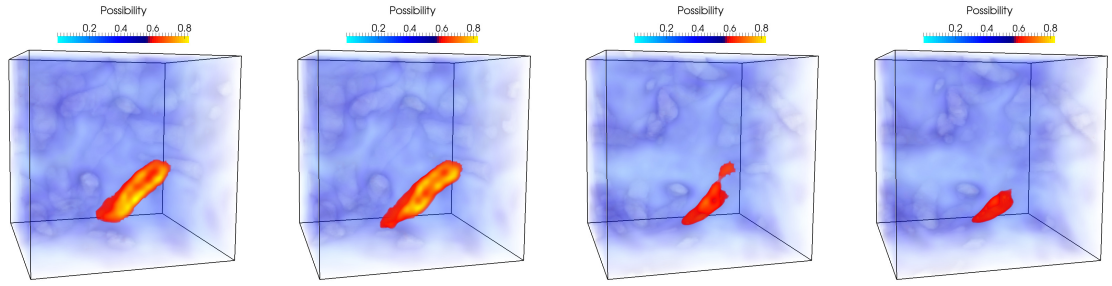


(b) Estimated GMM of the user interested feature.

Figure 6.13: Selected feature in Vortex data set, a zoomed in view and the GMM of the selected region.

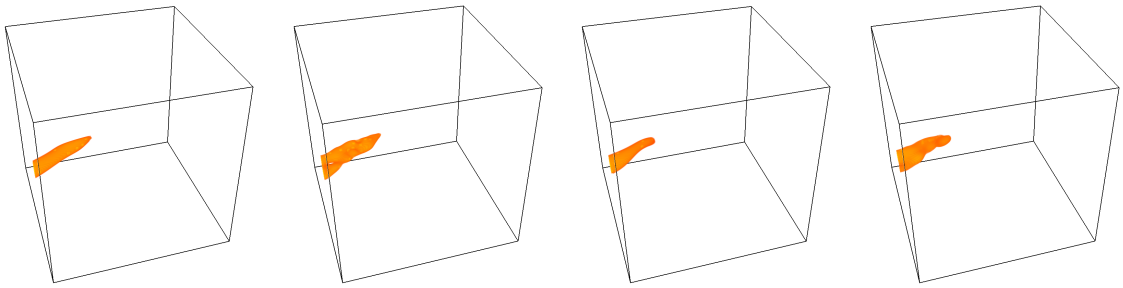
We used 30 time steps of the data set to demonstrate the effectiveness of our algorithm on this data set. The data set contains several tubular vortex cores which undergo rapid shape changes and complex events such as split, merge, creation, and dissipation.

Figure 6.13a shows a specific region which is selected for this case study from the first time step. Figure 6.13b depicts the GMM estimated from the selection as the feature to be tracked. From Figure 6.13a it is visible that there are several vortex regions in the data set and therefore, explicit correspondence is important in this case for accurate tracking of the selected feature. We have applied our tracking algorithm, presented earlier, for tracking the selected feature. In Figure 6.14 we demonstrate the results of extraction and tracking by showing the tracked feature at 4 selected time steps. Since, the features in this data set change their shape rapidly and the motion is not a dominant component, the value of γ is set to 0.15, so that we take larger contribution (85%) from the similarity based possibility measure to achieve higher accuracy in tracking. After the feature-aware classification fields are constructed, we use possibility value 0.58 as a threshold for identifying all the connected



(a) T=3. (b) T=6. (c) T=14. (d) T=15.

Figure 6.14: Extraction and tracking using Vortex data set. Tracked feature for 4 selected time steps are displayed.



(a) Feature extracted using volume tracking method at T=3.

(b) Feature extracted using proposed method at T=3.

(c) Feature extracted using volume tracking method at T=8.

(d) Feature extracted using proposed method at T=8.

Figure 6.15: A comparison between the volume tracking method and the proposed algorithm using Vortex data set.

regions in the data set. Then by applying the Algorithm 3 we detect and track the target feature.

From the Figures 6.14a - 6.14d, it is evident that the key feature gradually dissipates as time increases. Finally, the feature dissipates at time step 22 which is detected in our system by the predefined upper limit set for the distance value between the matched feature and tracked feature from previous time step. A detailed visual exploration shows that our method

is able to show the feature split phenomena clearly. From the tracked results presented in Figure 6.14, time steps $14 - 16$ are significant because the target feature undergoes a split in this time range. In Figure 6.14c we can see that the feature is about to split into two segments and the split happens in time step 15. As the split happens, we continue to track the closest component of the feature and report the time step where the split has happened. In our current system, time step 15 is detected to have a potential split since in this time step mass of the feature decreases 30.9% compared to its previous time step. Another observation here is that, as the feature gradually shrinks in size, the possibility values also drop. This trend is visible from the sub-figures in Figure 6.14 where the higher time steps show less yellow regions on the feature and more red regions, indicative of low possibility values.

In Figure 6.15 we present a comparison between the proposed method and the volume tracking method. This data set has multiple complex features (vortex cores) which are identified as isolated segmented regions where the segmentation criterion used is the region with scalar value $>= 7.0$. After the segmentation is done, all the isolated regions are treated as separate features and a segmented region is selected as the target feature and applied the volume tracking method to track it over time. Also for applying the proposed method on the same feature, a small region is selected from the target feature as a representative sample. Figure 6.15a and 6.15c show the extracted feature obtained by the volume tracking method at time steps 3 and 8 and Figure 6.15b and 6.15d show the results produced by the proposed method without the background context. By observing the results depicted in Figure 6.15, it is evident that the proposed method generates very similar results compared to the volume tracking algorithm and can robustly track the feature. So, in the absence of a predefined feature definition, the proposed method presents a tracking framework which allows users

to highlight their target region of interest directly in the raw data and automatically track it over time efficiently.

6.5 Performance Study

In Table 6.1 the timings are reported for the test cases. The classification field generation time includes I/O time and can be done separately before the tracking process. The incremental estimation of the GMMs and the foreground estimation algorithm requires only a linear scan of the raw data. The incremental algorithm used here is significantly less expensive in estimating the GMM parameters compared to the off-line EM algorithm and also suitable for streaming data/in-situ frameworks. The estimated GMMs are used for computing the feature similarity measure and finally, the previously measured foreground information is combined with the similarity measure to generate the classification fields. The advantage of the method is that even when the feature is changed, the algorithm does not require the access to the raw data and can generate the classification fields using the previously computed GMMs. It only needs raw data access for the first time step to re-estimate the feature GMM. Since we use a mixture of Gaussians to model the data, the storage complexity is significantly low as we have to store only the parameters of the GMM and a possibility value for each block for future use. For creating the final possibility field a segmentation based region growing algorithm is used. Table 6.1 shows the tracking time separately for all the case studies. Also, since the computation is done block-wise, therefore for significantly large data sets, the algorithm can be parallelized by distributing data over multiple nodes and processing each block in parallel.

Table 6.1: Data set descriptions and average CPU Time performance per time step for different computation components.

Data Sets	Block size	Classification Field Creation (secs.)	Tracking (secs.)
Tornado	$4X4X4$	3.580	1.432
Hurricane Isabel	$5X5X5$	5.041	0.737
Vortex	$4X4X4$	3.601	0.492
Earthquake	$5X5X10$	51.133	15.83
Flow Around a cylinder	$4X4X4$	1.1724	0.3378

6.6 Discussions

For any feature tracking algorithm, a robust extraction method is a key component, because if the extracted features are not reliable, then tracking them can lead to misleading outcomes. Almost all of the previous tracking works have assumed a predetermined feature definition for the extraction stage. However, little attention is paid when the precise feature definition cannot be obtained. In this work, we extend the capability of feature extraction and tracking algorithms by proposing a new distribution driven approach which is able to deal with the uncertainties inherent in the given fuzzy feature definition and allow reliable feature extraction and tracking.

For tracking *predefined* volumetric features, researchers have proposed comprehensive techniques [71, 131]. However, one potential disadvantage of those techniques is that, if the feature definition is changed, the algorithm would require going through the raw data again. In another work, a texture-based feature tracking algorithm was proposed in [25] where high-dimensional textural attribute vectors are used for feature representation. The technique obtained accurate results even with low temporal sampling. But the technique required

to find an appropriate neighborhood window for searching the feature. Also, the drifting problem is recognized as a limitation of this work. A recent trajectory based feature tracking algorithm [125] has demonstrated promising results, but it is limited to only data sets containing additional particle data and its accuracy is dependent on the particle density. A more general flow pattern extraction technique for 2D flow fields has been introduced earlier in the works of Schlemmer et al. [128] based on moment invariants. The method is able to detect critical points in flow fields and also find user defined (in circular domain) complex flow patterns from the data efficiently. The work presented in [128] and the proposed method, both achieve feature estimation by utilizing a pattern matching approach where we use distributions as a statistical pattern for the target feature. The goal of our algorithm in this work is to efficiently track vaguely defined volume features in 3D time-varying scalar fields.

Our method efficiently exploits both spatial and temporal coherency present in the data and utilizes them to compute the two key information: (1) motion and (2) similarity with target feature distribution. Since none of this information alone is sufficient for achieving a robust feature extraction, we combine them to construct a feature-aware classification which helps us to extract and track key features. So, in the absence of precise feature definition, the proposed method allows tracking of fuzzy features robustly. Another advantage of the proposed method is the use of the incremental framework for data modeling. Since the model does not require all the data beforehand and can work as a new data stream in, the method is suitable for a in-situ feature tracking framework. Also, the parametric distribution representation keeps the storage requirements tractable as the data size scales up. However, with increased block-size, the feature extraction accuracy gets affected since smaller features inside a block cannot be captured with sufficient details. Also, if multiple features exist

inside a block then, the proposed method will detect the block as part of the feature but the separation between them is not possible.

6.7 Conclusion

In the absence of a precise feature definition, the proposed method models the data space and the specified region of interest using mixtures of Gaussians and transforms the data into a feature-aware classified field where high valued regions reflect a higher possibility of the existence of the feature. Such a local region based (block) distribution driven classification allows us to construct a robust tracking algorithm where the tracking is performed in the classification field. In the future, we would like to adapt our method for feature tracking in time-varying ensemble data sets and also to multivariate data sets. Furthermore, we also want to study the effectiveness of our method on data sets with sparsely sampled time steps.

Chapter 7: A Study of Pointwise Information for Time-varying Multi-field Data Exploration

Effective feature exploration in time-varying multivariate data sets is challenging as a thorough understanding of the intricate relationships among multiple variables is involved. Oftentimes, instead of looking at the total correlation among variables, scientists search for specific value combinations of multi-variables which show positive or negative association to get in depth knowledge about the interaction of such variables. So, quantifying the importance of individual value combinations of multiple variables has gained significant importance in the recent years. Multi-field analysis based on their value combinations allows experts to understand how the total shared information among variables is distributed within all of its value combinations. It is to be noted that, a majority of the existing methods have mostly focused on studying the average behavior of the variables, but little focus is on how the specific values of the variables interact with each other. Analysis of importance of scalar values from a single variable system [7, 27, 45, 78, 126], and multi-field domain [15, 90] has been done in the past. But, a guideline to study the relationships of specific value combinations in time-varying multivariate data sets is still missing.

To address the aforementioned issues, an information-theoretic framework is presented in this work to help the scientists to conduct detailed analysis of time-varying multi-fields. The proposed framework efficiently utilizes global data distributions for computing various

information theoretic measures for analysis. We use mutual information (MI) and two of its decomposition (1) specific mutual information (SMI), and (2) pointwise mutual information (PMI) to quantify information of scalar values and value combinations. Time-varying study of multi-fields based on their value combinations using PMI constitutes the core of our framework. We observe the fact that, while decomposed hierarchically in a top-down fashion, MI conveys different facets of information, and by integrating all these information in a unified framework, in depth study of time-varying data sets based on their specific value combinations becomes possible.

The proposed framework enables the experts to select variable combinations based on their shared information content, and further guides them to identify interesting features from the data. We quantify information content of every spatial point by calculating its PMI. Using PMI values at each spatial location, a new scalar field is constructed, called *PMI field*, which are segmented into different regions with value combinations having strong co-occurrences or noticeably low association. High co-occurrence in a region indicates the existence of a joint feature, and regions with low co-occurrence are explored for any potential surprise. We employ this idea in the time domain to study the temporal evolution of such regions. By aggregating several PMI fields from consecutive time steps, we capture feature's temporal evolution. In order to identify temporally salient value combinations, we utilize the temporal information content of the value combinations following a refinement-based strategy. Positive feedback from the domain expert demonstrate the usefulness of our framework in time-varying data exploration.

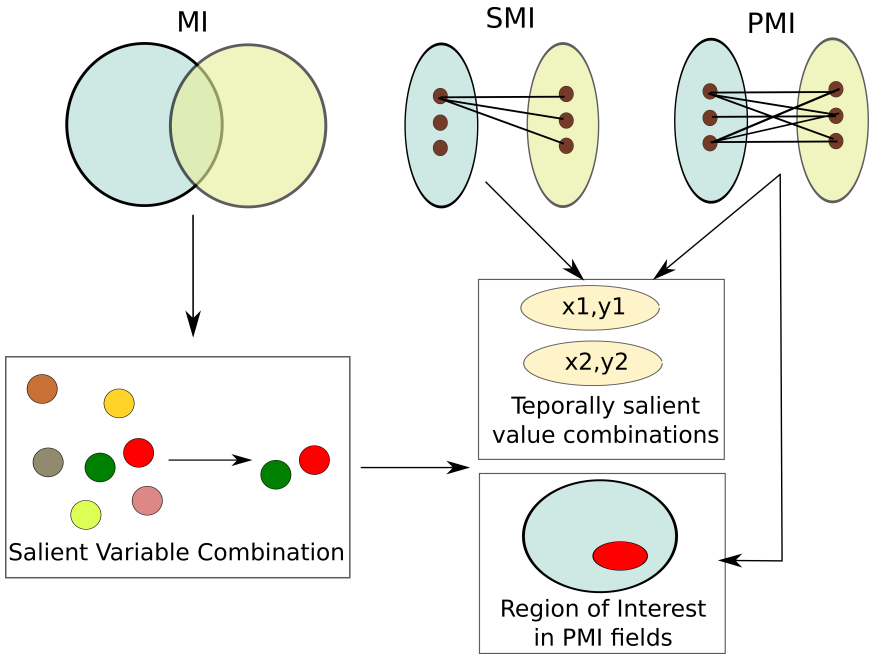


Figure 7.1: A schematic diagram of our information theoretic framework for the analysis of multivariate time-varying data sets.

7.1 Multivariate Temporal Analysis Framework

Given a time-varying multivariate data set, our analysis allows (1) selection of several important variable combinations, and a suitable time range for detailed exploration; (2) efficient detection of regions with joint or complementary multivariate features over time; (3) systematic identification of temporally salient scalar value combinations. Figure 7.1 shows a schematic view of our complete framework, where three hierarchical variations of MI are highlighted and how they are used to analyze multi-variate time-varying data is shown. In the following, we describe our framework in detail.

7.1.1 Defining Variable Interestingness

Typically in a time-varying multivariate domains, not all variables are relevant for analysis. Irrelevant variables can make the discovery of important features difficult. So, it is challenging to determine what variable combinations are salient and how the saliency of relationships evolves over time. Fortunately, domain experts usually have some prior knowledge about the data set and their first goal is to confirm those known facts using existing visualization techniques and then hypothesize new theories. Our variable selection technique exploits information theory to provide guidance for the scientists. When scientists start with any specific variable, we analyze the shared information of this reference variable with the other variables. Since this step is the beginning of our unified workflow and only requires us to quantify the total information shared between all the variables, MI seems a good choice for defining the information overlap between variables. It is defined as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{7.1}$$

where X and Y are two random variables, and $p(x)$, $p(y)$ are the marginal probabilities of observation x of X, and y of Y, and $p(x,y)$ is their joint probability. Since MI considers all the possible values and quantifies the total information overlap between two random variables, we can only conclude about the degree of overlap of information between two random variables. For a time-varying data, the relationships among the variables change over time, therefore, variables can be identified based on their *temporal interestingness*. We quantify this by observing the change of MI over the time for selected variables which depicts how the information overlap of the these variables changes with time. Formally, this can be captured by calculating the temporal gradient of MI for a given variable as:

$$I_t^{'}(X;Y) = \frac{dI(X;Y)}{dt} \tag{7.2}$$

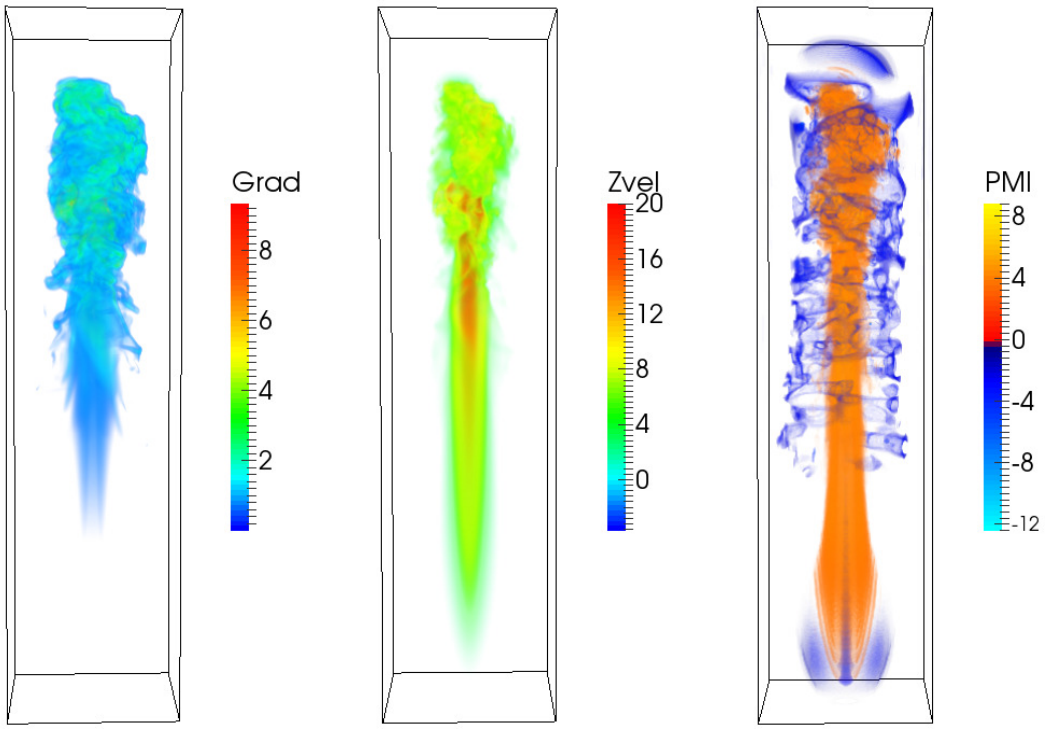
where X and Y are two selected variables and $I_t^{'}(X;Y)$ is the temporal gradient of MI between time steps t and $t+1$.

7.1.2 Combined and Complementary Informativeness Characterization

Given a variable combination, in this work, we demonstrate a workflow that enables users to employ more detailed analysis over a pair of variables by identifying their salient value combinations. Given two variables and a pair of scalar values selected from them, the existence of a strong association between the value pair can be concluded if they demonstrate high co-occurrence. The distribution of these value pairs in the spatial domain can represent a joint multivariate feature. Similarly, when the individual occurrences of the values dominate over their co-occurrence as a pair, the value pair tends to follow a complementary distribution. Here we introduce the information measure that quantifies the shared information for a specific value combination. For two random variables X and Y, if x is an observation of X and y for Y, then the information content between them is expressed as:

$$PMI(x,y) = \log \frac{p(x,y)}{p(x)p(y)} \tag{7.3}$$

where $p(x)$ is the probability of a particular occurrence x of X, $p(y)$ is the probability of y of variable Y and, $p(x,y)$ is their joint probability. This information measure is known as the pointwise mutual information (PMI), which was first introduced in the works of Church and Hanks [35] for the estimation of word association norms directly from computer readable corpora.

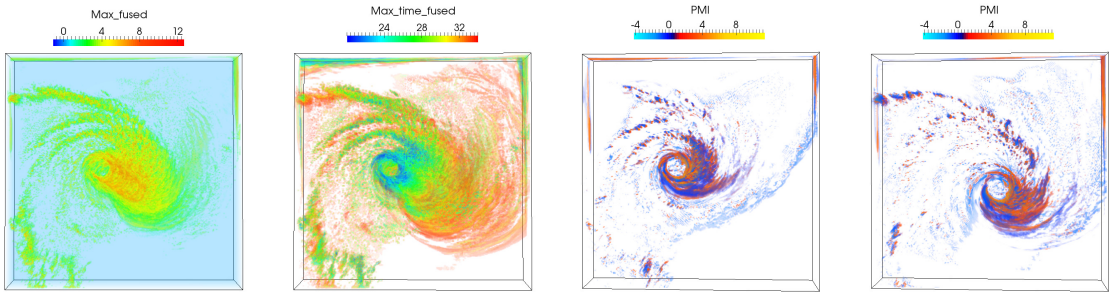


(a) (b) (c)

Figure 7.2: PMI fields of Plume data set. 7.2a shows the velocity gradient magnitude field, 7.2b shows the Zvel field, and 7.2c is the PMI field of these two variables.

- If $p(x,y) > p(x)p(y)$, $PMI(x,y) > 0$, then x and y have higher information sharing between them,
- If $p(x,y) < p(x)p(y)$, then $PMI(x,y) < 0$ indicating the two observations follow complementary distribution,
- When x and y do not have any significant information overlap then $p(x,y) \approx p(x)p(y)$ and $PMI(x,y) \approx 0$. In this case, x and y are considered as statistically independent.

It is to be noted that, mutual information $I(X;Y)$ yields the expected PMI value over all possible instances of variable X and Y [139].



(a) (b) (c) (d)

Figure 7.3: Visualization of PMI fields of Cloud (CLO) and Precipitation (PRE) of Hurricane Isabel data set using time steps between 20-35. 7.3a is the time aggregated PMI field using max function, 7.3b shows the corresponding time volume, 7.3c is the PMI field at T=20, and 7.3d is the PMI field at T=34.

$$I(X;Y) = E_{(X,Y)}[PMI(x,y)] \tag{7.4}$$

The sign and absolute value of PMI enables the categorization of the variable interaction as mentioned above. Using PMI values, both statistically associated and opposite or complementary regions in the data can be identified. The regions that have opposite information will be the unique features in the data which is best represented by one particular variable among the selected variables. Similarly, the regions with strong association highlights joint multivariate features characterized by high co-occurrence.

7.1.3 Multivariate PMI Fields

Given a multivariate time-varying data set, every spatial point in the domain has several scalar values associated with it, one from each variable. Since PMI measures the information content for any value pair, we can use it to obtain the information content for any location. To facilitate this fine grained analysis by preserving the spatial context, we create a new scalar

field, called *PMI field*. In PMI field, the spatial points contain the PMI values computed from the values of the variables at the point. If ζ is the scalar function that maps each spatial point to its PMI value, this multivariate interaction field can be formally expressed as: $\zeta : P \mapsto PMI(P)$, where P is a spatial location and $PMI(P)$ is the PMI value at P. The probabilities for the computation of PMI values are estimated from the histograms of the scalar values obtained from all the grid points.

Figure 7.2 shows one illustrative example of a PMI field where Z-velocity and velocity gradient magnitude field of the Solar Plume data set [116] are used to construct the PMI field. Figure 7.2c depicts the PMI field constructed using these two variables. It is evident from Figure 7.2c that, both Z-velocity and gradient magnitude field have strong statistical association in the turbulent region of the plume, which indicates that the scalar value pairs in this region have higher co-occurrence resulting positive PMI values. However, around the turbulent region, gradient magnitude has unique activity that is missing in the Z-velocity. In the PMI field, this region is considered to contain complementary information which is unique to the gradient magnitude.

7.1.4 Time-varying PMI Fields

Next, we aggregate several PMI fields into a single scalar field using an aggregation function. The aim of this aggregation is to combine information from a set of time steps into a single scalar field for capturing time-varying patterns. For example, if a time-varying feature is identified as a joint activity and if the feature moves spatially over time, then at every time step, the feature can be located by focusing on the regions with higher joint activity. Since positive and high PMI values reflect joint activity, and relatively low co-activity regions have negative PMI values, we use *max* and *min* functions for the aggregation.

If we use max as our aggregation criterion, then at every spatial location, PMI values for all the selected time steps are observed and the maximum value among them is selected to construct a *time aggregated PMI field*. Formally, for a spatial location P, the aggregation value is calculated as:

$$AggPMI(P) = \Psi(PMI_i(P)), \forall\, i = t_s, t_s + 1, .., t_e \tag{7.5}$$

where t_s and t_e represent starting and ending time steps, $PMI_i(P)$ is the value of PMI of point P at time step i, and $\Psi(.)$ is the aggregation function. We also create another scalar field where at every grid point we put the time step number from which the PMI value is selected. We call it the *time volume* which presents the temporal trace of the feature.

In Figure 7.3, we demonstrate the usefulness of this idea with an example where Cloud (CLO) and Precipitation (PRE) variables from Hurricane Isabel data set are used. Given the range of time steps between $20 - 35$, we construct the time aggregated PMI field using max aggregation function. Figure 7.3a shows the aggregated PMI field and 7.3b shows the associated time volume. From Figure 7.3b we can visualize how the feature has moved by looking at the change of color in Figure 7.3b. Note that, in Figure 7.3b the color signifies time steps and it varies from blue to red as time increases.

To allow analysis of the identified feature at specific time steps, we incorporate a threshold based visualization. Initially users select a threshold for both high and low PMI values which highlight their regions of interest at the first selected time step. Then we apply this threshold to all the other time steps to extract the regions that show either strong or weak statistical association. For maintaining consistency, before the threshold is applied, all the PMI fields from the selected time range are normalized so that the PMI values are scaled consistently over the selected time window. Figure 7.3c and 7.3d show the results of

the thresholding where snapshots of time steps 20 and 34 are displayed respectively. The *continuation* of the downward rotational movement of the cloud structures and precipitation bands are visible from these images.

7.1.5 Identification of Temporally Salient Scalar Value Combinations

Acknowledging the fact that the total number of value combinations can be significantly large, we present a refinement based strategy which aims at grouping value combinations with similar behavior. The proposed method exploits both SMI and the PMI to devise a top-down approach. In our work, SMI measure *predictability* is used which was introduced in [43]. Formally, given a value x of the variable X, its predictability is defined as:

$$\begin{aligned} SMI(x;Y) &= H(Y) - H(Y|x) \\ &= -\sum_{y \in Y} p(y) \log p(y) + \sum_{y \in Y} p(y|x) \log p(y|x) \end{aligned} \tag{7.6}$$

where Y is the other selected variable, $H(Y)$ is the entropy of Y, and $H(Y|x)$ is the entropy of Y given observation x. $SMI(x;Y)$ is called predictability because based on the value of $SMI(x;Y)$, it can be inferred, how well the observation x can predict the behavior of Y. Higher and positive values of $SMI(x;Y)$ reflects higher predictability, whereas, negative values of $SMI(x;Y)$ signifies increased uncertainty about Y after x is observed. So, using $SMI(x;Y)$, the scalar values of X can be divided into two groups: (1) scalars with positive $SMI(x;Y)$ i.e. the predictable scalars, and (2) scalars with negative $SMI(x;Y)$ containing the uncertain scalars of variable X.

After this initial grouping using PMI, the value combinations are further classified into two groups: (1) combinations with positive PMI values, and (2) combinations that have negative PMI values. Here we are focusing on the combined and complementary features of

multi-variables, so, we do not consider the combinations with PMI value 0. Since our goal is to quantify the temporal trends of the value combinations, we observe the PMI value of each value combination for all the selected time steps and group them separately if the value combinations have always positive or always negative values throughout the specified time range.

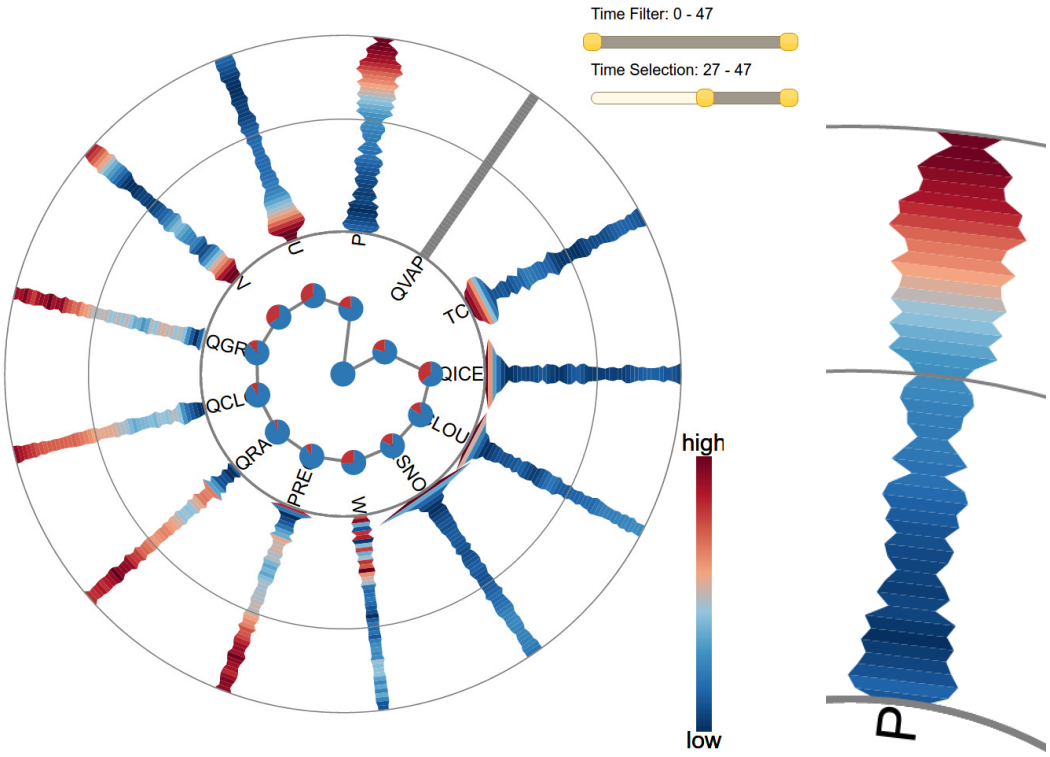
Based on the above discussion, the value combinations of variable X and Y are grouped into 4 distinct classes:

1. $\{(x_i, y_j) \mid \forall i, j \; where \; PMI(x_i, y_j) > 0 \; \& \; SMI(x_i; Y) < 0\}$
2. $\{(x_i, y_j) \mid \forall i, j \; where \; PMI(x_i, y_j) < 0 \; \& \; SMI(x_i; Y) < 0\}$
3. $\{(x_i, y_j) \mid \forall i, j \; where \; PMI(x_i, y_j) > 0 \; \& \; SMI(x_i; Y) > 0\}$
4. $\{(x_i, y_j) \mid \forall i, j \; where \; PMI(x_i, y_j) < 0 \; \& \; SMI(x_i; Y) > 0\}$

Given any class, for each value combination in it, we construct a time series using its PMI values and its temporal saliency is measured by the variation of the PMI values. Formally, the variation for a value combination is measured as:

$$Var(TS_i) = \sqrt{\sum_{j=t1}^{t2-1} |PMI_{i,j} - PMI_{i,j+1}|^2} \tag{7.7}$$

where, TS_i is the time series of ith value combination. $PMI_{i,j}$ is the PMI value of series TS_i at jth time step and the selected time step range is $t1 - t2$. A high variation value indicates that the value combination has weaker association among them and their occurrences are not consistent temporally. In contrast, the time series with low variation are likely to reveal a region that has higher statistical association. With this classification strategy, the complexity in the relationships among the large number of value combinations is reduced significantly.



(a) (b)

Figure 7.4: 7.4a Variable selection for Isabel data set when Qvapor (QVA) variable is selected. 7.4b Zoomed in Pressure axis.

7.2 Interactive Workflow

7.2.1 Identifying Temporally Related Variables

Interface Design. As domain scientists often have some prior knowledge about which variable is more interesting to initiate the exploration process, we are interested in seeing the relationships between the reference (initially picked by the user) variable and the other variables. Figure 7.4a shows our variable selection interface where Hurricane Isabel data set is used for demonstration. The reference variable (QVapor) is selected by the user and placed in the center, while the other variables are placed such that one is closer to the center

if it is strongly related to the reference variable. The layout has several benefits: (1) it emphasizes the stronger relationships as the central part of the visualization, which is more active in the user's visual field; (2) it preserves the user's mental map. The time-varying one-to-many relationships form a time series for each variable pair. We visualize these in multiple time plots, which are aligned with variables in the radar plot. Such a design is usually more efficient than shared space techniques [70] and results in an overview + detail visualization.

Visual Encoding. While the location of a variable in the radar plot shows the strength of the corresponding relationship, each variable itself is visualized as a pie chart to depict the percentage of the positive (blue) and negative (red) PMI values for each variable pair. The segments in a time plot in Figure 7.4a is colored by the strength of the relationship, mutual information in this case, while the width of each segment is modulated by the temporal gradient of the relationship's strength. In Figure 7.4a we also show the selection of time steps where the gray circles show the time range selected.

Guidelines for Variable Selection. With our design, once the user has selected a reference variable (QVA in this case), they can select another variable and an appropriate time window based on how the information is shared between it and the reference variable: *(1) Varying information overlap:* a variable showing a rapid change of colors with a wide time axis. (Selected time steps of P in Figure 7.4a, highlighted by the two gray circular rings), *(2) Constantly high information overlap:* a variable that has mostly red regions for a sequence of time steps. (Later time steps of PRE, QRA, QCL etc. in Figure 7.4a), *(3) Low information overlap with high variation:* a variable containing blue regions with a relatively wide time axis (later time steps of U in Figure 7.4a), *(4) Constantly low information overlap:*

a variable with mostly blue regions with a narrow time axis. (Majority parts of the time axis of QSN, QIC, CLO etc. in Figure 7.4a).

7.2.2 Analysis using PMI Fields

After a pair of variables are identified, we construct PMI fields for each selected time step using those variables and aggregate them using both max and min functions. This PMI field based visualizations allow scientists to directly interact with the information in spatial domain. After the informative regions and temporal trends are analyzed using the aggregated volumes, we allow users to interact with the specific value combinations so that the scalar values creating such joint temporal features can be specifically identified.

7.2.3 Identification of Temporally Salient Scalar Value Combinations.

In section 7.1.5 we have described how the value combinations can be grouped based on their informativeness. Next, we create a histogram of all the value combinations in each group using their PMI variation values. We allow brushing in the *variation histogram* so that users can select bins with high or low information variation. A parallel coordinates plot (PCP) is attached with the histogram, so that the selected value combinations can be easily visualized. Finally, users can brush the PCP to select specific value combinations and while visualizing isosurfaces of those selected value combinations, their PMI time series are also displayed. Users can change the time steps to inspect the temporal changes of such isosurfaces and also observe how their PMI values change. Figure 7.7a shows a variation histogram where the analysis is done using QVA and P variables of the Isabel data and the value combinations from the group 1 (described in Section 7.1.5) is chosen. The light yellow highlighted region shows the user selected bins and in Figure 7.7b and Figure 7.7c the corresponding PCP and the PMI time series are shown.

7.3 Case Studies

The experiments were done on a Linux machine with an Intel core i7-2600 CPU, 16 GB of RAM and an NVIDIA Geforce GTX 660 GPU with 2GB texture memory. The visualizations were generated using D3 library [18] and ParaView [6].

7.3.1 Hurricane Isabel Data Set

Hurricane Isabel data is a multivariate time-varying data consisting of 13 scalar fields. The data set is a courtesy of NCAR and the U.S. National Science Foundation (NSF), and was created using the Weather Research and Forecast (WRF) model. The resolution of the grid is $250 \times 250 \times 50$, and there are total 48 time steps. From Figure 7.4a, we see that Qvapor (QVA) is selected as the reference variable for this study. Given QVA, following our variable selection interface, Pressure (P) is selected as the second variable since it shows varying information overlap between time steps $27 - 47$.

Figure 7.5a and 7.5b show the aggregated PMI field and its time volume when max is used for aggregation. We see that the Hurricane eye has strong co-activity. Similarly, Figure 7.5c and 7.5d depict the aggregated PMI field and its time volume when min is used for aggregation. The eye wall of the storm is visible as a complementary feature whose temporal trend is observed from the associated time volume in Figure 7.5d. To facilitate exploration of identified regions at specific time steps, Figure 7.6 presents 3 selected PMI fields where we can visualize the combined and complementary regions at three individual time steps. Figure 7.6a, 7.6b, and 7.6c depict the temporal changes of the regions of interest at time steps 30, 40, and 45 respectively where reddish yellow regions signify regions with stronger co-activity and the light blue region which shows the eye wall of the storm is identified as the complementary informative region.

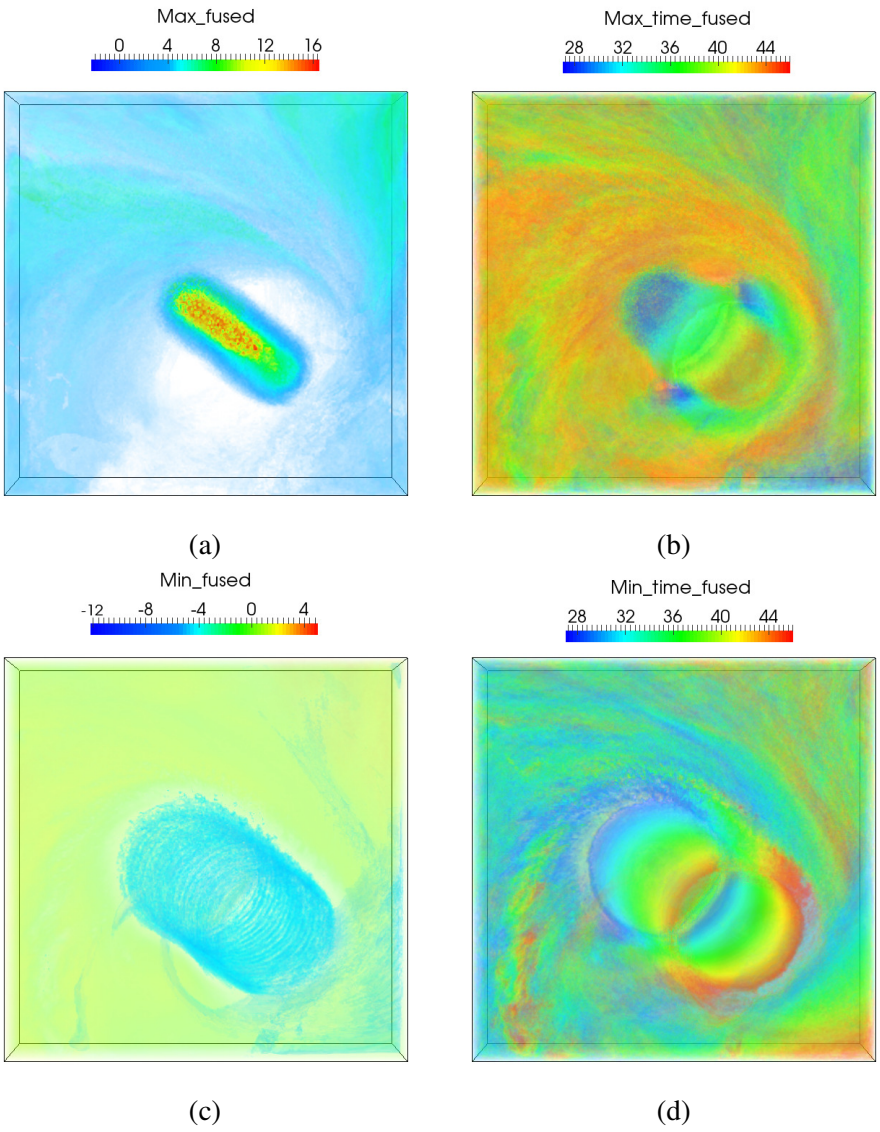
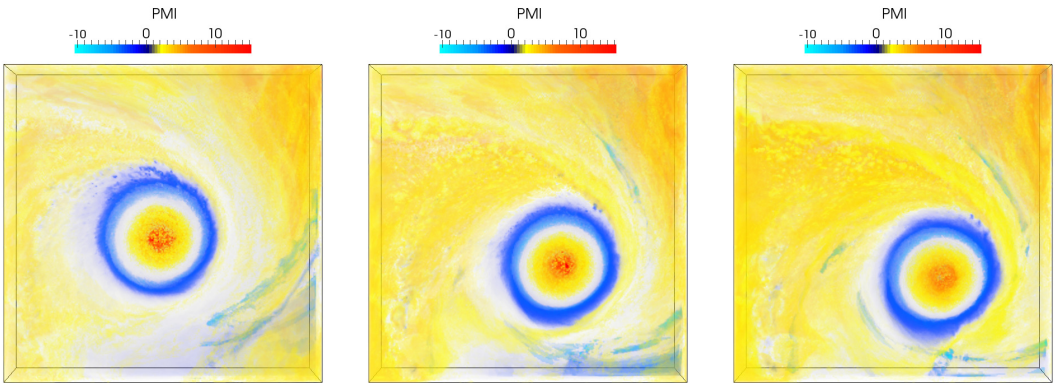


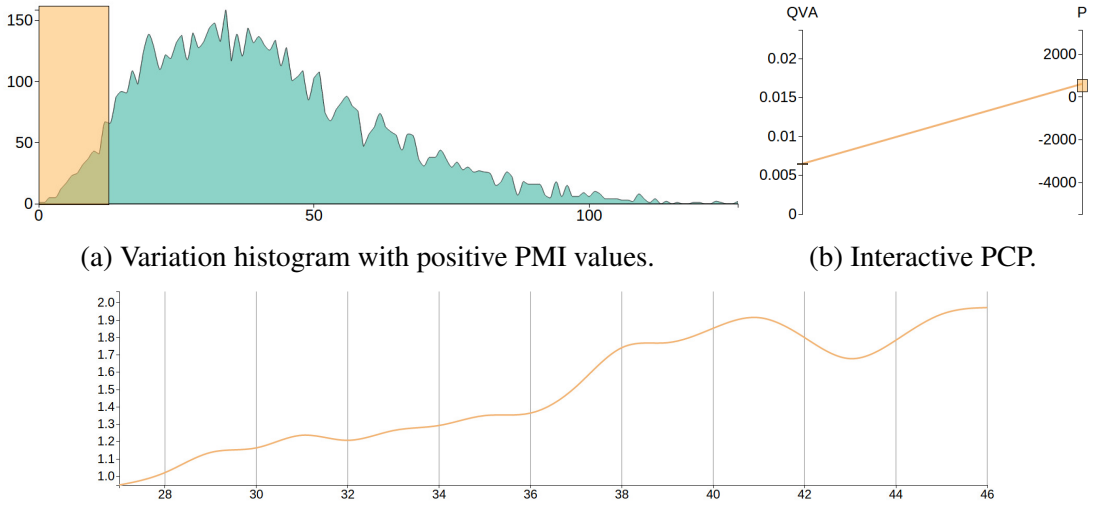
Figure 7.5: Aggregated PMI fields of P and QVA of Isabel data between time steps 27-47. 7.5a Aggregated PMI field using max function and 7.5b its Time volume; 7.5c Aggregated PMI field using min function, and 7.5d its Time volume.

Figure 7.7a shows the variation histogram of QVA and P representing the value combinations when the group with all positive PMI and negative SMI values of QVA are considered. Brushing some low variation bins (yellow highlighted region) yields the PCP in Figure



(a) PMI field at T=30. (b) PMI field at T=40. (c) PMI field at T=45.

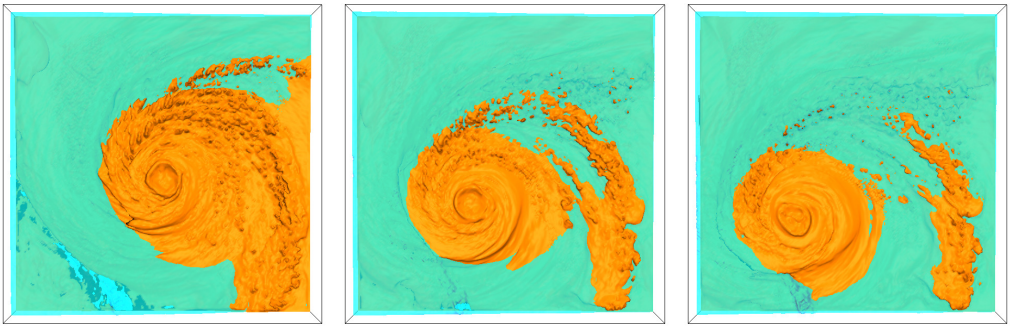
Figure 7.6: Time-varying PMI fields of Pressure (P) and Qvapor (QVA) of Hurricane Isabel data set between time steps 27-47.



(a) Variation histogram with positive PMI values. (b) Interactive PCP.

(c) PMI time series of selected value combinations.

Figure 7.7: Selection of salient scalar value combinations of Pressure (P) and Qvapor (QVA) variables of Hurricane Isabel data set between time steps 27- 47. Selected value combinations reflect combined activity of the selected variables.



(a) T=30. (b) T=40. (c) T=45.

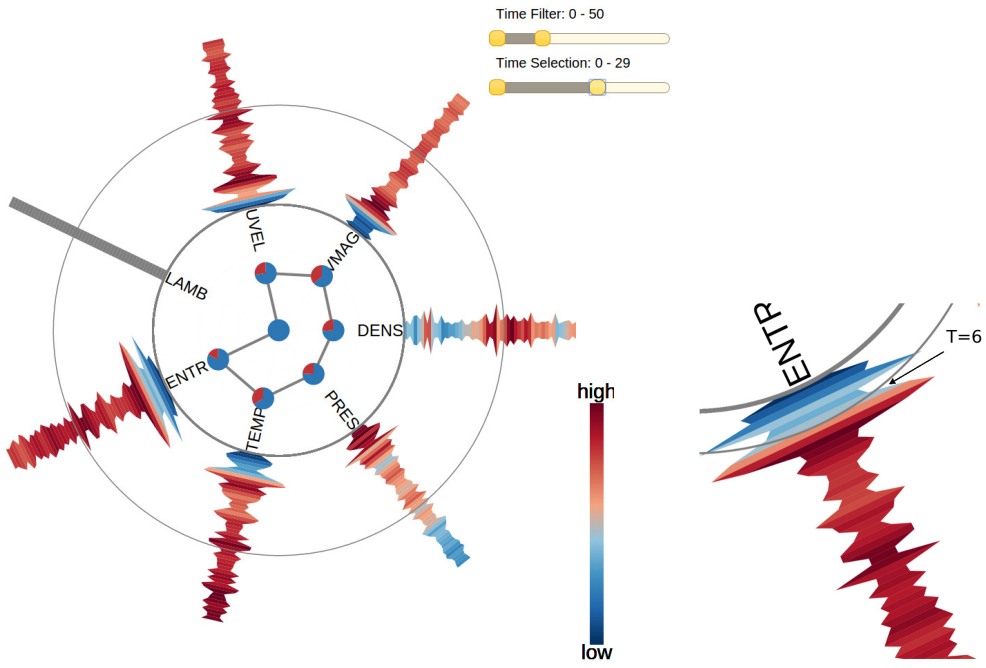
Figure 7.8: Temporally salient isosurface visualization of Pressure (P) = 617.04 (blue) and Qvapor (QVA) = 0.00647598 (orange) of Hurricane Isabel data set.

7.7b, from which a specific value combination is selected. In Figure 7.7c the PMI series is depicted. The trend shows that the magnitude of the PMI values increase over time which is reflected in the isosurfaces in Figure 7.8. In Figure 7.8a, 7.8b, and 7.8c isosurfaces of P = 617.04 (blue) and QVA = 0.00647598 (orange) are shown for time steps 30, 40 and, 45 respectively. We observe that, as the PMI values increase, the degree of the association between the value pair also strengthens which is revealed by their increased overlap.

7.3.2 Turbine Data Set

The Turbine data set is generated by a flow simulation TURBO, where the compressor is undergoing rotating stall [32]. It is a multi-block data consisting of 36 blade passages. The resolution of each passage/block is $151 \times 71 \times 56$ and has five variables: Density (DENS), Velocity momentum in x/y/z direction (MOM), and Total energy (TOTENR). These five variables are used to derive other variables. The simulation was run for 192 time steps.

In a stable state, the tip region of each blade develops a vortex known as *tip vortex*. Detailed study of this tip vortex during stall inception and identification of the associated



(a) (b)

Figure 7.9: 7.9a Variable selection for Turbine data set when $\lambda 2$ (LAMB) variable is selected, 7.9b Zoomed in ENTR axis.

variable interactions are essential for the scientist. In order to facilitate domain experts with a better understanding on the behavior of the tip vortex, we have computed vortex criterion $\lambda 2$ (LAMB). In Figure 7.9a we show our variable selection interface when $\lambda 2$ (LAMB) is selected as the reference variable by the expert. It is observed that variable entropy (ENTR) displays the highest information variation during the initial time steps and at time step 6 the shared information between these two variables become high which can be seen by the change of color of ENTR axis from initial blue to red in Figure 7.9b. To investigate this, ENTR becomes a suitable second variable for analysis and time steps between $1-29$ is selected for detailed investigation. For highlighting this region, in Figure 7.9a we only show time steps between $0-50$ using the time step filter slider.

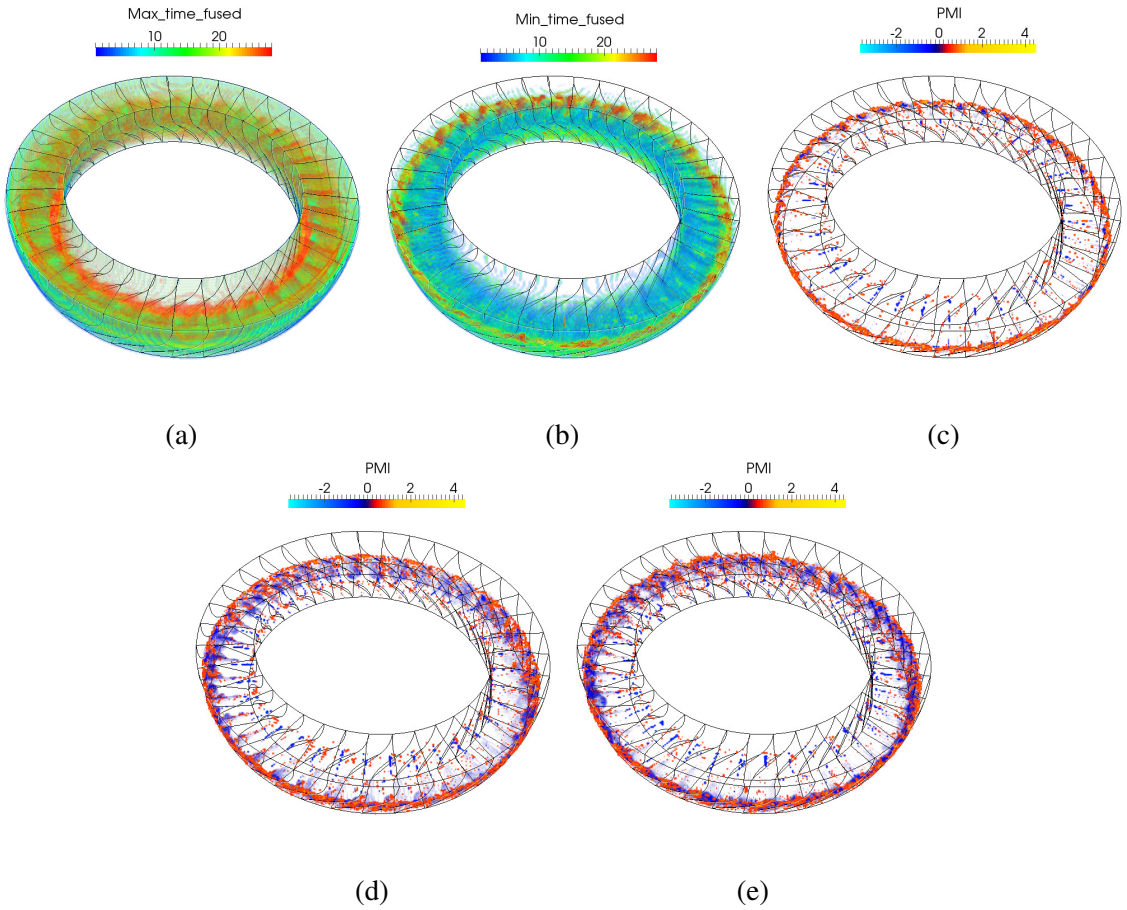
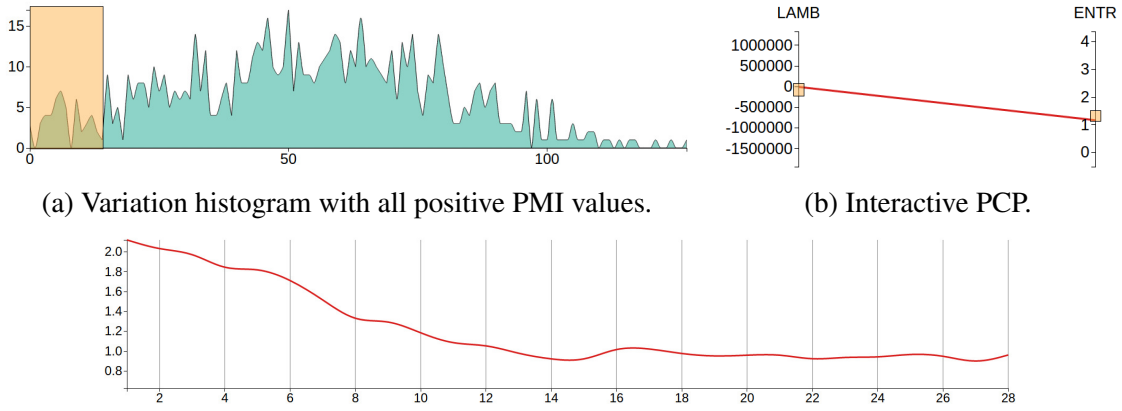


Figure 7.10: Visualization of PMI fields of Turbine data when variables $\lambda 2$ (LAMB) and Entropy (ENTR) are selected between time steps 1 - 29. 7.10a Time volume with max function, 7.10b Time volume with min function, 7.10c PMI field at T=5, 7.10d PMI field at T=15, and 7.10e PMI field at T=25

Figure 7.10a and 7.10b show the max and min time volumes of the selected variables. In Figure 7.10a, we observe that the the regions away from the tip show stronger statistical association during the later time steps, hence more red regions are located away from the tip region. The min time volume displayed in Figure 7.10b, show the opposite trend. Here the tip regions show more opposite activity during later time steps identified by the reddish yellow regions. Figure 7.10c, 7.10d, and 7.10e present the PMI fields of three selected time



(a) Variation histogram with all positive PMI values.

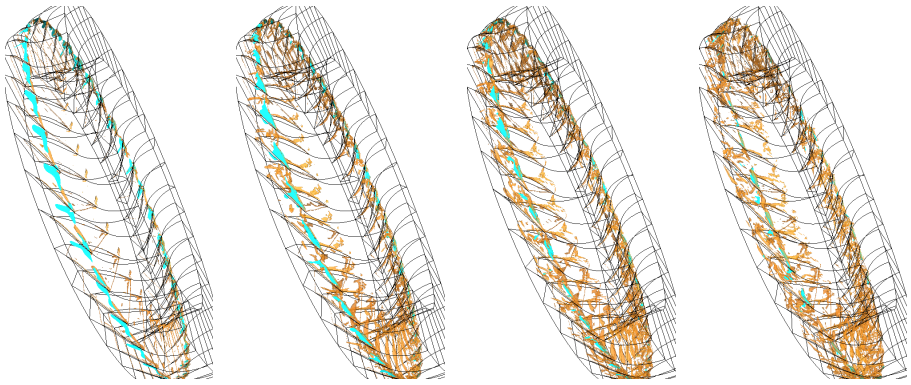
(b) Interactive PCP.

(c) PMI time series of selected value combinations.

Figure 7.11: Selection of salient scalar value combinations of $\lambda 2$ (LAMB) and Entropy (ENTR) variables of Turbine data set between time steps 1-29. Selected value combinations reflect combined activity of the selected variables.

steps, 5, 15 and 25 respectively. We can see that as time progresses, more regions along the tip with a complementary activity appear. In this case study, we see that the complementary region of interest grows over time which signifies the increase in opposite information between these two selected variables.

Figure 7.11a depicts the variation histogram with all positive PMI and negative SMI values for LAMB. Figure 7.11b shows when a specific value combination is picked by brushing the histogram first, and then filtering from the PCP. Figure 7.12 presents simultaneous isosurfaces of the picked value combination (LAMB = -5784.25 (orange) and ENTR = 1.16768 (blue)) for time steps 1, 6, 7, and 15 respectively. Note that, the isosurface of LAMB we visualize here shows vortices (tip vortex in this case). We find that from time step 6 on-wards, the LAMB isosurface becomes fragmented which represents the *breakdown of the tip vortices*. This is an indication of the stall inception. In the variable selection interface (Figure 7.9a), we observe that time steps $5-7$ cause sudden change of MI between these



(a) T=1. (b) T=6. (c) T=7. (d) T=15.

Figure 7.12: Temporally salient isosurface visualization of variables $\lambda 2$ (LAMB) = -5784.25 (orange) and Entropy (ENTR) = 1.167 (blue) of Turbine data set.

two variables when the tip vortices breakdown. Also, after the tip vortices break down, the degree of co-occurrence of this value combination is reduced which can be seen from Figure 7.11c, where the magnitude of PMI value gradually decreases.

The domain expert who evaluated our system has more than 25 years of experience in the computational fluid dynamics simulations and is one of the developers of the TURBO simulation code which we used for this study. The feedback were collected through several meetings with the expert during which we explained our system to the expert in detail. According to the expert, identification of the breakdown of tip vortices was helpful for a detailed study of stall. Comparing our tools with the existing tools such as *FieldView*, the expert pointed that, our tool was able to provide the information-theoretic guidance when little knowledge was available. Also, the expert confirmed that the PMI field based identification of salient regions allowed to locate interesting regions where the variables show strong or weak statistical association. The expert also mentioned that the time series based analysis of value combinations provided a new tool to perform detailed multivariate

Table 7.1: Timings for computing PMI fields and aggregation of PMI fields.

Data Set	Avg. time per PMI field creation (secs.)	Avg. time for aggregation (secs.)
Hurricane Isabel	0.461	0.210
Turbine	3.817	0.325

temporal study on the scalar values. This helped in identifying salient value combinations which were either strongly correlated or showed weak association.

The average computation time for the PMI field creation and their aggregation is shown in Table 7.1. The computation of all pair MI for our variable selection interface was done as a pre-processing step. Note that, since creation of PMI fields are independent for each time step, they can be generated in parallel which will further improve the performance.

7.4 Discussion and Conclusion

In this work, we present an information theoretic approach for exploration of multivariate time-varying data sets. The information theoretic measures are computed from the global data distributions which are much smaller in size compared to the actual raw data and as a result the analysis becomes much more tractable. We use pointwise mutual information (PMI) to measure the information content for specific value combinations of multiple variables and further use such information to construct PMI fields. The PMI field allows us to analyze variable relationships using the pointwise mutual information keeping the spatial perspective intact. For identification of time-varying features, we extend the PMI field into temporal domain by aggregating several PMI fields using various aggregation criteria. To identify salient value combinations, we measure temporal information variation of each

value combination and group the value combinations using their variation values. In the future, we wish to use our framework for different data types such as vector and ensemble data.

Chapter 8: Conclusion and Future Works

8.1 Conclusion

In this dissertation, we have demonstrated the prospects of different types of *in situ* data summarization schemes and their effectiveness in analyzing and visualizing extreme-scale scientific data sets. To tackle the big data avalanche, in Chapter 3 we present an off-line learning and *in situ* feature prediction strategy which produces feature-driven data summaries that can be easily compared in the post-hoc analysis phase in a timely manner with minimal effort. Furthermore, when analysis using a predetermined feature-driven summarization is not sufficient, we have proposed several statistical data summarization schemes in Chapter 4 which are more general purpose and perform *in situ* distribution-guided data summarization and output compact and reduced distribution data for flexible post-hoc analysis. Our studies presented in Chapters 5 and 6 show that, when the size of the raw data in sufficiently large, local region-wise distribution-based data summaries can be used as a replacement of the raw data, and various types of tasks such as: (a) flow instability detection in extreme-scale CFD data sets; (b) feature extraction and tracking in large time-varying data; (c) flexible post-processing analysis with hypotheses generation and verification etc. can be done efficiently. We also highlight that our distribution-based data summaries have the capability of uncertainty analysis during exploration which allows

the scientists to make more accurate judgments from the results. To show the impact of our proposed schemes in real life applications, we have applied our proposed techniques to a high-resolution computational fluid dynamics simulation verifying the efficacy in solving domain specific problems. Furthermore, extensive *in situ* performance study has been done to validate that the proposed data summarization schemes are well suited and practical for the *in situ* environment and do not overburden the simulation run. Besides the local region-wise data summaries, we have also exploited the global distribution-based data models in Chapter 7 for facilitating multivariate time-varying data analysis. We have used various information theoretic measures computed from the global data distributions for enabling systematic exploration of large time-varying data sets. By quantifying information content of the scalar value combinations of multi-variables, we enable a low-level multivariate data exploration, which can enhance the scientists understanding about multivariate interaction greatly.

We believe that, in the era of big data analytics, our compact and informative statistical data summarization techniques can serve as a practical pathway for interactive data analytics and visualization for scientific data sets. The proposed analysis techniques also open up several possible future directions of research which we discuss in the following.

8.2 Future Research Directions

In this section, we discuss some potential future avenues of research which will extend our work to a wider application domain, as well as, address several new data visualization problems. Our proposed local region-based data summarization scheme uses each variable separately while creating distribution-based representations. With this summary data, accurate multivariate data analysis is not possible since, by summarizing each variable individually, we do not explicitly preserve their correlation information. Multivariate data

analysis is essential in many scientific applications as the study of relationships among different variable combinations can facilitate enhanced understanding of the data. To achieve this, we can extend our technique by simply estimating multivariate data distributions instead of univariate distributions which will capture the variable correlations, however, as the number of variables increase, the memory footprint of the multivariate distributions will be sufficiently large compared to their univariate representations. Also, estimation of multivariate distributions in the form of GMMs will take significantly longer time and may not be readily computed in the *in situ* environment as it may overburden the simulation. Therefore, new scalable and *in situ* friendly solutions are needed to be developed in the future to perform multivariate extreme-scale data summarization and analysis. Also, we have demonstrated the usefulness of pattern learning algorithms for generating feature-driven summarizations in the *in situ* environment which can accelerate the post-hoc visual analysis significantly. Since features in the scientific simulations are becoming increasingly intricate, more sophisticated predictive feature detection algorithms are required which will strengthen the *in situ* feature detection capabilities essential for performing extreme-scale visual analytics. Future endeavors in this direction can exploit various state-of-the-art machine learning algorithms for designing more efficient, accurate, and robust *in situ* feature exploration techniques.

Bibliography

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.

[2] V. Adhinarayanan, W. C. Feng, J. Woodring, D. Rogers, and J. Ahrens. On the greenness of in-situ and post-processing visualization pipelines. In *2015 IEEE International Parallel and Distributed Processing Symposium Workshop*, pages 880–887, May 2015.

[3] J. Ahrens, S. Jourdain, P. OLeary, J. Patchett, D. H. Rogers, and M. Petersen. An image-based approach to extreme scale in situ visualization and analysis. In *SC14: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 424–434, 2014.

[4] Hiroshi Akiba, Nathaniel Fout, and Kwan-Liu Ma. Simultaneous classification of time-varying volume data based on the time histogram. In *Proceedings of the Eighth Joint Eurographics / IEEE VGTC conference on Visualization*, pages 171–178, 2006.

[5] T. Athawale, E. Sakhaee, and A. Entezari. Isosurface visualization of data with nonparametric models for uncertainty. *IEEE Trans. on Vis. and Comp. Graphics*, 22(1):777–786, 2016.

[6] Utkarsh Ayachit. *The ParaView Guide: A Parallel Visualization Application*. Kitware Inc., 4.3 edition, 2015. ISBN 978-1-930934-30-6.

[7] C.L. Bajaj, V. Pascucci, and D.R. Schikore. The contour spectrum. In *Visualization '97., Proceedings*, pages 167–173, Oct 1997.

[8] Arunava Banerjee, Haym Hirsh, and Thomas Ellman. Inductive learning of feature-tracking rules for scientific visualization. In *Workshop on Machine Learning in Engineering (IJCAI-95*, 1995.

[9] Andrew C. Bauer, Hasan Abbasi, James Ahrens, Hank Childs, Berk Geveci, Scott Klasky, Kenneth Moreland, Patrick O'Leary, Venkatram Vishwanath, Brad Whitlock, and E. W. Bethel. In Situ Methods, Infrastructures, and Applications on High Performance Computing Platforms. *Computer Graphics Forum*, 2016.

[10] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum*, 30(3):911–920, 2011.

[11] J. Bernard, M. Hutter, M. Zeppelzauer, D. Fellner, and M. Sedlmair. Comparing visual-interactive labeling with active learning: An experimental study. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):298–308, Jan 2018.

[12] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[13] Jeff Bilmes. A gentle tutorial on the em algorithm including gaussian mixtures and baum-welch. Technical report, International Computer Science Institute, 1997.

[14] A. Biswas, D. Thompson, Wenbin He, Q. Deng, Chun-Ming Chen, Han-Wei Shen, R. Machiraju, and A. Rangarajan. An uncertainty-driven approach to vortex analysis using oracle consensus and spatial proximity. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pages 223–230, April 2015.

[15] Ayan Biswas, Soumya Dutta, Han-Wei Shen, and Jonathan Woodring. An information-aware framework for exploring multivariate data sets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2683–2692, 2013.

[16] Georges-Pierre Bonneau, Hans-Christian Hege, Chris R. Johnson, Manuel M. Oliveira, Kristin Potter, Penny Rheingans, and Thomas Schultz. *Scientific Visualization: Uncertainty, Multifield, Biomedical, and Scalable Visualization*, chapter Overview and State-of-the-Art of Uncertainty Visualization, pages 3–27. Springer London, 2014.

[17] U.D. Bordoloi and Han-Wei Shen. View selection for volume rendering. In *Visualization, 2005. VIS 05. IEEE*, pages 487–494, 2005.

[18] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, December 2011.

[19] R. Bramon, I. Boada, A. Bardera, J. Rodriguez, M. Feixas, J. Puig, and M. Sbert. Multimodal data fusion based on mutual information. *Visualization and Computer Graphics, IEEE Transactions on*, 18(9):1574 –1587, sept. 2012.

[20] R. Bramon, M. Ruiz, A. Bardera, I. Boada, M. Feixas, and M. Sbert. Information theory-based automatic multimodal transfer function design. *Biomedical and Health Informatics, IEEE Journal of*, 17(4):870–880, July 2013.

[21] Roger Bramon, Marc Ruiz, Anton Bardera, Imma Boada, Miquel Feixas, and Mateu Sbert. An information-theoretic observation channel for volume visualization. *Comput. Graph. Forum*, 32(3):411–420, 2013.

[22] Roger Bramon, Marc Ruiz, Anton Bardera, Imma Boada, Miquel Feixas, and Mateu Sbert. An information-theoretic observation channel for volume visualization. *Comput. Graph. Forum*, 32(3):411–420, 2013.

[23] Ken Brodlie, Rodolfo Allendes Osorio, and Adriano Lopes. *Expanding the Frontiers of Visual Analytics and Visualization*, chapter A Review of Uncertainty in Data Visualization, pages 81–109. Springer London, 2012.

[24] Stefan Bruckner and Torsten Möller. Isosurface similarity maps. *Computer Graphics Forum*, 29:773–782, 2010.

[25] Jesus Caban, Alark Joshi, and Penny Rheingans. Texture-based feature tracking for effective time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1472–1479, November 2007.

[26] S. Camarri, M.-V. Salvetti, M. Buffoni, and A. Iollo. Simulation of the three-dimensional flow around a square cylinder between parallel walls at moderate Reynolds numbers. In *XVII Congresso di Meccanica Teorica ed Applicata*, 2005.

[27] Hamish Carr, Jack Snoeyink, and Ulrike Axen. Computing contour trees in all dimensions. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '00, pages 918–926, Philadelphia, PA, USA, 2000. Society for Industrial and Applied Mathematics.

[28] Ohio Supercomputer Center. Oakley supercomputer. `http://osc.edu/ark:/19495/hpc0cvqn`, 2012.

[29] A. Chaudhuri, T. H. Wei, T. Y. Lee, H. W. Shen, and T. Peterka. Efficient range distribution query for visualizing scientific data. In *2014 IEEE Pacific Visualization Symposium*, pages 201–208, March 2014.

[30] Chun-Ming Chen, S. Dutta, Xiaotong Liu, G. Heinlein, Han-Wei Shen, and Jen-Ping Chen. Visualization and analysis of rotating stall for transonic jet engine simulation. *IEEE Trans. on Vis. and Comp. Graphics*, 22(1):847–856, 2016.

[31] J. Chen, D. Silver, and L. Jiang. The feature tree: visualizing feature tracking in distributed amr datasets. In *Parallel and Large-Data Visualization and Graphics, 2003. PVG 2003. IEEE Symposium on*, pages 103–110, Oct 2003.

[32] Jen-Ping Chen, Michael D. Hathaway, and Gregory P. Herrick. Prestall behavior of a transonic axial compressor stage via time-accurate numerical simulation. *Journal of Turbomachinery*, 130(4):041014, 2008.

[33] Jen-Ping Chen, Robert Webster, Michael Hathaway, Gregory Herrick, and Gary Skoch. Numerical simulation of stall and stall control in axial and radial compressors. In *44th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, 2006.

[34] Hank Childs. Data exploration at the exascale. *Supercomputing frontiers and innovations*, 2(3), 2015.

[35] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th annual meeting on Association for Computational Linguistics*, ACL '89, pages 76–83, Stroudsburg, PA, USA, 1989. Association for Computational Linguistics.

[36] Robert T. Clemen and Robert L. Winkler. Combining probability distributions from experts in risk analysis. *Risk Analysis*, 19(2):187–203, 1999.

[37] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal. Automated multi-modality image registration based on information theory. In *Information Processing in Medical Imaging*, 1995.

[38] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[39] R.A. Crawfis and N. Max. Texture splats for 3d scalar and vector field visualization. In *Visualization, 1993. Visualization '93, Proceedings., IEEE Conference on*, pages 261–266, Oct 1993.

[40] Ralph B. D'agostino, Albert Belanger, and Ralph B. D'agostino Jr. A suggestion for using powerful and informative tests of normality. *The American Statistician*, 44(4):316–321, 1990.

[41] I. J. Day, T. Breuer, J. Escuret, M. AU - Cherrett, and A. AU - Wilson. Stall inception and the prospects for active control in four high-speed compressors. *Journal of Turbomachinery*, 121(1):18–27, 1999.

[42] M.C.F. de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: a survey. *Visualization and Computer Graphics, IEEE Transactions on*, 9(3):378–394, 2003.

[43] Michael R. DeWeese and Markus Meister. How to measure the information gained from one symbol. *Network: Computation in Neural Systems*, pages 325–340, nov 1999.

[44] Werner Dubitzky, Martin Granzow, and Daniel Berrar. *Fundamentals of Data Mining in Genomics and Proteomics*. Springer US, 2007.

[45] B. Duffy, H. Carr, and T. Moller. Integrating isosurface statistics and histograms. *Visualization and Computer Graphics, IEEE Transactions on*, 19(2):263–277, Feb 2013.

[46] N. Fabian, K. Moreland, D. Thompson, A. C. Bauer, P. Marion, B. Gevecik, M. Rasquin, and K. E. Jansen. The paraview coprocessing library: A scalable, general purpose in situ visualization library. In *2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, 2011pages = 89-96, doi = 10.1109/LDAV.2011.6092322,.

[47] Miquel Feixas, Esteve Del Acebo, Philippe Bekaert, and Mateu Sbert. An Information Theory Framework for the Analysis of Scene Complexity. *Computer Graphics Forum*, 1999.

[48] Miquel Feixas, Mateu Sbert, and Francisco González. A unified information-theoretic framework for viewpoint selection and mesh saliency, 2006.

[49] Nathaniel Fout and Kwan-Liu Ma. Fuzzy volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2335–2344, 2012.

[50] R. Fuchs, J. Waser, and M. E. Groller. Visual human+machine learning. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1327–1334, Nov 2009.

[51] C. Garth, X. Tricoche, and G. Scheuermann. Tracking of vector field singularities in unstructured 3d time-dependent datasets. In *Visualization, 2004. IEEE*, pages 329–336, Oct 2004.

[52] L.J. Gosink, C. Garth, J.C. Anderson, E.W. Bethel, and K.I. Joy. An application of multivariate statistical analysis for query-driven visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 17(3):264–275, March 2011.

[53] Yi Gu and Chaoli Wang. TransGraph: hierarchical exploration of transition relationships in time-varying volumetric data. *IEEE Trans. on Vis. and Comp. Graphics*, 17(12):2015–24, 2011.

[54] S. Gumhold. Maximum entropy light source placement. In *Visualization, 2002. VIS 2002. IEEE*, pages 275–282, 2002.

[55] H. Guo, W. He, T. Peterka, H. W. Shen, S. M. Collis, and J. J. Helmus. Finite-time lyapunov exponents and lagrangian coherent structures in uncertain unsteady flows. *IEEE Transactions on Visualization and Computer Graphics*, 22(6):1672–1682, June 2016.

[56] Martin Haidacher, Stefan Bruckner, and Meister Eduard Gröller. Volume analysis using multimodal surface similarity. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1969–1978, October 2011.

[57] Martin Haidacher, Stefan Bruckner, Armin Kanitsar, and Meister Eduard Gröller. Information-based transfer functions for multimodal visualization. In *VCBM*, pages 101–108. Eurographics Association, oct 2008.

[58] Robert Haimes. pv3: A distributed system for large-scale unsteady cfd visualization. In *AIAA paper*, pages 94–0321, 1994.

[59] Peter Hastreiter, Jörg Freund, Günther Greiner, and Thomas Ertl. Fast mutual information based registration and fusion of registered tomographic image data. In *Eds.), Workshop: Digitale Bildverarbeitung in der Medizin, Vol. 5, GI, Deutsche Gesellschaft fuer medizinische Informatik (GMDS*, pages 146–151, 1997.

[60] M.D. Hathaway, G. Herrick, J. Chen, and R. Webster. Time accurate unsteady simulation of the stall inception process in the compression system of a US army helicopter gas turbine engine. In *31st Annual International Symposium on Computer Architecture, 2004. Proceedings*, pages 166–177, 2004.

[61] Y. He, M. Mirzargar, S. Hudson, R.M. Kirby, and R.T. Whitaker. An uncertainty visualization technique using possibility theory: Possibilistic marching cubes. *International Journal for Uncertainty Quantification*, 5(5):433–451, 2015.

[62] D. L. G. Hill, P. G. Batchelor, M. Holden, and D. J. Hawkes. Medical image registration. *Physics in Medicine and Biology*, 46(3):R1, 2001.

[63] William M. Wells III, Paul Viola, Hideki Atsumi, Shin Nakajima, and Ron Kikinis. Multi-modal volume registration by maximization of mutual information. *Medical Image Analysis*, 1(1):35 – 51, 1996.

[64] International CFD Database, http://cfd.cineca.it/.

[65] H. Jänicke, M. Bottinger, and G. Scheuermann. Brushing of attribute clouds for the visualization of multivariate data. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1459–1466, 2008.

[66] H. Jänicke and G. Scheuermann. Visual analysis of flow features using information theory. *Computer Graphics and Applications, IEEE*, 30(1):40–49, 2010.

[67] H. Jänicke, A. Wiebel, G. Scheuermann, and W. Kollmann. Multifield visualization using local statistical complexity. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1384–1391, 2007.

[68] H. Jänicke, A. Wiebel, G. Scheuermann, and W. Kollmann. Multifield visualization using local statistical complexity. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1384–1391, Nov 2007.

[69] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In *2012 IEEE Pacific Visualization Symposium (PacificVis)*, pages 1–8, 2012.

[70] Waqas Javed, Bryan McDonnel, and Niklas Elmqvist. Graphical perception of multiple time series. *IEEE Transactions on Visualization and Computer Graphics*, 2010.

[71] Guangfeng Ji, Han-Wei Shen, and R. Wenger. Volume tracking using higher dimensional isosurfacing. In *Visualization, 2003. VIS 2003. IEEE*, pages 209–216, Oct 2003.

[72] Guangfeng Ji and Han wei Shen. Feature tracking using earth mover's distance and global optimization, pacific graphics, 2006.

[73] Shan Jiang, Xiaobo Zhou, Tom Kirchhausen, and Stephen T. C. Wong. Tracking molecular particles in live cells using fuzzy rule-based system. *Cytometry Part A*, 71A(8):576–584, 2007.

[74] C.R. Johnson and J. Huang. Distribution-driven visualization of volume data. *Visualization and Computer Graphics, IEEE Transactions on*, 15(5):734–746, Sept 2009.

[75] Heike Jänicke, Michael Böttinger, Xavier Tricoche, and Gerik Scheuermann. Automatic detection and visualization of distinctive structures in 3d unsteady multi-fields. *Comput. Graph. Forum*, 27(3):767–774, 2008.

[76] David Kao, Alison Luo, Jennifer L. Dungan, and Alex Pang. Visualizing spatially varying distribution data. In *Proceedings of the Sixth International Conference on Information Visualisation, 2002*, pages 219–225, 2002.

[77] Vasileios Karavasilis, Christophoros Nikou, and Aristidis Likas. Visual tracking using the earth mover's distance between gaussian mixtures and kalman filtering. *Image and Vision Computing*, 29(5):295–305, 2011.

[78] M. Khoury and R. Wenger. On the fractal dimension of isosurfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1198–1205, Nov 2010.

[79] J. M. Kniss, R. Van Uitert, A. Stephens, G. S. Li, T. Tasdizen, and C. Hansen. Statistically quantitative volume visualization. In *VIS 05. IEEE Visualization, 2005.*, pages 287–294, Oct 2005.

[80] Ulrich Kohler and Frauke Kreuter. *Data Analysis using Stata, 2nd Edition*. StataCorp LP, 2009.

[81] A. Kumpf, B. Tost, M. Baumgart, M. Riemer, R. Westermann, and M. Rautenhaus. Visualizing confidence in cluster-based ensemble weather forecast analyses. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):109–119, Jan 2018.

[82] Sriram Lakshminarasimhan, Neil Shah, Stephane Ethier, Scott Klasky, Rob Latham, Rob Ross, and Nagiza F. Samatova. Compressing the incompressible with isabela: In-situ reduction of spatio-temporal data. In Emmanuel Jeannot, Raymond Namyst, and Jean Roman, editors, *Euro-Par 2011 Parallel Processing*, pages 366–379, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[83] Teng-Yok Lee and Han-Wei Shen. Visualizing time-varying features with tac-based distance fields. In *Visualization Symposium, 2009. PacificVis '09. IEEE Pacific*, pages 1–8, April 2009.

[84] Teng-Yok Lee and Han-Wei Shen. Efficient local statistical analysis via integral histograms with discrete wavelet transform. *IEEE Trans. on Vis. and Comp. Graphics*, 19(12):2693–702, 2013.

[85] H. Lehmann and B. Jung. In-situ multi-resolution and temporal data compression for visual exploration of large-scale scientific simulations. In *2014 IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV)*, pages 51–58, Nov 2014.

[86] H. Lehmann and B. Jung. In-situ multi-resolution and temporal data compression for visual exploration of large-scale scientific simulations. In *IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV), 2014*, pages 51–58, 2014.

[87] Xinyue Li and Han-Wei Shen. Adaptive Volume Rendering using Fuzzy Logic Control. In *IEEE VGTC Symposium on Visualization*, 2001.

[88] H. Ling and K. Okada. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29(5):840–853, 2007.

[89] Shusen Liu, J.A. Levine, P. Bremer, and V. Pascucci. Gaussian mixture model based volume visualization. In *Large Data Analysis and Visualization (LDAV), 2012 IEEE Symposium on*, pages 73–77, Oct 2012.

[90] Xiaotong Liu and Han-Wei Shen. Association analysis for visual exploration of multivariate scientific data sets. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):955–964, Jan 2016.

[91] Jay F. Lofstead, Scott Klasky, Karsten Schwan, Norbert Podhorszki, and Chen Jin. Flexible IO and Integration for Scientific Codes Through the Adaptable IO System (ADIOS). In *Proceedings of the 6th International Workshop on Challenges of Large Applications in Distributed Environments*, CLADE '08, pages 15–24. ACM, 2008.

[92] S.L. Lohr. *Sampling: Design and Analysis*. Advanced (Cengage Learning). Cengage Learning, 2009.

[93] Claes Lundstrom, Patric Ljung, and Anders Ynnerman. Local histograms for design of transfer functions in direct volume rendering. *IEEE Trans. on Vis. and Comp. Graphics*, 12(6):1570–1579, 2006.

[94] Alison Luo, David Kao, and Alex Pang. Visualizing spatial distribution data sets. In *Proceedings of the Symposium on Data Visualisation 2003*, VISSYM '03, pages 29–38, 2003.

[95] Jun Ma, Chaoli Wang, and Ching-Kuang Shene. Coherent view-dependent streamline selection for importance-driven flow visualization. *Proc. SPIE*, 8654:865407–865407–15, 2013.

[96] K. L. Ma. Machine learning to boost the next generation of visualization technology. *IEEE Computer Graphics and Applications*, 27(5):6–9, Sept 2007.

[97] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *Medical Imaging, IEEE Transactions on*, 16(2):187–198, April 1997.

[98] A.R. Martin and M.O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Visualization, 1995. Visualization '95. Proceedings., IEEE Conference on*, pages 271–, 1995.

[99] N. M. McDougall, N. A. Cumpsty, and T. P. Hynes. Stall inception in axial compressors. *Journal of Turbomachinery*, 112(1):116–123, 1990.

[100] C. Muelder and K. L. Ma. Interactive feature extraction and tracking by utilizing region coherency. In *2009 IEEE Pacific Visualization Symposium*, pages 17–24, April 2009.

[101] C. Muelder and Kwan-Liu Ma. Interactive feature extraction and tracking by utilizing region coherency. In *Visualization Symposium, 2009. PacificVis '09. IEEE Pacific*, pages 17–24, April 2009.

[102] B. Nouanesengsy, J. Woodring, J. Patchett, K. Myers, and J. Ahrens. Adr visualization: A generalized framework for ranking large-scale scientific data using analysis-driven refinement. In *Large Data Analysis and Visualization (LDAV), 2014 IEEE 4th Symposium on*, pages 43–50, Nov 2014.

[103] Harald Obermaier and Kenneth I. Joy. Local data models for probabilistic transfer function design. In *Eurographics Conference on Visualization (EuroVis 2013) Short Papers*, pages 43–47, 2013.

[104] S. Ozer, D. Silver, K. Bemis, and P. Martin. Activity detection in scientific visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 20(3):377–390, March 2014.

[105] S. Ozer, Jishang Wei, D. Silver, Kwan-Liu Ma, and P. Martin. Group dynamics in scientific visualization. In *Large Data Analysis and Visualization (LDAV), 2012 IEEE Symposium on*, pages 97–104, Oct 2012.

[106] N.R. Pal, V.K. Eluri, and G.K. Mandal. Fuzzy logic approaches to structure preserving dimensionality reduction. *Fuzzy Systems, IEEE Transactions on*, 10(3):277–286, Jun 2002.

[107] Josien P. W. Pluim, J. B. Antoine Maintz, and Max A. Viergever. Mutual-information-based registration of medical images: a survey. *IEEE Transcations on Medical Imaging*, pages 986–1004, 2003.

[108] E. Polat and M. Ozden. A nonparametric adaptive tracking algorithm based on multiple feature distributions. *Multimedia, IEEE Transactions on*, 8(6):1156–1163, Dec 2006.

[109] Frits H. Post, Benjamin Vrolijk, Helwig Hauser, Robert S. Laramee, and Helmut Doleisch. The state of the art in flow visualisation: Feature extraction and tracking. *Comput. Graph. Forum*, 22(4):775–792, 2003.

[110] Kai Pöthkow and Hans-Christian Hege. Positional uncertainty of isocontours: Condition analysis and probabilistic measures. *IEEE Trans. on Vis. and Comp. Graphics*, 17:1393–1406, 2011.

[111] Kai Pöthkow and Hans-Christian Hege. Nonparametric models for uncertainty visualization. In *Proceedings of the 15th Eurographics Conference on Visualization*, EuroVis '13, pages 131–140, 2013.

[112] Kai Pöthkow, Britta Weber, and Hans-Christian Hege. Probabilistic marching cubes. In *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis'11, pages 931–940, 2011.

[113] Kristin Potter, Joe Kniss, Richard Riesenfeld, and Chris R. Johnson. Visualizing summary statistics and uncertainty. *Computer Graphics Forum (Proceedings of Eurovis 2010)*, 29(3):823–831, 2010.

[114] Kristin Potter, Jens Krüger, and Christopher Johnson. Towards the visualization of multi-dimensional stochastic distribution data. In *Proceedings of The International Conference on Computer Graphics and Visualization (IADIS) 2008*, 2008.

[115] P. Purkait, N. R. Pal, and B. Chanda. A fuzzy-rule-based approach for single frame super resolution. *IEEE Transactions on Image Processing*, 23(5):2277–2290, May 2014.

[116] Mark Peter Rast. Compressible plume dynamics and stability. *Journal of Fluid Mechanics*, 369:125–149, 1998.

[117] Freek Reinders, Frits H. Post, and Hans J. W. Spoelder. Attribute-based feature tracking. In *Data Visualization '99*, pages 63–72. Springer Verlag, 1999.

[118] J. Rigau, M. Feixas, and M. Sbert. Shape complexity based on mutual information. In *Shape Modeling and Applications, 2005 International Conference*, pages 355–360, June 2005.

[119] T.J. Ross. *Fuzzy Logic with Engineering Applications*. Wiley, 2004.

[120] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[121] Brian E. Ruttenberg and Ambuj K. Singh. Indexing the earth mover's distance using normal distributions. *Proceedings of the VLDB Endowment*, 5(3):205–216, 2011.

[122] H. Saikia and T. Weinkauf. Global feature tracking and similarity estimation in time-dependent scalar fields. *Computer Graphics Forum*, 36(3):1–11, 2017.

[123] Ravi Samtaney, Deborah Silver, Norman Zabusky, and Jim Cao. Visualizing features and tracking their evolution. *Computer*, 27:20–27, 1994.

[124] Roman Sandler and Michael Lindenbaum. Nonnegative matrix factorization with earth mover's distance metric for image analysis. *IEEE trans. on pattern analysis and machine intelligence*, 33(8):1590–1602, 2011.

[125] Franz Sauer, Hongfeng Yu, and Kwan-Liu Ma. Trajectory-based flow feature tracking in joint particle/volume datasets. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints):1, 2014.

[126] Carlos E. Scheidegger, John M. Schreiner, Brian Duffy, Hamish Carr, and Claudio T. Silva. Revisiting histograms and isosurface statistics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1659–1666, 2008.

[127] Konrad Schindler and Hanzi Wang. Smooth foreground-background segmentation for video processing. In *Proceedings of the 7th Asian Conference on Computer Vision - Volume Part II*, ACCV'06, pages 581–590, Berlin, Heidelberg, 2006. Springer-Verlag.

[128] M. Schlemmer, M. Heringer, F. Morr, I. Hotz, M.-H. Bertram, C. Garth, W. Kollmann, B. Hamann, and H. Hagen. Moment invariants for the analysis of 2d flow fields. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1743–1750, Nov 2007.

[129] G. Sfikas, C. Constantinopoulos, A. Likas, and N. P. Galatsanos. An analytic distance metric for gaussian mixture models with application in image retrieval. In *Proceedings of the 15th International Conference on Artificial Neural Networks: Formal Models and Their Applications - Volume Part II*, ICANN'05, pages 835–840, Berlin, Heidelberg, 2005. Springer-Verlag.

[130] D. Silver and X. Wang. Volume tracking. In *In Proceedings of the Visualization '96 Conference*, pages 157–164. Computer Society Press, 1996.

[131] D. Silver and X. Wang. Tracking scalar features in unstructured data sets. *Proceedings Visualization '98 (Cat. No.98CB36276)*, 98, 1998.

[132] Chris Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages –252 Vol. 2, 1999.

[133] Jun Tao, Jun Ma, Chaoli Wang, and Ching-Kuang Shene. A unified approach to streamline selection and viewpoint selection for 3d flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):393–406, March 2013.

[134] H. Theisel and H.-P. Seidel. Feature flow fields. In *Proceedings of the Symposium on Data Visualisation 2003*, VISSYM '03, pages 141–148, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[135] D. Thompson, J. A. Levine, J. C. Bennett, P. T. Bremer, A. Gyulassy, V. Pascucci, and P. P. Pébay. Analysis of large-scale scalar data using hixels. In *Large Data Analysis and Visualization (LDAV), 2011 IEEE Symposium on*, pages 23–30, 2011.

[136] Thanh T.L. Tran, Liping Peng, Boduo Li, Yanlei Diao, and Anna Liu. Pods: A new model and processing algorithms for uncertain data streams. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 159–170, New York, NY, USA, 2010. ACM.

[137] F. Y. Tzeng, E. B. Lum, and K. L. Ma. An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Vis. and Computer Graphics*, 11(3):273–284, May 2005.

[138] F.-Y. Tzeng and Kwan-Liu Ma. Intelligent feature extraction and tracking for visualizing large-scale 4d flow simulations. In *Supercomputing, 2005. Proceedings of the ACM/IEEE SC 2005 Conference*, pages 6–6, Nov 2005.

[139] Tim Van de Cruys. Two multivariate generalizations of pointwise mutual information. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, DiSCo '11, pages 16–20, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[140] Pere-Pau Vázquez, Miquel Feixas, Mateu Sbert, and Wolfgang Heidrich. Automatic View Selection Using Viewpoint Entropy and its Application to Image-Based Modelling. *Computer Graphics Forum*, 22(4):689–700, 2003.

[141] I. Viola, M. Feixas, M. Sbert, and M.E. Groller. Importance-driven focus of attention. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):933–940, 2006.

[142] V. Vishwanath, M. Hereld, and M. E. Papka. Toward simulation-time data analysis and i/o acceleration on leadership-class systems. In *2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pages 9–14, 2011.

[143] W. von Funck, T. Weinkauf, H. Theisel, and H.-P. Seidel. Smoke surfaces: An interactive flow visualization technique inspired by real-world flow experiments. *IEEE Transactions on Visualization and Computer Graphics (Proceedings Visualization 2008)*, 14(6):1396–1403, November - December 2008.

[144] C. Wang, H. Yu, and K. L. Ma. Application-driven compression for visualizing large-scale time-varying data. *IEEE Computer Graphics and Applications*, 30(1):59–69, 2010.

[145] Chaoli Wang, Hongfeng Yu, R.W. Grout, Kwan-Liu Ma, and J.H. Chen. Analyzing information transfer in time-varying multivariate data. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 99–106, 2011.

[146] Chaoli Wang, Hongfeng Yu, and Kwan-Liu Ma. Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547–1554, nov 2008.

[147] Chaoli Wang, Hongfeng Yu, and Kwan-Liu Ma. Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1547–1554, nov 2008.

[148] Yang Wang, Hongfeng Yu, and Kl Ma. Scalable Parallel Feature Extraction and Tracking for Large Time-varying 3D Volume Data. *Eurographics Symposium on Parallel Graphics and Visualization*, D:55–62, 2013.

[149] Yunhai Wang, Wei Chen, Jian Zhang, Tingxing Dong, Guihua Shan, and Xuebin Chi. Efficient volume exploration using the gaussian mixture model. *Visualization and Computer Graphics, IEEE Transactions on*, 17(11):1560–1573, Nov 2011.

[150] Gunther H. Weber, Peer-Timo Bremer, Marcus S. Day, John B. Bell, and Valerio Pascucci. Feature tracking using reeb graphs. In Valerio Pascucci, Xavier Tricoche, Hans Hagen, and Julien Tierny, editors, *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*, pages 241–253. Springer Verlag, 2011. LBNL-4226E.

[151] Tzu-Hsuan Wei, Chun-Ming Chen, and Ayan Biswas. Efficient local histogram searching via bitmap indexing. *Computer Graphics Forum*, 34(3):81–90, 2015.

[152] Tzu-Hsuan Wei, Teng-Yok Lee, and Han-Wei Shen. Evaluating isosurfaces with level-set-based information maps. In *Proceedings of the 15th Eurographics Conference on Visualization*, EuroVis '13, pages 1–10, Aire-la-Ville, Switzerland, Switzerland, 2013. Eurographics Association.

[153] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, S. Leman, and C. North. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):131–141, Jan 2018.

[154] M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multidimensional histogram. *CVGIP: Graphical Models and Image Processing*, 32(3):328–336, 1983.

[155] Brad Whitlock, Jean M. Favre, and Jeremy S. Meredith. Parallel in situ coupling of simulation with a fully featured visualization system. In *Proceedings of the 11th Eurographics Conference on Parallel Graphics and Visualization*, EGPGV '11, pages 101–109. Eurographics Association, 2011.

[156] Pak Chung Wong and R. Daniel Bergeron. 30 years of multidimensional multivariate visualization. In *Scientific Visualization, Overviews, Methodologies, and Techniques*, pages 3–33, Washington, DC, USA, 1997. IEEE Computer Society.

[157] J. Woodring, J. Ahrens, J. Figg, J. Wendelberger, S. Habib, and K. Heitmann. In-situ sampling of a large-scale particle simulation for interactive visualization and analysis. In *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization*, pages 1151–1160. Eurographics Association, 2011.

[158] J. Woodring, M. Petersen, A. Schmeier, J. Patchett, J. Ahrens, and H. Hagen. In situ eddy analysis in a high-resolution ocean climate model. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):857–866, Jan 2016.

[159] Jonathan Woodring, James Ahrens, Timothy J. Tautges, Tom Peterka, Venkatram Vishwanath, and Berk Geveci. On-demand unstructured mesh translation for reducing memory pressure during in situ analysis. In *Proceedings of the 8th International Workshop on Ultrascale Visualization*, pages 3:1–3:8. ACM, 2013.

[160] Jonathan Woodring and Han-Wei Shen. Multiscale time activity data exploration via temporal clustering visualization spreadsheet. *IEEE Trans. Vis. Comput. Graph.*, 15(1):123–137, 2009.

[161] J. Xie, F. Sauer, and K. L. Ma. Fast uncertainty-driven large-scale volume feature extraction on desktop pcs. In *Large Data Analysis and Visualization (LDAV), 2015 IEEE 5th Symposium on*, pages 17–24, Oct 2015.

[162] Di Yang, E.A. Rundensteiner, and M.O. Ward. Analysis guided visual exploration of multivariate data. In *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*, pages 83–90, 2007.

[163] Chang Huai You, Kong Aik Lee, and Haizhou Li. Gmm-svm kernel with a bhattacharyya-based distance for speaker recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(6):1300–1312, Aug 2010.

[164] Hamid Younesy, Torsten Möller, and Hamish Carr. Visualization of time-varying volumetric data using differential time-histogram table. In *Proceedings of the Fourth Eurographics / IEEE VGTC Workshop on Volume Graphics 2005, Stony Brook, NY, June 20-21, 2005*, pages 21–29. Eurographics Association, 2005.

[165] H. Yu, C. Wang, R. W. Grout, J. H. Chen, and K. L. Ma. In situ visualization for large-scale combustion simulations. *IEEE Computer Graphics and Applications*, 30(3):45–57, 2010.

[166] L. Zhang, Q. Deng, R. Machiraju, A. Rangarajan, D. Thompson, D. K. Walters, and H.-W. Shen. Boosting techniques for physics-based vortex detection. *Computer Graphics Forum*, 33(1):282–293, 2014.