Explorando conjuntos de dados volumétricos
multidimensionais variantes no tempo usando
projeções

Christian Jorge Daniel Wong Cruz

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura:

Explorando conjuntos de dados volumétricos multidimensionais variantes no tempo usando projeções

Christian Jorge Daniel Wong Cruz

***Orientadora:* Profa. Dra. Rosane Minghim**

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

USP – São Carlos
Novembro de 2012

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

W837e Wong Cruz, Christian Jorge Daniel
Explorando conjuntos de dados volumétricos
multidimensionais variantes no tempo usando projeções
/ Christian Jorge Daniel Wong Cruz; orientadora
Rosane Minghim. -- São Carlos, 2012.
107 p.

Dissertação (Mestrado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2012.

1. Visualização. 2. Visualização Volumétrica. 3.
Projeções Multidimensionais. I. Minghim, Rosane,
orient. II. Título.

A meus pais, Jaime e Lily.

Agradecimentos

É difícil escrever aqui depois de ter vivido tantas experiências em tão pouco tempo. Não posso deixar de agradecer a todos os que estiveram do meu lado nesse período fora do meu país.

Agradeço a Deus, por não ter-me deixado sozinho em momento nenhum, por colocar do meu lado sempre as pessoas certas, sempre na hora certa. *“Tudo posso naquele que me conforta”* (Filipenses: 4,13).

Agradeço aos meus pais, Jaime e Lily, por ter-me motivado sempre a crescer. Pela educação e amor que nunca deixaram de me dar, é por vocês que estou aqui e por vocês chegarei mais longe. Agradeço também aos meus irmãos Grace e Jaime que não deixaram de torcer por mim e por terem-se sentido sempre orgulhosos de cada passo que dei. Agradeço ao meu pequeno sobrinho e amigo Alejandro, por ter-me esperado em casa, sempre com muita alegria.

Agradeço à Juliana, linda companheira, confidente e cúmplice, por ter ficado do meu lado até no momento mais difícil da minha vida. Não tenho palavras para expressar o alívio de ter conhecido você naquela aula de forró. Não posso deixar de lado essa linda família que me adotou em São Carlos, muito obrigado Seu Julio, Dona Ana e Lu.

Agradeço à minha orientadora, Profa. Dra. Rosane Minghim, pela paciência e apoio durante o tempo do meu mestrado. Sem o seu apoio profissional, acadêmico e pessoal, eu não teria concluído esse trabalho com sucesso.

Agradeço aos professores Maria Cristina e João, pelos aportes ao trabalho na minha qualificação e pelos conselhos na escrita do artigo para publicação. Agradeço também ao professor Leandro por ter disponibilizado um conjunto de dados para testar no meu trabalho.

Agradeço mais uma vez aos professores João Paulo Gois (UFABC) e Haim Levkowitz (UMass Lowe) por terem aceitado participar da minha banca e pelas importantes contribuições que ajudaram a aprimorar esse trabalho.

Agradeço a todos os meus amigos que fiz aqui em São Carlos. Especialmente aos meus colegas e amigos do grupo de Visualização, Imagens e Computação Gráfica (VICG-ICMC/USP). Danilo cantor e Thiago foram peças importantes para clarear e formalizar as minhas idéias e ao Carlos, Francisco e a Ju pelos conselhos para melhorar a escrita da dissertação. Agradeço à Magda e o Marco Antônio, por terem aberto as portas da sua casa para que eu pudesse ficar em São Carlos.

Queria reservar um agradecimento especial aos meus amigos que organizaram um churrasco para a compra da passagem de volta ao Brasil quando passei dificuldades familiares, dentre todos eles estão os meus amigos peruanos Alex Ocsa, Alex Cuadros, Alvaro, Aurea, Arturo, Carolina, Edwin, Edward, Eduardo, Erick, Fernando, Fredy, Frizzi, Gabriel, Gladys, Ivar, JP, Jeaneth, Johana, Jorge Valverde, Jorge Ramirez, Juan Carlos Gutiérrez, Laura, Lizbeth, Lucesita, Lucia, Lourdes, Mayra, Nathalie, Nils, Oscar, Pedro, Renato, Rosa, Roni, Victor e Jorge, junto com os meus amigos “estrangeiros” Alceu, Bruno, Danilo Medeiros, Hitoshi, Ju, Mariana, Pablo, Paulo, Renato, Rosane Maffei, Sergis Kan, Thiago, Valéria e a sua mãe, Luize, Patricia Hilario, Leonardo e os amigos do LMACC que ajudaram anonimamente, Vinicius Caridá, Marcos e Miguel Caro. Muito obrigado pela ajuda nesses momentos difíceis, a amizade de vocês me deu muita força para continuar.

Agradeço a CAPES e a FAPESP (Processo: 2009/13438-9) pelo auxílio financeiro, que foi de grande importância para o desenvolvimento desse trabalho.

Enfim, agradeço a todas as pessoas que, de uma forma ou de outra, vem contribuindo com meu crescimento profissional e pessoal.

Sumário

Sumário	vi
Lista de Figuras	ix
Lista de Tabelas	xi
Resumo	xiii
Abstract	xv
1 Introdução	1
1.1 Contextualização	1
1.2 Motivação	2
1.3 Contribuições do trabalho	3
1.4 Organização	3
2 Visualização de volumes variantes no tempo	5
2.1 Considerações iniciais	5
2.2 Conceitos básicos	6
2.3 Codificação e Compressão	9
2.4 <i>Rendering</i>	12
2.5 Malhas irregulares	13
2.6 Detecção de Características	14
2.7 Multidimensionalidade	16
2.8 Considerações finais	18
3 Visualização multidimensional baseada em projeções	21
3.1 Considerações iniciais	21
3.2 Conjuntos de dados multivariados	22
3.2.1 Dados atmosféricos do furacão Isabel	22
3.2.2 Simulação de turbulências em Astrofísica	23
3.2.3 Visualização de instabilidades de Dinâmica dos Fluidos	23
3.3 Técnicas de projeção multidimensional	24
3.3.1 Fastmap	25
3.3.2 LSP	29
3.3.3 Piece wise Laplacian-based Projection, PLP	31

3.3.4	PLMP	32
3.4	Trabalhos baseados em Projeções Multidimensionais	33
3.5	Considerações finais	37
4	Projetando dados volumétricos variantes no tempo	39
4.1	Considerações iniciais	39
4.2	<i>Fastmap*</i> para dados variantes no tempo	40
4.2.1	Abordagem proposta	40
4.2.2	Seleção dos pivôs	40
4.2.3	Implementação iterativa	42
4.2.4	Complexidade Computacional	44
4.3	<i>Recursive Laplacian-based Neiboorhood Projection</i>	45
4.3.1	Abordagem proposta	45
4.3.2	Pré-processamento	46
4.3.3	Solução do sistema linear	47
4.3.4	Codificação incremental	48
4.3.5	Complexidade Computacional	48
4.4	Resultados e Avaliação das Técnicas propostas	48
4.4.1	Tempo	49
4.4.2	<i>Stress</i>	50
4.4.3	<i>Layout</i> gerado	52
4.4.4	Alta dimensionalidade	55
4.5	Considerações finais	56
5	Melhorando a exploração através de projeções	59
5.1	Considerações iniciais	59
5.2	<i>Cluttering</i> de agrupamentos	60
5.2.1	Descrição da proposta: Reprojeção de agrupamentos	61
5.2.2	Seleção de atributos relevantes	62
5.2.3	Exemplos e Resultados	64
5.3	Análise de atributos específicos	69
5.3.1	Descrição da proposta: <i>Scatter Projections</i>	70
5.3.2	<i>Fastmap*</i> aplicado a <i>Scatter Projections</i>	70
5.3.3	Exemplos e Resultados	72
5.4	Considerações finais	74
6	Conclusões	77
A	Apêndice A	91
B	Apêndice B	95

Lista de Figuras

2.1	Representação dos <i>voxels</i> do volume armazenados em uma grade regular (Engel et al., 2004).	6
2.2	Definição de voxel segundo Paiva et al. (1999), os círculos indicam pontos na grade com valor associado. Na Figura 2.2(a) o <i>voxel</i> possui um valor constante em toda a área do hexaedro. Enquanto na Figura 2.2(b) o valor dentro do <i>voxel</i> é obtido interpolando valores dos pontos marcados.	8
2.3	interface de visualização de dados volumétricos multivariados, usando coordenadas paralelas para extração de características. Extraído de Akiba et al. (2007).	14
2.4	Proposta de visualização para detecção de características em volumes variantes no tempo. Extraído de Hadwiger et al. (2008).	15
2.5	Interface principal da ferramenta de exploração de volumes definida em Blaas e Post (2008). A parte superior contém os componentes de visualização espacial (duas visões de fatias do volume e uma de sua iso-superfície 3D). O gráfico de coordenadas paralelas está posicionado no meio e a interface de controle na parte inferior.	17
2.6	Representações gráficas resultantes da abordagem definida em Linsen et al. (2008).	18
3.1	Ilustração da <i>Lei dos Cossenos</i> . Projeção de O_i sobre a reta definida pelos pivôs O_a e O_b . O valor da projeção de O_i é o valor resultante de P_{ik} ($k = 1$) calculado na equação 3.3. * Adaptado de (Faloutsos e Lin, 1995)	26
3.2	Projeção dos pontos O_i e O_j no hiperplano H , perpendicular à linha definida por O_a e O_b da figura anterior 3.2(a). Abaixo, a Figura 3.2(b), mostra a composição final da projeção bidimensional usando como eixos os valores calculados nas duas primeiras recursões do algoritmo. * Adaptado de (Faloutsos e Lin, 1995)	27
3.3	Exemplo de projeção de um volume usando o algoritmo da Fastmap em dois instantes diferentes. As Projeções de diferentes <i>timesteps</i> não preservam a coerência temporal do conjunto de dados.	29
3.4	Alguns exemplos de projeções de coleção de documentos. Estes são coloridos com base sobre o assunto dos documentos (Paulovich et al., 2008).	31
3.5	Interface da ferramenta de visualização de fibras DTI baseado em projeção multi-dimensional. Extraído de Chen et al. (2009).	34
3.6	Visualização de fibras DTI coordenada com outras técnicas de visualização: Projeção (superior direito), <i>clustering</i> hierárquico (inferior esquerdo) e codificação de cor (inferior direito). Extraído de Jianu et al. (2009).	34
3.7	Exploração de fibras proposta baseada em projeções. Extraído de Poco et al. (2012).	35

3.8	Sistema de visualização de campos vetoriais apoiado por projeções interativas. Extraído de Daniels et al. (2010).	35
3.9	Extraído de Guo et al. (2011b)	36
4.1	Heurística linear para busca dos pivôs (O_a e O_b). Ela falha quando um objeto P cai fora da região definida pelos pivôs, mas não prejudica a qualidade da projeção.	41
4.2	Valor de <i>stress</i> ao longo do tempo das projeções usando os pivôs encontrados nos 5 subconjuntos propostos.	42
4.3	Diagramas de blocos da <i>Fastmap</i> (4.3(a)), o pré-processamento da <i>Fastmap*</i> (4.3(b)) e a projeção da <i>Fastmap*</i> (4.3(c)).	44
4.4	Execução das recursões da <i>RLNP</i> . A primeira recursão é similar à executada pela <i>PLP</i> onde extraímos pontos de controle locais, na segunda recursão são extraídas relações de vizinhança dos pontos de controle locais e são identificados os pontos de controle temporais, sendo estes finalmente projetados usando uma técnica <i>MDS</i> . As regiões (a) e (b) indicam onde são usadas técnicas de <i>MDS</i> na <i>PLP</i> e <i>RLNP</i> , respectivamente.	46
4.5	Valor de <i>stress</i> das projeções <i>Fastmap</i> , <i>Fastmap*</i> , <i>PLP</i> , <i>RLNP</i> e <i>PLMP</i> usando diferentes conjuntos de dados.	51
4.6	<i>Layouts</i> produzidos pelas técnicas <i>PLMP</i> , <i>Fastmap*</i> e <i>RLNP</i> quando usadas para projetar os instantes de tempo 20 do <i>VisContest 2004</i> , 90 do <i>VisContest 2008</i> e 01 do <i>VisContest 2011</i> .	53
4.7	<i>Layout</i> da projeção produzida pela <i>RLNP</i> . As áreas limitadas pelos círculos amarelo e vermelho mostram um exemplo de estruturas encontradas no interior do furacão.	54
4.8	Seleção de 6 agrupamentos na projeção <i>Fastmap*</i> do conjunto de dados do <i>VisContest 2008</i> junto com a visualização no espaço de objeto de 5 deles.	54
4.9	<i>Layout</i> da projeção <i>RLNP</i> do conjunto de dados do <i>VisContest 2011</i> .	55
4.10	Comparação dos valores de <i>stress</i> dos <i>layouts</i> produzidos pelas <i>Fastmap*</i> e <i>RLNP</i> em função à dimensionalidade dos dados.	56
5.1	Várias estruturas internas do volume são identificadas no espaço de atributos. Essas estruturas são agrupadas em regiões muito próximas limitando a descoberta de novas regiões de interesse.	60
5.2	Passos da abordagem proposta para solucionar o problema da separação de subagrupamentos descartando os atributos selecionados por um algoritmo de seleção de atributos.	62
5.3	Esquema das abordagens filtro (Figura 5.3(a)) e <i>wrapper</i> (Figura 5.3(b)). Extraído de John et al. (1994).	64
5.4	Experimento 1: Seleção e visualização de um pequeno agrupamento na projeção.	65
5.5	Melhores resultados obtidos na reprojeção dos pontos do subagrupamento selecionado. As três abordagens de seleção de atributos conseguiram gerar um melhor <i>layout</i> na segunda projeção.	65
5.6	Resultados não satisfatórios obtidos pela seleção de atributos usando a classe <i>ClassifierSubsetEval</i> .	66
5.7	Experimento 2: Seleção e visualização de um pequeno agrupamento na projeção.	67
5.8	Melhores resultados obtidos na reprojeção dos pontos do subagrupamento selecionado. As três abordagens de seleção de atributos conseguiram gerar um melhor <i>layout</i> na segunda projeção.	67
5.9	Resultados não satisfatórios obtidos pela seleção de atributos usando a as três abordagens.	68

5.10	Análise da concentração de H_2 no instante 90 do conjunto de dados da explosão. A projeção do mesmo instante de tempo mostra duas seleções feitas por um filtro nos dados (Figura 5.10(a)) e pelo usuário (Figura 5.10(b)) junto com os respectivos <i>boxplots</i> (Figuras 5.10(c) e 5.10(d)) para visualizar melhor os valores dos pontos selecionados. As seleções tentam representar os pontos com maior concentração de H_2 . Observamos também que o mapeamento do atributo usando uma escala de cor (<i>azul-verde-vermelho</i>) não foi muito bem sucedida por causa do mapeamento.	71
5.11	Esquema de construção do Scatter Projection. O eixo horizontal mapeia o valor de um atributo, enquanto no eixo vertical são mapeados os objetos multidimensionais por similaridade.	72
5.12	Dois <i>Scatter Projections</i> mostrando duas regiões diferentes para análise de atributos específicos.	73
5.13	Comparação dos resultados obtidos por Linsen et al. (2009) e os obtidos usando <i>Scatter Projections</i> . Observamos a turbulência encontra-se em volta da região onde o H_2 é mais predominante.	74

Lista de Tabelas

3.1	Tópicos dos concursos anuais de visualização da Conferência <i>IEEE VisWeek</i>	22
3.2	Comparação entre a <i>LSP</i> e a <i>PLP</i> . Tempo de execução em segundos (Paulovich et al., 2010a).	32
4.1	Tempos de execução das técnicas originais e modificadas. $pre(t)$ é o tempo de pré-processamento e $proj(t)$ é o tempo que demora para completar a execução da projeção. As técnicas originais não precisam o tempo de pré-processamento. mCPU <i>Fastmap*</i> é a <i>Fastmap*</i> paralelizada em CPU.	50
4.2	<i>Stress</i> obtido pelas projeções dos conjuntos de dados do <i>VisContest</i> usando as técnicas <i>Fastmap*</i> , <i>RLNP</i> e <i>PLMP</i>	52

Resumo

A área de visualização volumétrica engloba um conjunto de técnicas utilizadas na representação, manipulação e exibição de dados associados à região de um volume, possibilitando, assim, a exploração e melhor compreensão do interior de objetos de natureza tridimensional. Contudo, algumas limitações ainda são encontradas nessa área, como, por exemplo, a exploração de mais de um valor simultaneamente em conjuntos de dados volumétricos multivariados. Além desse desafio, outro objeto de grande interesse da comunidade científica é a exploração de volumes variantes no tempo. A complexidade nesse caso está em tratar ou processar uma quantidade muito grande de dados buscando descobrir propriedades, estruturas ou características que variam com o tempo. O presente trabalho propõe técnicas e abordagens, baseadas no conceito de projeções multidimensionais, visando dar apoio à análise de conjuntos volumétricos multivariados que variam no tempo. A primeira técnica proposta, denominada *Fastmap**, possibilitou a projeção de espaços de alta dimensionalidade em fluxo contínuo. A segunda técnica apresentada, denominada *RLNP*, permitiu a projeção de dados por vizinhança mantendo a coerência temporal nos dados projetados, além de possuir a capacidade de projetar espaços de alta dimensão com um nível de *stress* baixo. Também, propomos uma abordagem para a análise baseada em atributos, denominada *Scatter Projection*, que facilita a exploração focada em um atributo específico junto com a similaridade dos dados entre eles. Finalmente, propõe-se uma abordagem baseada na reprojeção de agrupamentos usando técnicas de seleção de atributos para tentar identificar melhor as estruturas internas do volume. Assim, o presente trabalho contribui no sentido de levantar e discutir limitações das técnicas disponíveis, e em seguida, buscar possibilidades de solução para tais questões, propondo técnicas e abordagens que possibilitam a exploração de grandes conjuntos de dados volumétricos multivariados, mantendo a coerência temporal.

Abstract

The area of volume visualization encompasses a set of techniques used for representation, manipulation and display of data associated with a region of a volume, thus enabling the exploration and understanding of the interior of three-dimensional objects. However, some limitations are still encountered in this area. For example, the simultaneous exploration of more than one value in multivariate volumetric datasets. Beyond this challenge, another issue of great interest to the scientific community is the exploration of time-varying volumes. The complexity of this case lies in treatment or processing of a very large amount of data, seeking to discover properties, structures, or characteristics that may vary in time. This work proposes techniques and approaches, based on the concept of multidimensional projections, in order to support multivariate volumetric analysis of time varying data sets. The first technique proposed, called *Fastmap**, enables the projection of high dimensional streaming data. The second technique presented, called *Recursive Laplacian-based Neighborhood Projection*, allows the projection of data sets based on neighborhoods, maintaining the temporal coherence in the projected data, besides having the ability to project high-dimensional spaces with a low level of stress. Also, we propose an approach for the analysis of specific attributes, referred to as *Scatter Projection*, which facilitates the exploration focused on a specific attribute and on the similarity between them. Finally, we propose an approach based on reprojection of groups using feature selection techniques for better identification of internal structures of the volume. Thus, this study contributes towards surveying and discussing limitations of the area, and then seeks ways of solving these issues, proposing techniques and approaches that enable the exploration of multidimensional volumetric time varying data sets, maintaining the temporal coherence.

Introdução

1.1 Contextualização

A Visualização é um método computacional que transforma dados simbólicos em geométricos, possibilitando que os usuários observem os resultados de simulações computacionais de fenômenos diferentes. A visualização científica oferece um método para compreender o que não está explicitamente visível. Ela enriquece o processo de análise científica e promove descobertas profundas e até não esperadas. Em vários campos, a visualização já está revolucionando a forma como os cientistas fazem ciência. Um dos principais objetivos da visualização científica é alavancar os métodos existentes provendo novas formas de descoberta por meio de métodos visuais ([McCormick et al., 1987](#)).

A visualização está convencionalmente dividida em duas subáreas: Visualização Abstrata ou Visualização de Informação (InfoVis) e Visualização Científica (SciVis). A primeira delas, Visualização Abstrata, tem como objetivo visualizar dados que não tem representação física predefinida. A visualização pode ser utilizada em diferentes áreas, como na análise visual de texto ([Dunsmuir et al., 2012](#); [Luo et al., 2012](#); [Thom et al., 2012](#); [Luo et al., 2012](#)), imagens ([Hu et al., 2010](#); [Eler et al., 2008](#); [Zhang et al., 2007](#); [Rafla, 2005](#)), músicas ([Zhu e Lu, 2005](#)), redes sociais ([Gabriel et al., 2011](#); [Gove et al., 2011](#); [Peterson, 2011](#)) dentre outros. Por outro lado, a visualização científica está focada na exploração e no melhor entendimento de dados que possuem representação física definida. Nessa subárea encontramos representações visuais e processos interativos para lidar com problemas onde as instâncias dos dados são complexas, o que tem gerado desafios aos pesquisadores da área. Um exemplo desses dados complexos são os dados volumétricos multivariados.

Esses volumes podem ter características particulares, como o tipo da malha, múltiplos atributos por nó e variação no tempo. Os dados usados neste trabalho possuem características que estão dentro dos dois últimos tipos. Portanto o foco do trabalho está na exploração de dados volumétricos multivariados variantes no tempo.

1.2 Motivação

A exploração de conjuntos de dados volumétricos está centrada em apenas um parâmetro, que pode ser um escalar, vetor, ou tensor ([Linsen et al., 2008](#)). Poucas tentativas têm sido feitas na exploração de mais de um valor (*feature*) simultâneo em conjuntos volumétricos multidimensionais. Dentre essas, as mais proeminentes, e que têm apresentado melhores resultados, são as que empregam técnicas de visualização de informação para explorar o espaço de atributos ou características em coordenação com o espaço do objeto ([Chen et al., 2009](#); [Daniels et al., 2010](#); [Guo et al., 2011b](#)).

Apesar de ser possível explorar mais de uma *feature* com o emprego de técnicas de visualização de informação, ainda existem poucos trabalhos que suportam a exploração de todo espaço de *features* simultaneamente. Dentre essas podemos destacar as que empregam coordenadas paralelas no processo de exploração (Capítulo 2). Embora a técnica de coordenadas paralelas seja efetiva para se revelar o relacionamento entre os atributos presentes no conjunto de dados, ela pouco pode revelar sobre as relações entre as instâncias de dados, apresentando, portanto, suporte limitado para a realização de tarefas para a exploração de padrões estruturais envolvendo entidades (voxels), como, por exemplo, a identificação de uma região de anomalias dentro de um volume que envolva combinações de atributos.

Além do desafio de explorar todos os atributos de um volume, um outro objeto de grande interesse da comunidade científica é a exploração de volumes variantes no tempo. Nesse caso, a idéia de um único volume é substituída por um conjunto de volumes representando fatias de tempo subsequentes sobre o mesmo fenômeno natural ou simulação. A complexidade nesse caso está em se tratar ou processar uma quantidade muito grande de dados buscando descobrir propriedades, estruturas ou características que variam com o tempo. Como exemplo podemos citar o entendimento dos fatores, ou combinação desses, que levam ao surgimento de uma estrutura nos dados. Casos típicos são vórtices em fluidos ou tumores em órgãos. Recentemente foram desenvolvidas abordagens baseadas em técnicas de projeção multidimensional ([Tejada et al., 2003](#)) para resolver esse tipo de problemas. Portanto o estudo é baseado nesse tipo de técnicas para explorar volumes multivariados com variação no tempo.

Poucas pesquisas estão sendo desenvolvidas no sentido de tratar volumes variantes no tempo utilizando o espaço de características. Novamente, nos deparamos com exemplos envolvendo o uso de coordenadas paralelas e que apresentam, portanto, a mesma limitação discutida anteriormente,

ou seja, a não capacidade ou capacidade limitada de revelar estruturas entre os *voxels* de um volume ou conjunto de volumes.

1.3 Contribuições do trabalho

No desenvolvimento desse trabalho estudamos algumas limitações das técnicas de projeção multidimensional para trabalhar com tipo de dado de interesse, que são os volumes multivariados variantes no tempo. A primeira dificuldade apresentada pela maioria das técnicas de projeção é a falta de suporte à coerência temporal dos dados, isto é, a não concordância dentre os *layouts* produzidos pela mesma técnica em diferentes instantes de tempo.

Para atacar esse problema é proposta a hipótese de que a fixação ou projeção em conjunto dos objetos representativos de cada instante de tempo torna estáveis os *layouts* produzidos por meio da projeção de diferentes instantes de tempo. A comprovação dessa hipótese resultou em duas técnicas de projeção multidimensional que suportam a coerência temporal do conjunto de dados. Essas técnicas de projeção são as principais contribuições desse trabalho de mestrado.

Além dessas contribuições principais, este trabalho de mestrado alcançou alguns resultados adicionais, os quais são descritos a seguir:

- Uma nova visualização denominada *Scatter Projection*, que é um híbrido entre Projeções Multidimensionais e Scatterplots;
- Uma nova abordagem de exploração de conjuntos de dados produzidos por simulações onde os agrupamentos de dados não são bem diferenciados uns dos outros;
- Um paper aceito no *IEEE Visualization Contest 2011* como contribuição sobre visualização de instabilidades em fluidos usando projeções multidimensionais;
- Um artigo submetido para publicação sobre o trabalho do mestrado.

O conjunto de ferramentas produzidas neste trabalho fornece novas formas de exploração de volumes variantes no tempo, anteriormente não disponíveis.

1.4 Organização

A presente dissertação está organizada em 4 capítulos principais. No Capítulo 2, são descritas as técnicas de visualização de volumes multidimensionais variantes no tempo existentes na literatura. Seguido da descrição de técnicas de projeção multidimensional que podem ser aplicadas para grandes conjuntos de dados que são detalhadas no Capítulo 3. No Capítulo 4, propõe-se duas técnicas para projeção de dados volumétricos variantes no tempo. Explorando os dados volumétricos usando as técnicas existentes e propostas, percebe-se que existem ainda algumas dificuldades

que devem ser tratadas. Essas dificuldades são exploradas no Capítulo 5, onde são propostas duas abordagens para exploração de volumes apoiados por projeções multidimensionais com foco em análise de atributos. Finalmente as nossas conclusões estão no Capítulo 6.

Visualização de volumes variantes no tempo

2.1 Considerações iniciais

A visualização é um método computacional que transforma dados simbólicos em geométricos, possibilitando a análise de simulações e computações feitas por analistas e cientistas. Através desse método computacional é possível identificar o que não é explicitamente visível, enriquecendo assim o processo de análise científica.

O desenvolvimento de aplicações de visualização fornece aos usuários novas ferramentas, que irão facilitar inúmeras tarefas, as quais, junto com as técnicas de visualização, são usadas na análise, exibição e exploração de grandes volumes de dados multivariados, que geralmente variam com o tempo, permitindo ao usuário identificar as características significativas e obter os resultados desejados.

O objetivo deste Capítulo é contextualizar e apresentar conceitos e classificações dentro da área de visualização de volumes variantes no tempo, e está dividido como segue: na Seção 2.2 é apresentada uma contextualização e conceitualização da visualização volumétrica em geral; nas seções seguintes será apresentada uma classificação dos métodos de visualização de volumes variantes no tempo, sendo essas a codificação e compressão (Seção 2.3), *rendering* (Seção 2.4), malhas irregulares (Seção 2.5), detecção de características (Seção 2.6) e multidimensionalidade (Seção 2.7); na Seção 2.8 são apresentadas algumas observações e discussões sobre a área.

2.2 Conceitos básicos

Nos últimos 20 anos, a área da visualização volumétrica tem sido estudada e definida como um conjunto de técnicas utilizadas na visualização de dados associados a regiões de um volume, tendo como principal objetivo a exibição do interior de objetos de natureza tridimensional, a fim de explorar sua estrutura e facilitar sua compreensão (McCormick et al., 1987).

Paiva et al. (1999) conceitualizam a visualização volumétrica como a classe de métodos relacionada com a representação, manipulação e exibição de conjuntos de dados volumétricos. No mesmo trabalho, a visualização volumétrica é generalizada como um processo de *rendering*, que tem como objetivo a obtenção de uma melhor compreensão da estrutura contida nos dados volumétricos.

Esses dados volumétricos podem ser obtidos por processos de simulação, assim como em satélites de observação da Terra, instrumentos médicos de aquisição, entre outros. Formalmente, uma simulação produz informação construída a partir de algum modelo matemático e a aquisição produz informação resultante da conversão de objetos existentes na natureza. Essas informações usualmente são armazenadas em uma grade computacional regular (ou simplesmente grade), retilínea, curvilínea ou irregular. O uso de uma grade específica depende do domínio da aplicação, sendo que em algumas aplicações os dados são amostrados em pontos aleatórios, e a grade pode ser esférica, curva, etc. Em alguns casos são usadas as grades estruturadas que também são denominadas malhas regulares (Figura 2.1), mas essa generalização pode exigir maior custo computacional dos algoritmos de visualização volumétrica (Engel et al., 2004). Outra forma de armazenar dados volumétricos é usando malhas irregulares ou grades não estruturadas, esse último caso é aplicado quando algumas regiões do volume requerem maior nível de detalhe (Ma et al., 2003).

Figura 2.1: Representação dos *voxels* do volume armazenados em uma grade regular (Engel et al., 2004).

Para explorar esses volumes precisa-se projetar o conjunto de dados em um plano de imagem bidimensional, com o objetivo de obter algum conhecimento da estrutura contida no dado volumé-

trico. Nesse caso, o processo de projeção do volume 3D em uma tela 2D é denominado *rendering*. Usualmente, o conjunto de dados é definido em uma malha tridimensional com um ou mais valores escalares e, possivelmente, um ou mais valores vetoriais em cada vértice da malha. Esse conjunto de valores faz com que o conjunto de dados seja denominado multivariado ou multidimensional. Como os dados volumétricos possuem natureza tridimensional optaremos por usar o termo multivariado no restante do texto, com exceção dos casos onde *multidimensional* seja o termo mais apropriado e não ambíguo dependendo do contexto.

As técnicas de visualização de volumes devem oferecer representações de dados simples, manipulação rápida dos dados e um processo de *rendering* razoavelmente rápido (Ling e Zhi-Yu, 2011; Elvins, 1992). Os usuários deveriam ser capazes de manipular os parâmetros e apreciar o resultado instantaneamente. Atualmente, poucos são os algoritmos capazes de atingir esse tipo de rendimento e, por isso, o desenvolvimento e refinamento de algoritmos de visualização de volumes são áreas importantes de estudo. O entendimento dos algoritmos fundamentais e a representação do volume são requisitos para essa tarefa.

O dado volumétrico pode ser definido por uma função f do espaço tridimensional:

$$f : R^3 \rightarrow R^n \quad (2.1)$$

Quando a função f é uma função discreta, a propriedade volumétrica é definida apenas em alguns pontos do espaço delimitado pelo volume. Nesse caso é necessário que o dado volumétrico seja reconstruído. Para este fim, é utilizada uma função de interpolação para que f seja contínua.

Pode-se utilizar a interpolação de ordem zero ou constante, a qual é a mais simples, uma vez que associa o valor da função f em qualquer ponto do espaço ao valor da função no ponto mais próximo onde ela é definida. O método de interpolação de ordem zero gera um conjunto de regiões com valores constantes de f ao redor dos pontos onde a função é definida.

No caso das regiões com valores constantes esse tipo de interpolação pode gerar *aliasing* na imagem final sendo, por isso, usada uma forma mais apropriada de interpolação, conhecida como interpolação de primeira ordem ou interpolação trilinear, assumindo que a propriedade volumétrica varia linearmente em cada uma das três direções axiais. Dado que a interpolação linear é mais suave que a interpolação de ordem zero, são geradas imagens de melhor qualidade e com menos *aliasing* (Movania et al., 2009).

Cada vetor multivariado do volume é denominado *voxel* e são geralmente organizados numa grade regular. Na literatura não há um consenso com relação ao conceito de *voxels*. Em trabalhos como o desenvolvido por Paiva et al. (1999) são explicados os motivos da ambiguidade. Em algumas publicações, *voxel* é o hexaedro definido em torno do valor amostrado. Em outras, o voxel é entendido como sendo o hexaedro cujos vértices são os valores amostrados. Esses conceitos são mostrados na Figura 2.2.

Para diferenciar os dois tipos de *voxel* Elvins (1992) define como *voxel* a área hexaedral de valor constante em torno do ponto central da grade, deixando a definição de célula para aqueles

(a) Hexaedro definido em torno do valor amostrado. (b) Hexaedro cujos vértices são os valores amostrados.

Figura 2.2: Definição de voxel segundo [Paiva et al. \(1999\)](#), os círculos indicam pontos na grade com valor associado. Na Figura 2.2(a) o *voxel* possui um valor constante em toda a área do hexaedro. Enquanto na Figura 2.2(b) o valor dentro do *voxel* é obtido interpolando valores dos pontos marcados.

hexaedros onde a contribuição do *voxel* para a imagem decai tanto quando aumenta a distância desde o centro. Diante disso, existem duas formas de abordar a visualização volumétrica. A abordagem do *voxel* tem a vantagem de não fazer suposições sobre o comportamento dos dados entre os pontos da grade. Por exemplo, somente os valores conhecidos dos dados são usados para gerar uma imagem. Pode-se dizer que, em alguns casos, a abordagem do *voxel* está baseada na interpolação de ordem zero. Por outro lado, na abordagem das células, o volume é visto como uma coleção de hexaedros onde o valor armazenado dentro dele não é constante. Essa técnica tenta estimar valores no interior da célula por interpolação entre os valores nas esquinas da célula. As imagens geradas usando a abordagem de células são mais suaves do que as imagens criadas com a abordagem de *voxel*. No entanto, a validade das células em dados baseados em imagens não pode ser verificada. As funções de interpolação comumente usadas nesta abordagem são a trilinear e tricúbica.

O valor do dado volumétrico (*voxel* ou célula), que pode ou não variar de acordo com a interpolação utilizada, é chamado propriedade ou grandeza. No mundo físico, essa propriedade é escalar, vetorial, matricial ou tensorial, e, tipicamente, representa uma densidade em cada ponto da imagem, como temperatura, pressão ou um valor medido ou simulado no espaço como a velocidade. Entretanto, essa propriedade pode também ser multivariada, isto é, cada *voxel* pode ter mais de um valor amostrado. A questão da multidimensionalidade é uma área em estudo em visualização científica e um objetivo parcial do projeto apresentado neste trabalho. Aqui é tratado o problema de lidar com múltiplos atributos ao mesmo tempo para a análise de volumes.

Outra área de estudo da visualização volumétrica é o desafio do tratamento de grandes conjuntos de dados volumétricos, cujo tamanho é, em geral, da ordem de vários *gigabytes*. Esses dados volumétricos podem ser obtidos como resultado de simulações, as quais, algumas vezes,

são a única forma possível de estudar um determinado fenômeno. Nesses casos, a visualização volumétrica passa a ser fundamental no estudo do problema. Existem também dados volumétricos obtidos pela digitalização do objeto de interesse através de métodos como a Ressonância Magnética (MRI), Tomografia Computadorizada (CT), Emissão de Póstron (PET) e Emissão de Fotón (SPECT) ou técnicas como varredura por laser (microscópios confocais) e varredura por elétrons (microscópios eletrônicos). Nesse caso, os dados volumétricos são adquiridos e digitalizados.

As diversas fontes dos dados volumétricos têm possibilitado que a visualização volumétrica seja amplamente utilizada em medicina, geociências, sensoriamento remoto, meteorologia, astrofísica, química, metalurgia, morfologia, engenharia mecânica e várias outras áreas científicas, sendo as propriedades de medida mais utilizadas a densidade, pressão, temperatura, carga eletrostática e velocidade. Por exemplo, na área da dinâmica dos fluidos, os cientistas utilizam a visualização volumétrica para gerar imagens de valores tridimensionais representantes do fluxo de um fluido, permitindo a observação de locais de alta vorticidade e a compreensão do comportamento do sistema.

A visualização volumétrica também tem sido aplicada à exploração de volumes que possuam alguma variação nos valores dos *voxels* em função do tempo. Nesses trabalhos, o foco principal da pesquisa é criar ou adaptar técnicas existentes para exploração e visualização de dados volumétricos para manter a coerência temporal. O objetivo principal desse trabalho de mestrado é providenciar esta coerência espacial ao longo do tempo para volumes multivariados.

A Seção seguinte resume os trabalhos desenvolvidos nessa subárea de visualização volumétrica variante no tempo. Ma (2003) divide os métodos de visualização de volumes variantes no tempo em cinco categorias: codificação e compressão (Seção 2.3), *rendering* (Seção 2.4), malhas irregulares (Seção 2.5), detecção de características (Seção 2.6) e multidimensionalidade (Seção 2.7).

2.3 Codificação e Compressão

Os conjuntos de dados volumétricos apresentam uma enorme quantidade de elementos. Porém, pode-se reduzir o tamanho do conjunto de dados usando codificação e compressão. Isso pode permitir que o conjunto de dados caiba na memória de textura, eliminando-se a necessidade de transferência de dados entre a memória de textura (existentes em alguns modelos de placas gráficas) e a memória principal. Também pode tornar possível o armazenamento temporal de grandes conjuntos de dados na memória principal, evitando-se assim a excessiva leitura de informação no disco. Outro benefício da compressão é a redução em I/O (*Input/Output*). Mesmo que o conjunto de dados não couber na memória de textura, a transferência dos dados compactados pode ser substancialmente mais rápida do que com os dados descompactados, permitindo a visualização interativa. Além disso, a leitura de dados comprimidos a partir do disco requer menos tempo do que a leitura de dados sem compressão.

Tanto na codificação como na compressão dos dados, esses podem ser tratados segundo duas abordagens principais: dados homogêneos e dados não homogêneos.

Na primeira abordagem de **conjuntos de dados homogêneos**, o volume variante no tempo é tratado como um dado em quatro dimensões (4D), sendo as três primeiras a posição espacial do dado e o tempo a quarta dimensão. O tratamento do volume como um dado 4D depende das características dos dados. Por exemplo, se a diferença entre as resoluções espacial e temporal fosse grande, poderia tornar-se difícil localizar a coerência temporal em certos subdomínios dos dados. Por conseguinte, deve-se considerar o domínio do tempo separadamente para codificação.

Entre as abordagens que tratam os dados como homogêneos, tem-se o trabalho proposto por [Wilhelms e Gelder \(1994\)](#), que codifica os dados variantes no tempo com uma árvore 4D (extensão da *octree*) e um modelo de erro/importância para controlar a taxa de compressão e qualidade da imagem. [Linsen et al. \(2002\)](#) introduzem um modelo baseado no esquema de subdivisão *4th-root-of-2* acoplado com um esquema *wavelet B-spline* linear para representar dados volumétricos variantes no tempo em múltiplos níveis de detalhe. [Woodring et al. \(2003\)](#) visualizam dados em 4D por fatias ou *rendering* volumétrico no espaço 4D. Os hiperplanos e as hiperprojeções resultantes mostram algumas características espaço-temporais. [Kraus e Ertl \(2002\)](#) propõem uma representação adaptativa dos dados de textura que armazena tanto os índices dos blocos de dados empacotados quanto os fatores de escala para especificação das resoluções dos blocos de dados. A representação proposta por eles pode ser usada para duas, três ou quatro dimensões, e a decodificação é feita na placa gráfica. [Binotto et al. \(2003\)](#) empacotam efetivamente os dados variantes no tempo numa textura em três dimensões, que armazena os índices adaptativamente em blocos não homogêneos e os valores em blocos homogêneos. A abordagem funciona bem quando o volume é altamente correlacionado tanto espacial quanto temporalmente.

[Wang et al. \(2009\)](#) também realizam uma compressão de dados homogêneos. Os autores propõem uma abordagem de compressão composta por três passos principais, sendo o primeiro a partição do volume separadamente em cada instante de tempo, criando uma lista de blocos com diferentes tamanhos. Essa partição é baseada no domínio de conhecimento estabelecido pelos cientistas. Depois, os blocos espaciais são misturados em nós de uma *octree* ao longo da dimensão do tempo para criar blocos espaço-temporais e utilizar a coerência temporal na compressão dos dados. Finalmente, os blocos são codificados baseados nas faixas dos dados. Esse processo de compressão é executado como um passo de pré-processamento enquanto a decodificação é feita em tempo real usando os recursos da placa gráfica do computador.

Seguindo a mesma idéia, [Cao et al. \(2010\)](#) propõem um esquema de codificação baseado na técnica de compressão DXT (*Data Extract*). Essa codificação dos dados está composta por duas etapas: pré-processamento para compressão DXT e *rendering* baseado nos dados comprimidos. No pré-processamento, o conjunto de dados originais são divididos em múltiplos blocos que são adaptados para o tamanho da memória da placa gráfica. Os valores das variáveis são quantizados de valores em ponto flutuante para binários, conseguindo uma compressão da quarta parte do tamanho original. Posteriormente, os blocos quantizados, são comprimidos na placa gráfica usando

a compressão DXT atingindo um sexto do tamanho dos dados, sendo a compressão total de 24 : 1. A técnica de compressão DXT é novamente aproveitada por [Cao et al. \(2011\)](#) propondo uma abordagem denominada *Compressão Inteligente*. Essa abordagem está baseada em diretrizes de análise automática, as quais conseguem manter as características dos dados na etapa da compressão.

Na segunda abordagem, os dados são tratados como **dados não homogêneos**, onde a dimensão do tempo é separada das dimensões espaciais. Por exemplo, as diferenças na codificação da coerência de dados entre os intervalos de tempo consecutivos podem resultar em uma redução significativa do tamanho dos dados. No entanto, essa abordagem é limitada à navegação sequencial do aspecto temporal dos dados, ou seja, a navegação deve começar a partir do primeiro instante de tempo e não desde algum instante de interesse.

Pode-se usar quantização, juntamente com *octree* e codificação diferencial, isto é, a eliminação de informação redundante em instantes de tempo seguidos. Essas técnicas são usadas para comprimir eficazmente tanto no domínio espacial e temporal dos domínios dados ([Du et al., 2009](#)). Subárvores podem ser fundidas para as etapas de tempos consecutivos. Durante o processamento, pode-se reutilizar imagens parciais construídas a partir de subárvores que não mudam no tempo em etapas subseqüentes.

Um exemplo de tratamento dos dados como não homogêneos é o trabalho de [Akiba et al. \(2005\)](#), onde foi proposto um processo de visualização de três etapas baseada em compressão. A primeira etapa do processo é a conversão dos dados de ponto flutuante para múltiplos níveis de resolução, que podem ser acessadas progressivamente em cada resolução com potência de dois. Esse processo é realizado usando um esquema de multirresolução baseado em wavelets. Na segunda etapa, os dados são quantizados em quantidades inteiras necessárias para o *rendering* do volume em *hardware*. A seguir, o usuário seleciona uma função de transferência de opacidade e os *voxels* interessantes são empacotados em um esquema eficiente de representação de textura para posterior *rendering*. Esse esquema de empacotamento reduz até 80% dos dados. Finalmente, o volume é desempacotado e renderizado na placa gráfica para obter uma visualização interativa.

Entre outros trabalhos, tem-se a proposta de [Shen e Johnson \(1994\)](#), que introduz o *rendering* diferencial do volume e explora a coerência temporal dos dados, comprimindo-os de um modo substancial, limitando-se, porém, a uma busca sequencial do aspecto temporal dos dados. [Ma et al. \(1998\)](#) integram quantização não uniforme usando *octrees* junto com codificação diferencial e *rendering* rápido compartilhando subárvores entre espaços de tempo consecutivos. [Shen et al. \(1999\)](#) refinam o desenho derivando uma estrutura de dados hierárquica chamada *Time-Space Partitioning (TS) tree*, que captura a coerência temporal e espacial de um campo variante no tempo. Neste trabalho, uma *octree* é usada para particionar espacialmente o volume, e uma árvore binária é empregada para armazenar a informação temporal. Por outro lado, [Li et al. \(2003\)](#) constroem pacotes de *voxels* de volumes simples. Cada *voxel* é colocado num bloco de textura juntando *voxels* similares e retirando os espaços vazios. Esses pacotes ou blocos de textura são organizados numa árvore BSP para leitura e *rendering* eficiente dos blocos visíveis em uma ordem correta. O *rendering* é feito na placa gráfica.

Percebemos que, nos últimos anos, a abordagem de dados não homogêneos não está sendo utilizada, dado que não aproveita diretamente a natureza temporal dos conjuntos de dados.

2.4 *Rendering*

Embora uma codificação apropriada facilite o processo de *rendering*, ainda existem muitas questões associadas com o tempo de execução dessa etapa. Quando os recursos computacionais disponíveis não podem acomodar interação direta com o conjunto de dados para que os cientistas possam navegar por ele livremente em ambos os domínios espacial e temporal, deve-se procurar representações alternativas dos dados ou recursos de computação adicionais. Uma representação alternativa dos dados relaciona-se com a compressão adicional, a geração automática da função de transferência ou de extração de características. Recursos adicionais poderiam ser obtidos por técnicas de computação paralela distribuída.

Nos últimos anos, os paradigmas da programação em placas gráficas evoluíram rapidamente, porém, muitas aplicações baseadas em *hardware* têm sido desenvolvidas. Uma área que se beneficia da capacidade de processamento gráfico é a visualização volumétrica, especialmente no processo de *rendering*.

No caso das técnicas de *rendering* de volumes acelerado por *hardware*, tenta-se carregar o volume na placa gráfica. Neste caso, o volume pode ser renderizado em tempo real. Os principais limitantes da programação em placa gráfica são o tamanho reduzido de memória e a lentidão de transmissão dos dados da memória principal para a memória da placa gráfica, em comparação com a velocidade de transmissão da memória da placa gráfica para os processadores gráficos.

Nos trabalhos desta área, tem-se [Lum et al. \(2001\)](#) que utiliza codificação temporal dos dados volumétricos indexados, sendo esses rapidamente decodificados em placa gráfica, enquanto [Schneider e Westermann \(2003\)](#) usam quantização vetorial para comprimir dados variantes no tempo no domínio espacial com compressão e *rendering* feito em *hardware*.

No trabalho de [Anagnostou et al. \(2000\)](#), explora-se a coerência temporal para renderizar só as partes variantes de cada fatia e usa-se codificação *run-length* para comprimir o domínio espacial dos dados. Procurando aproveitar a coerência temporal dos dados, [Gao et al. \(2004\)](#) explora a coerência de oclusão temporal (ou TOC do inglês *Temporal Occlusion Coherence*) para acelerar o *rendering* de volumes variantes no tempo. Os autores desenharam um algoritmo paralelo de *rendering* usando funções OPO (Optimized Plenoptic Opacity functions) ([Gao et al., 2003](#)) que codifica a mínima obstrução visual possível, procurando manter a efetividade e escalabilidade do *rendering*. [Wang et al. \(2005\)](#) apresenta um *framework* para *rendering* paralelo de volumes usando o *Wavelet-based time-space partitioning tree* ou WTSP. Nele, os dados são convertidos em uma hierarquia de multiresolução espaço-temporal usando a transformada *Wavelet*, para ser armazenada posteriormente no *time-space partitioning tree*. O *rendering* compõe as diferentes resoluções espaço-temporais para gerar a imagem final em paralelo.

Podemos observar em trabalhos como [Wang et al. \(2009\)](#); [Cao et al. \(2010, 2011\)](#) que as categorias de codificação/compressão e *rendering* tem-se aproximado muito nos últimos anos. Dado que é possível aproveitar a codificação dos dados para acelerar o *rendering*. Podemos resumir essa junção das duas categorias como um processo de duas etapas, o pré-processamento dos dados (codificação e compressão), e a geração da imagem (*rendering*).

2.5 Malhas irregulares

Outra categoria dos métodos de visualização de volumes variantes no tempo, proposta por [Ma \(2003\)](#), refere-se às malhas irregulares ou grades não estruturadas, que são úteis para modelar conjuntos de dados com geometria complexa. Nesses modelos são utilizadas malhas mais finas somente em regiões onde é requerida maior precisão. Com isso é reduzido o tempo computacional no processamento assim como o espaço para armazenar os dados. Por outro lado, o uso de modelos discretizados em grades não estruturadas complica a tarefa de visualização, uma vez que os conjuntos de dados resultantes são geométrica e topologicamente irregulares. Uma limitação de trabalhar com esse tipo de dados não estruturados é que a necessidade de armazenar e acessar informação adicional sobre a estrutura da grade resulta em algoritmos com maior gasto de memória e sobrecarga computacional. Dada essas limitações, as maiores contribuições encontrados em pesquisas nessa categoria estão focados em analisar a coerência espacial mas não a coerência temporal, isto é, as pesquisas focaram inicialmente no estudo de instantes de tempo separadamente.

Dentre as propostas que buscam diminuir a sobrecarga computacional do *rendering* em dados variantes no tempo encontra-se o trabalho desenvolvido por [Wald et al. \(2007b\)](#), que descreve um sistema de *rendering* interativo de isosuperfícies de campos escalares tetraedrais usando técnicas de *ray casting* na CPU. Com isso, os autores conseguem processar na CPU grandes quantidades de dados não estruturados que variam no tempo. Para isso, usam métodos atuais de *ray casting* poligonal como *packet/frustum traversal* na hierarquia do volume ([Reshetov et al., 2005](#); [Wald et al., 2007a](#)) que são detalhados por [Reshetov et al. \(2005\)](#) e [Wald et al. \(2007a\)](#).

[Bernardon et al. \(2007\)](#) propõem um *framework* adaptativo para visualização de grades não estruturadas variantes no tempo. O *framework* é composto por quatro componentes principais: compressão, transferência de dados, visualização volumétrica assistida por hardware e *nível-de-detalle* dinâmico para interação. A compressão é feita sobre o domínio temporal. Posteriormente, os dados são descomprimidos na CPU e transferidos para o renderizador/visualizador que está baseado em uma versão modificada do algoritmo HAVS (do inglês *Hardware-Assisted Visibility Sorting*), proposto por [Callahan et al. \(2005b\)](#), que adiciona apoio para visualizar isosuperfícies de volumes que variam no tempo. Finalmente, os autores adicionam a abordagem chamada *nível-de-detalle* ([Callahan et al., 2005a](#)), adaptada para dados temporais.

Destaco mais uma vez a dificuldade em se encontrar trabalhos nessa área, que utilizem as malhas irregulares para a visualização de volumes variantes no tempo. Alguns trabalhos, como

por exemplo (Cignoni et al., 2004; Miranda e Celes, 2011; Muigg et al., 2011; Rocha et al., 2011), fazem uso das malhas irregulares para visualização de volumes, mas não ampliam essa aplicação a questão da variação no tempo. A seguir, descrevemos a quarta categoria de visualização de volumes variantes no tempo.

2.6 Detecção de Características

A busca de características em vários *terabytes* de dados apresenta muitas dificuldades. Por isso, a extração automática de características pode significar uma redução importante de armazenamento, bem como a diminuição do custo computacional para renderizar a visualização final.

A detecção de características variantes no tempo pode ser uma tarefa complicada, especialmente quando não podemos definir a característica de interesse analiticamente. Akiba et al. (2007) propõem uma interface de visualização que ajuda a definir as características por meio da coordenação de um conjunto de funções de transferência junto com o espaço de coordenadas paralelas para o *rendering* de vários atributos de uma forma interativa. Na Figura 2.3 observamos um exemplo do uso das coordenadas paralelas para extração de características em um conjunto de dados da área de mecânica dos fluidos onde o usuário pode selecionar faixas de valores nas coordenadas paralelas para visualizar uma característica específica.

(a) Nos eixos horizontais estão representados quatro atributos. A seleção em vermelho indica as características que foram extraídas dos atributos iniciais. (b) Visualização resultante da extração de características nas coordenadas paralelas.

Figura 2.3: interface de visualização de dados volumétricos multivariados, usando coordenadas paralelas para extração de características. Extraído de Akiba et al. (2007).

Hadwiger et al. (2008) propõem uma abordagem de detecção de características que permite a exploração interativa sem necessidade de processar novamente o volume. Para isso, são calculadas características do volume junto com algumas estruturas de dados adicionais onde estão contidos os resultados da detecção de características sobre os domínios dos parâmetros em lugar de parâmetros fixos. Esse processo é calculado somente uma vez e permite explorar o volume interativamente junto com todas as características contidas nele. A interação é feita por meio de seleções no

domínio de histogramas que mapeiam a densidade do *voxel* (eixo x) junto com uma característica específica (eixo y). A Figura 2.4 mostra diferentes seleções nos histogramas ao longo do tempo (ver Figura 2.4(a)) assim como a visualização resultante no espaço do objeto (ver Figura 2.4(b)).

(a) Seleção de características ao longo do tempo sobre um histograma que mapeia a densidade do *voxel* junto com outro atributo específico. (b) Visualização no espaço do objeto das características selecionadas pelo usuário.

Figura 2.4: Proposta de visualização para detecção de características em volumes variantes no tempo. Extraído de [Hadwiger et al. \(2008\)](#).

Ao contrário da maioria das técnicas de detecção de características temporais, [Muelder e Ma \(2009\)](#) tentam detectar as características aproveitando a coerência espaço-temporal dos dados, através de um método de predição e correção que usa uma etapa de predição para obter as melhores regiões de características possíveis, fazendo um seguimento das bordas das regiões para uma extração coerente das características variantes no tempo. Seguindo a mesma motivação de aproveitar a coerência temporal, [Jang et al. \(2012\)](#) usa uma representação funcional junto com técnicas de codificação utilizando a similaridade temporal entre instantes de tempo. Resultando em um *rendering* eficiente das representações funcionais por meio de árvores de particionamento binário de texturas.

É importante destacar que algumas das abordagens exploradas nesta seção têm como objetivo apresentar ao usuário uma ferramenta para a descoberta de características ocultas dentro do volume variante no tempo. Pode-se observar uma propriedade em comum, a qual está baseada na seleção de uma faixa ou região no espaço de características. Para estender essa seleção de características ao longo do tempo, os autores propõem a fixação das mesmas no espaço de características de diferentes instantes de tempo. Assim, podemos concluir que o sucesso de uma visualização de dados variantes no tempo está na coerência temporal do espaço de características oferecido ao usuário.

2.7 Multidimensionalidade

A visualização de volumes normalmente lida com o processamento de um grande número de instâncias de dados (ou voxels), tipicamente mais de alguns milhões de unidades, bem como o tratamento de múltiplos campos (vários valores de diferentes tipos – escalar, vetorial, tensorial) por voxel. O processo de se definir maneiras apropriadas para explorar completamente um conjunto de dados volumétrico é um dos grandes desafios atuais para a visualização volumétrica. A seleção de regiões ou objetos de interesse, bem como a definição de funções de transferências apropriadas, são parte das tarefas que um analista deve realizar de forma a extrair toda informação necessária de um volume.

Devido à habilidade das técnicas de visualização de informação de lidar com informação multivariada, existem esforços de se empregar essas técnicas para a análise das associações entre variáveis amostradas no espaço do objeto. Um dos primeiros exemplos da ligação explícita entre técnicas de visualização de informação com visualização de volumes foi o sistema WEAVE ([Gresh et al., 2000](#)). Essa idéia foi posteriormente expandida por [Doleisch et al. \(2003\)](#) por meio da formalização da definição de atributos ou características (variáveis) que podem ser usadas para coordenar (*linking-and-brushing*) uma visualização de *scatterplots* ([Cleveland, 1993](#)) com a visualização de um volume multivariado. Em todos os casos, a coordenação entre a visualização e o volume apenas considera duas características do espaço de atributos simultaneamente. [Blaas et al. \(2007\)](#) usaram *scatter plots* no espaço de atributos, no qual os dados multivariados são projetados em planos arbitrários.

[Sauber et al. \(2006\)](#) sugeriram a utilização de multigrafos para gerar as combinações de múltiplos campos escalares, no qual o número de nós no grafo cresce exponencialmente de acordo com o número de dimensões. Similarmente, [Woodring e Shen \(2006\)](#) definiram um conjunto de operadores Booleanos para a visualização dos campos escalares, enquanto [Ivanovska e Linsen \(2007\)](#) definiram operações no espaço de atributos, no qual o espaço de atributos é tridimensional e representado pelo espaço de cores *CIE $L^*a^*b^*$* . Eles acoplaram algoritmos automáticos de agrupamento baseados na subdivisão do espaço multivariado de atributos com interações para dividir e unir agrupamentos de forma a se definir as características de interesse.

Recentemente, algumas técnicas buscam lidar com todas as características simultaneamente, mas tais técnicas apresentam ainda algumas limitações a serem tratadas. Entre elas, destaca-se o trabalho de [Akiba e Ma \(2007\)](#) que sugeriram uma visualização triespacial que une a técnica de coordenadas paralelas ([Inselberg, 1985](#)) no espaço de atributos com o *rendering* do volume no espaço do objeto, além de *plots* unidimensionais variando no tempo. Similarmente, o trabalho de [Blaas e Post \(2008\)](#) define uma abordagem alternativa para a construção do gráfico de coordenadas paralelas que admite a exploração de grandes conjuntos de dados em tempo real. A Figura 2.5 apresenta a interface da ferramenta resultante desse trabalho.

[Linsen et al. \(2008\)](#) apresentaram uma abordagem baseada na extração de superfícies a partir de volumes de dados multivariados. A superfície segmenta os dados com respeito a uma função

Figura 2.5: Interface principal da ferramenta de exploração de volumes definida em [Blaas e Post \(2008\)](#). A parte superior contém os componentes de visualização espacial (duas visões de fatias do volume e uma de sua iso-superfície 3D). O gráfico de coordenadas paralelas está posicionado no meio e a interface de controle na parte inferior.

multivariada sobre o espaço de atributos. Nessa abordagem, a exploração do espaço de atributos é executada por meio de um método de agrupamento hierárquico baseado em densidade, cujos agrupamentos resultantes são mostrados na forma de conjuntos de níveis de densidade em um *layout* produzido pela técnica *3D star coordinates* ([Kandogan, 2001](#)). Com base nesse *layout*, o usuário pode selecionar os agrupamentos de características de interesse que correspondem a superfícies no espaço do objeto, ou seja, do volume. A Figura 2.6 apresenta as representações gráficas definidas nessa abordagem.

Alguns trabalhos recentes em visualização baseada em projeção têm melhorado a precisão dessas técnicas para conjuntos de dados abstratos com variado número de atributos. As projeções multidimensionais são basicamente abordagens para gerar mapeamentos de dados reduzindo a dimensionalidade dos dados para um espaço visual (2-D ou 3-D). Diante dos resultados apresentados nesses estudos com dados abstratos, o presente trabalho aproveita a capacidade das projeções multidimensionais de explorar o espaço de atributos aplicado à visualização de conjuntos de dados volumétricos variantes no tempo. Assim, no capítulo seguinte apresentamos os conceitos de projeção multidimensional, bem como algumas técnicas e aplicações recentes na área de visualização científica.

Figura 2.6: Representações gráficas resultantes da abordagem definida em [Linsen et al. \(2008\)](#).

2.8 Considerações finais

Nesse capítulo descrevemos os conceitos básicos de visualização volumétrica e os trabalhos que tratam da variação no tempo dos dados volumétricos. Com relação ao *rendering*, observamos que foram desenvolvidas técnicas para manter o tempo de processamento em um nível que não impeça aos cientistas interagir em tempo real. Algumas dessas abordagens sugerem a implementação paralela ou distribuída dos algoritmos de *rendering*. No nosso trabalho, usamos técnicas de *rendering* volumétrico direto (DVR) ou técnicas de extração de superfícies disponíveis na biblioteca *VTK* ([Schroeder et al., 2000](#)).

Observamos também na categoria de malhas irregulares que não foram encontrados muitos trabalhos relacionados a dados temporais. Nos próximos capítulos demonstraremos que a nossa abordagem de projeções multidimensionais pode ser facilmente aplicada à exploração de fenômenos modelados em malhas irregulares.

Dentre as categorias estudadas, duas se relacionam mais com o nosso trabalho, sendo essas detecção de características e multidimensionalidade. Na primeira, observamos que os trabalhos de detecção de características são similares. As duas propostas estudadas propõem a seleção de características feita em um ambiente flexível e com coerência temporal, sendo essa a principal estratégia adotada neste trabalho.

Com relação a multidimensionalidade dos dados, observamos que as pesquisas existentes possuem a tendência de usar técnicas da área de visualização de informação para explorar dados científicos. Dentre as técnicas de visualização de informação previamente utilizadas podemos citar as coordenadas paralelas, que estão sendo utilizadas em alguns trabalhos recentes a fim de explorar dados multivariados. Diante disso, discutimos no presente trabalho a utilização de projeções

multidimensionais na exploração de dados volumétricos variantes no tempo. No próximo capítulo, apresentaremos alguns conceitos relacionados às projeções multidimensionais, bem como algumas técnicas de projeções e trabalhos que estão sendo desenvolvidos nessa área.

Visualização multidimensional baseada em projeções

3.1 Considerações iniciais

Um dos objetivos desse trabalho de mestrado foi a exploração de volumes multivariados que variam no tempo. Para isso, enfocamos esse capítulo na descrição de técnicas de Projeção Multidimensional, a qual reduz o número de dimensões do espaço multivariado para um espaço visual. Assim, a exploração final é feita diretamente no espaço de características equivalente ao espaço multivariado original.

Dentre os benefícios de explorar dados por meio de projeções multidimensionais, ressaltamos principalmente a capacidade de exploração do espaço de características, onde o objetivo é encontrar agrupamentos de dados no espaço de projeção que serão representados como estruturas internas do volume na visualização do espaço do objeto. Por outro lado, como é descrito nesse capítulo, duas grandes limitações das técnicas são a quantidade de dados a serem processados e a coerência temporal dos dados. Dado que a maioria das técnicas de projeção multidimensional foram projetadas para resolver um problema específico, falaremos das técnicas que podem ser adaptadas para explorar os dados de interesse neste trabalho.

Assim, nesse capítulo temos como objetivo apresentar e discutir a possibilidade de utilização de técnicas de projeção para conjuntos de dados multidimensionais variantes no tempo. Começamos o capítulo com a descrição dos conjuntos de dados usados ao longo desta dissertação (Seção 3.2). Na Seção 3.3 são apresentadas algumas definições relacionadas às projeções multidimensionais. Em seguida, são descritas as técnicas de projeção com potencial de aplicação a volumes multiva-

riados que variam no tempo, sendo essas *Fastmap* (Subseção 3.3.1), LSP (Subseção 3.3.2), PLP (Subseção 3.3.3) e finalmente PLMP (Subseção 3.3.4). Na Seção 3.4 apresentamos alguns trabalhos de visualização científica baseados em projeções multidimensionais. Finalizando o capítulo, na Seção 3.5 fazemos considerações sobre as técnicas de projeção descritas e sua relação com o presente trabalho.

3.2 Conjuntos de dados multivariados

Os conjuntos de dados usados nesse trabalho foram obtidos do Concurso de Visualização denominado *IEEE Visualization Contest* ou simplesmente *VisContest*. As informações completas dos concursos estão disponibilizadas na *Internet* e podem ser acessados em [VisContest \(2012\)](#).

A Tabela 3.1 lista os temas dos concursos passados, dos quais adotamos os conjuntos de dados com características volumétrica, multivariada e variante no tempo, que correspondem aos anos 2004 - “Dados atmosféricos do furacão Isabel” ([VisContest, 2004](#)), 2008 - “Simulação de turbulências em Astrofísica” ([VisContest, 2008](#)) e 2011 - “Visualização de instabilidades em Dinâmica dos Fluidos” ([VisContest, 2011](#)) que são descritos nas subseções seguintes.

Ano	Tópico
2004	Dados atmosféricos do furacão Isabel
2005	<i>Rendering</i>
2006	Simulação de Terremotos
2008	Simulação de turbulências em Astrofísica
2010	Visualização avançada para planejamento em Neurocirurgia
2011	Visualização de instabilidades em Dinâmica dos Fluidos
2012	Visualização de mudanças de fase em configurações atômicas

Tabela 3.1: Tópicos dos concursos anuais de visualização da Conferência *IEEE VisWeek*.

Para não perder informação temporal dos dados, o número de instantes de tempo é reduzido dinamicamente na hora da visualização. Essa última tarefa é feita de acordo com as necessidades do usuário.

3.2.1 Dados atmosféricos do furacão Isabel

O concurso de visualização do ano 2004 ([VisContest, 2004](#)) foi o primeiro dos concursos desse tipo promovidos dentro do *VisWeek*, que teve por objetivo fomentar a comparação de técnicas novas com as já estabelecidas, promover pontos de referência para a comunidade e criar um ponto de discussão dentro da conferência do *VisWeek*.

Intitulado como “fusão de visualizações”, o concurso de visualização liberou um conjunto de dados do Centro Nacional de Pesquisas Atmosféricas dos Estados Unidos que trata de uma

simulação do furacão Isabel. O conjunto de dados é composto por 48 instantes de tempo contendo vários valores escalares e vetoriais. Cada volume é composto por uma grade regular de dimensões $500 \times 500 \times 100$, que foi reduzida aplicando amostragem regular para $125 \times 125 \times 25$.

Como foi descrito anteriormente, cada *voxel* é composto por uma série de valores escalares e vetoriais contando um total de 13 onde seis delas são taxas de mistura primárias: **QCLOUD** (umidade nas nuvens), **QGRAUP** (precipitação de neve em forma de pequenas esferas de gelo), **QICE** (gelo nas nuvens), **QSNOW** (neve), **QVAPOR** (vapor de água) e **QRain** (chuva); duas taxas de mistura derivadas: **CLOUD** (umidade total, composta por **QCLOUD**+**QICE**) e **PRECIP** (precipitação total, composta por **QGRAUP**+**QRain**+**QSNOW**); dois escalares: **P** (pressão) e **TC** (temperatura); e um valor vetorial composto por três escalares correspondentes a: **U**, **V**, **W** (velocidade do ar nos eixos x , y e z). O vetor de velocidade do ar foi convertido em um valor escalar por meio do cálculo da magnitude, resultando assim, em um vetor com 11 atributos escalares para cada *voxel*.

3.2.2 Simulação de turbulências em Astrofísica

O segundo conjunto de dados que usamos nesse trabalho foi lançado no concurso de visualização do ano 2008 ([VisContest, 2008](#)) e consiste em uma série de volumes multivariados, produto de uma simulação desenhada para entender a formação de galáxias. Esse conjunto de dados foi produzido com o objetivo de estudar as instabilidades da explosão, nas quais podem-se encontrar frentes de ionização radiativas espalhadas ao redor de massas de gás fundamentais. Os dados estão organizados em malhas regulares de dimensão $600 \times 248 \times 248$ e descrevem 200 instantes de tempo. Cada *voxel* descreve 10 valores escalares, sendo eles: **Densidade** total de partículas e **temperatura** do gás, junto com 8 **tipos de abundâncias de massa** (H, H+, He, He+, He++, H-, H2, H2+). Adicionalmente, cada *voxel* possui também um vetor de velocidade. Desse vetor, extraímos a magnitude, a magnitude do *curl* e a divergência. Reduzimos também todos os instantes de tempo aplicando amostragem regular, gerando assim um volume de dimensões $150 \times 62 \times 62$.

3.2.3 Visualização de instabilidades de Dinâmica dos Fluidos

Esse terceiro conjunto de dados foi usado no concurso de visualização do ano 2011 ([VisContest, 2011](#)) e modela a simulação de uma bomba centrífuga, a qual é descrita por três modelos chamados *RANS*, *LES* e *LES/RANS* híbridos. Cada conjunto de dados completa uma rotação inteira da bomba centrífuga descrita por 80 instantes de tempo. Cada volume é descrito por uma malha irregular no qual cada nó possui 6 atributos associados. Os valores escalares são pressão estática, pressão total, e pressão cinética. Os valores vetoriais são velocidade relativa e velocidade absoluta. As malhas descrevem várias partes ou domínios da bomba centrífuga. Esse trabalho foi focado no domínio do rotor.

Em colaboração com outros estudantes do laboratório, o conjunto de dados foi processado e resultou em 22,478 *streamlines* geradas a partir do mesmo número de vértices aleatórios, nas quais foram posteriormente aplicadas a transformada de Fourier. O resultado foi um conjunto de dados derivado com 39 instantes de tempo contendo um total de 22,478 *streamlines* descritos por 57 características. O processamento dos dados está detalhado no Apêndice A.

Esses três conjuntos de dados foram processados para serem visualizados e explorados usando uma técnica de visualização de informação denominada Projeção Multidimensional, a qual será descrita na próxima seção junto com alguns métodos de projeção que suportam grandes conjuntos de dados.

3.3 Técnicas de projeção multidimensional

Considere um conjunto de dados D contendo n instâncias de dados representadas por vetores de m características em um espaço m -dimensional. Uma projeção multidimensional é definida como uma função que mapeia cada instância $d_i \in D$ em um elemento gráfico (por exemplo, um ponto, círculo, etc.) embutida em um espaço visual (1-D, 2-D ou 3-D). No *layout* resultante, as posições relativas dos elementos refletem algum tipo de relacionamento entre as instâncias de dados, tal como similaridade ou vizinhança dada por uma função de distância definida em um espaço m -dimensional (Paulovich e Minghim, 2008). Desta forma, usuários podem empregar sua habilidade visual para reconhecer padrões e estruturas baseadas em similaridade. Exploração é usualmente baseada na localização de grupos e subgrupos de elementos que possuem, idealmente, alta correlação no caso de serem projetados em uma mesma vizinhança.

Dois exemplos clássicos de técnicas de projeção são *Sammon's Mapping* (Sammon, 1964) e *Classical Scaling* (também conhecido como *Multidimensional Scaling (MDS)*) (Cox e Cox, 2000). A primeira técnica define uma função de perda baseada na diferença entre as distâncias dos elementos no espaço visual e as distâncias desejadas calculadas entre as instâncias de dados no espaço m -dimensional e então minimiza essa função empregando um processo iterativo conhecido como gradiente descendente. Na segunda técnica, uma matriz de distâncias duplamente centrada entre todos os pares de instâncias de dados é definida, e uma decomposição espectral é aplicada para se recuperar as coordenadas cartesianas dos elementos no espaço visual. Embora estas técnicas definam resultados altamente precisos em termos da preservação de distâncias, ambas apresentam alta complexidade computacional ($O(n^2)$), limitando a aplicação dessas em grandes conjuntos de dados.

Outra estratégia bem conhecida para criar projeções multidimensionais, originalmente definida como uma heurística para desenho de grafos, é a *Force-Directed Placement (FDP)* (Fruchterman e Reingold, 1991). No modelo FDP, cada instância é modelada como uma partícula e as suas

posições são determinadas como o local onde a soma das forças agindo sobre elas, gerada por todas as outras partículas (instâncias), é zero. Estas forças são proporcionais à diferença entre as distâncias desejadas entre os elementos visuais e as distâncias atuais. O *layout* final é obtido mudando a posição das partículas até que o sistema atinja um estado de equilíbrio. Como cada instância é afetada por todas as outras instâncias, uma iteração do modelo FDP é $O(n^2)$. Para reduzir tal complexidade, Chalmers (1996) definiu uma abordagem com iteração linear, onde as forças são determinadas usando amostras ao invés de todo conjunto de dados. Contudo, como n iterações são necessárias para se produzir o *layout* final, esta ainda é uma técnica com complexidade alta ($O(n^2)$).

Buscando reduzir essa complexidade, Morrison et al. (2002); Morrison e Chalmers (2004) criaram um modelo híbrido aproximando a técnica definida por Chalmers. Nesta abordagem, uma amostra pequena das instâncias de dados é inicialmente projetada no espaço visual. Então, as instâncias restantes são interpoladas considerando as instâncias amostradas mais similares. Isso reduz a complexidade para $O(n \sqrt[4]{n})$. Esse processo foi posteriormente otimizado por Jourdan e Melançon (2004), resultando em uma técnica $O(n \log n)$. Embora tais técnicas realmente reduzam a complexidade do modelo original, os *layouts* produzidos normalmente apresentam baixa preservação de similaridade ou vizinhança, especialmente para conjuntos de dados de alta dimensionalidade, devido às aproximações empregadas.

Posteriormente, foi apresentada uma outra abordagem chamada de *Least Square Projection* (LSP) (Paulovich et al., 2008), a qual tende a produzir resultados melhores em termos da preservação de vizinhança com custo computacional razoável ($O(n \sqrt{n})$). Na LSP, diferentemente da maioria das outras técnicas, a preservação de similaridade é buscada em pequenas vizinhanças e entre grupos de instâncias altamente relacionadas, característica indicada para espaços de alta dimensão, tais como dos volumes sendo considerados nesse trabalho. Outras técnicas de projeção multidimensional que podem ser aplicadas ou adaptadas para análise de dados volumétricos são a *Fastmap* (Faloutsos e Lin, 1995), *PLP* (Paulovich et al., 2011) e *PLMP* (Paulovich et al., 2010b) que são descritas nas próximas subseções. Escolhemos descrever com mais detalhe aquelas que foram adaptadas aos propósitos deste trabalho.

3.3.1 Fastmap

A *Fastmap*, proposta por Faloutsos e Lin (1995), é uma técnica de redução da dimensionalidade que pode ser empregada para projetar dados de um espaço n -dimensional em um espaço k -dimensional menor, por meio da projeção dos pontos originais sobre k retas mutuamente ortogonais. A ideia principal da técnica é projetar recursivamente os objetos originais em k hiperplanos, considerando os valores resultantes em cada um dos hiperplanos como as coordenadas dos pontos projetados.

Para definir as k retas ortogonais escolhemos dois Objetos O_a e O_b de cada hiperplano, sendo esses objetos denominados pivôs. A posição dos outros objetos O_i do conjunto será determinada

em relação aos dois pivôs usando a *Lei dos Cossenos* (Casey, 1889), sendo representada pela equação:

$$d(O_b, O_i) = d(O_a, O_i)^2 + d(O_a, O_b)^2 - 2x_i d(O_a, O_b) \quad (3.1)$$

que podemos usar para definir a posição de x_i do Objeto O_i a ser projetado. A equação resultante é:

$$x_i = \frac{d(O_a, O_i)^2 + d(O_a, O_b)^2 - d(O_b, O_i)^2}{2d(O_a, O_b)} \quad (3.2)$$

Os valores obtidos de x_i na equação são os valores da projeção quando queremos resolver o problema para $k = 1$. Na nossa notação o valor de x_i é o valor de P_{i1} ou, em geral, P_{ik} , sendo que i é o ID do objeto e k a dimensão do espaço visual. Então, a equação pode ser reescrita da seguinte forma:

$$P_{ik} = \frac{d(O_a, O_i)^2 + d(O_a, O_b)^2 - d(O_b, O_i)^2}{2d(O_a, O_b)} \quad (3.3)$$

Uma ilustração da *Lei dos Cossenos* é mostrada na figura 3.1.

Figura 3.1: Ilustração da *Lei dos Cossenos*. Projeção de O_i sobre a reta definida pelos pivôs O_a e O_b . O valor da projeção de O_i é o valor resultante de P_{ik} ($k = 1$) calculado na equação 3.3.

Adaptado de (Faloutsos e Lin, 1995)

Assim que o problema é resolvido para $k = 1$, esse pode ser generalizado para qualquer valor de $k < m$. Para isso, na segunda recursão do algoritmo projetamos os objetos em um hiperplano $(m - 1)$ -dimensional que é perpendicular à reta definida por O_a e O_b , sendo o problema da projeção na segunda iteração o mesmo problema original. A Figura 3.2 apresenta os objetos O_i e O_j projetados sobre um hiperplano H perpendicular à reta definida por O_a e O_b .

O último conceito que iremos descrever é como as novas distâncias $d(O'_i, O'_j)'$ são calculadas entre os objetos projetados no novo hiperplano. Essa nova distância é apresentada na equação 3.4 e é derivada da distância euclidiana:

(a) Exemplo da segunda recursão do algoritmo da Fastmap.

(b) Composição final da projeção no espaço 2D.

Figura 3.2: Projeção dos pontos O_i e O_j no hiperplano H , perpendicular à linha definida por O_a e O_b da figura anterior 3.2(a). Abaixo, a Figura 3.2(b), mostra a composição final da projeção bidimensional usando como eixos os valores calculados nas duas primeiras recursões do algoritmo.

Adaptado de (Faloutsos e Lin, 1995)

$$d(O'_i, O'_j)' = \sqrt{d(O_i, O_j)^2 - (P_{i(k-1)} - P_{j(k-1)})^2} \quad (3.4)$$

Com a definição dos hiperplanos e as funções de distância, os pontos podem ser projetados em um espaço k -dimensional aplicando-se o algoritmo recursivamente k vezes.

Para que a Fastmap seja capaz de obter boas projeções, a reta na qual os objetos serão projetados deve ser formada pelos objetos mais distantes do hiperplano. Pelo fato da busca dos pivôs

ter um custo computacional de $O(n^2)$, foi proposta por Faloutsos e Lin (1995) uma busca com complexidade aproximada de $O(n)$. Nessa busca é escolhido um objeto arbitrário e em seguida procura-se o mais distante dele. O próximo passo é a procura do objeto mais distante desse segundo, sendo, finalmente os dois últimos objetos utilizados como pivôs.

Um dos pontos positivos da *Fastmap* é que essa é uma técnica de baixa complexidade computacional, sendo possível, assim, a sua aplicação a grandes conjuntos de dados. Cabe ressaltar que, como é uma técnica recursiva, os conjuntos de dados devem possuir poucas dimensões para não ter um tempo de execução elevado, dado que para projetar uma recursão arbitrária, é necessário ter projetado previamente todos os pontos das recursões anteriores. Podemos ver os detalhes da técnica nos Algoritmos 3.1 e 3.2.

Algoritmo 3.1: Seleção dos pivôs

Entrada: D_s : Subconjunto de dados

Saída: O_a, O_b : Pivôs do hiperplano

- 1 $O_b \leftarrow$ um objeto arbitrário de D_s
- 2 $O_a \leftarrow$ o objeto mais distante de O_b de acordo com a função de distância na recursão atual
- 3 $O_b \leftarrow$ o objeto mais distante de O_a de acordo com a função de distância na recursão atual
- 4 Retornar os objetos O_a e O_b como os pivôs da recursão.

Algoritmo 3.2: Projeção Fastmap

Entrada: D_s : Subconjunto de dados

k : Recursão atual

Saída: D'_s : Conjunto de dados projetados

- 1 **se** $k \leq 0$ **então**
- 2 retorna
- 3 **fim**
- 4 Seja O_a e O_b os pivôs selecionados no Algoritmo 3.1
- 5 **se** $d(O_a, O_b) = 0$ **então**
- 6 $D'_{x,y} \leftarrow 0$
- 7 **senão**
- 8 **para** cada P em D_s **faça**
- 9 Calcular o valor do ponto projetado usando a Equação 3.3 armazenando o resultado em D'_s .
- 10 **fim**
- 11 **fim**
- 12 Executar o Algoritmo 3.2 com $k - 1$

Um exemplo da aplicação dessa técnica de projeção multidimensional para dados volumétricos é mostrado na Figura 3.3. As projeções dos instantes 60 e 90 baseadas nos valores das variáveis são demonstradas nas Figuras 3.3(a) e 3.3(c), assim como a visualização no espaço do objeto representado nas Figuras 3.3(b) e 3.3(d), respectivamente. O *layout* gerado na projeção é similar

ao gerado pelas outras técnicas (*PLP* (Paulovich et al., 2011), *PLMP* (Paulovich et al., 2010b), etc.). Como pode ser observado, duas projeções de diferentes instantes de tempo não preservam a coerência temporal do conjunto de dados.

Figura 3.3: Exemplo de projeção de um volume usando o algoritmo da *Fastmap* em dois instantes diferentes. As Projeções de diferentes *timesteps* não preservam a coerência temporal do conjunto de dados.

Finalmente, o custo computacional do *Fastmap* é determinado pelos dois passos executados k vezes. A escolha dos pivôs requer três iterações sobre os dados, resultando em uma complexidade de $O(3n) = O(n)$. O segundo passo é uma iteração sobre os dados calculando o valor da coordenada projetada usando a *lei dos cossenos*. Assim, a complexidade total é $O(k(3n + n)) = O(n)$, considerando que k (1, 2 ou 3) é um número baixo.

3.3.2 LSP

A *Least-Square Projection (LSP)* (Paulovich et al., 2008) é uma técnica de projeção multidimensional que engloba as boas características dos métodos de projeção linear e não linear. E, além de ser uma técnica rápida, pode ser empregada com sucesso em espaços esparsos de alta dimensionalidade, resultando em um posicionamento preciso dos pontos em termos de vizinhança. Para

o caso de documentos de texto, *LSP* pode gerar mapeamentos de alta precisão para aplicações que devem confiar na exploração do conteúdo ajudando localizar material interessante, e ainda é suficientemente rápida para lidar com uma quantidade razoável de documentos.

A *LSP* assume que cada elemento p_i de um conjunto de dados D pode ser escrito como uma combinação convexa de seus vizinhos mais próximos no espaço mapeado. Em termos matemáticos, seja $N_i = \{p_{i_1}, \dots, p_{i_{k_i}}\}$ o conjunto dos k_i vizinhos mais próximos dos p_i (assumindo uma medida de distância definida em D), e denotado por $(x_{p_{i_j}}, y_{p_{i_j}})$ as coordenadas de cada elemento $p_{i_j} \in N_i$ mapeado para $\mathbb{R}^2$. Portanto, as coordenadas bidimensionais de p_i podem ser calculadas por

$$(x_{p_i}, y_{p_i}) = \sum_{p_{i_j} \in N_i} \alpha_{ij} (x_{p_{i_j}}, y_{p_{i_j}}) \quad (3.5)$$

onde $\alpha_{ij} > 0$ e $\sum \alpha_{ij} = 1$.

Cada elemento em D gera uma equação vetorial, resultando em dois sistemas lineares homogêneos

$$Lx = 0; Ly = 0 \quad (3.6)$$

onde x e y são vetores representando as coordenadas x e y dos elementos mapeados no plano e L é a matriz derivada da Equação 3.5.

Os conjuntos N_i definem um *Nearest Neighbors Graph (NNG)* de D , isto é, um grafo conectando cada elemento em D aos seus vizinhos mais próximos. Pode ser demonstrado que o *rank* de L é $n - q$, onde n é o número de elementos em D e q é o número de componentes conexas do *NNG* (Sorkine e Cohen-Or, 2004). Assim, a fim de obter uma única solução não trivial para os sistemas da Equação 3.6, o *NNG* deveria ter apenas um componente conexo, o que pode ser garantido através da adição de novas arestas ligando componentes desconexos do *NNG*.

LSP adiciona informação geométrica ao mapa através da adição de pontos de controle como restrições aos sistemas da Equação 3.6. Pontos de controle são representantes de grupos de instâncias, cuidadosamente escolhidos para representar grupos de elementos altamente correlatos. A posição cartesiana dos pontos de controle é determinada através de uma técnica de posicionamento de alta precisão, tal como o *Classical Multidimensional Scaling* Cox e Cox (2000). Como eles são apenas uma fração dos dados, o tempo de processamento deste posicionamento de alta precisão não é relevante para o tempo computacional final da técnica.

Um exemplo da *LSP* é mostrado na Figura 3.4. Na Figura 3.4(a) é mostrado um mapa de uma coleção de documentos, composto de 300 mensagens de três diferentes grupos de discussão Hettich e Bay (1999). A Figura 3.4(b) apresenta uma projeção de um conjunto de dados obtidos a partir de um repositório de *Internet*, que inclui 1.624 artigos científicos sobre os temas da Bibliográfica Acoplamento (BC), Co-citação *Analysis* (SC), *Milgrams* (MG) e Visualização de Informação (IV) 2. Os maços BC e SC (vermelho e azul na figura, respectivamente) não são distinguíveis pois eles tratam os mesmos temas, de modo que eles são desenhados em conjunto, um resultado esperado para um mapeamento adequado.

(a) Projeção de mensagens de três diferentes grupos de discussão. (b) Projeção de trabalhos científicos em quatro temas distintos.

Figura 3.4: Alguns exemplos de projeções de coleção de documentos. Estes são coloridos com base sobre o assunto dos documentos (Paulovich et al., 2008).

3.3.3 Piece wise Laplacian-based Projection, PLP

A ideia da *PLP* é particionar o conjunto de dados original em subconjuntos menores nos quais a *LSP* original possa ser eficientemente empregada.

Para o particionamento dos dados usa-se o algoritmo *bisecting k-means* (Steinbach et al., 2000), que desempenha tão bem quanto outros métodos matematicamente bem formados como o PDDP (do inglês *Principal Direction Divisive Partitioning*) (Savaresi e Boley, 2004), gerando grupos bem balanceados de elementos similares. A coerência dos mapeamentos dos subconjuntos é um outro elemento a ser considerado. Escolhendo-se e posicionando-se adequadamente os pontos de controle é possível definir um relacionamento global entre os grupos de instâncias sem perder a qualidade da organização local fornecida pela *LSP* original. Esta escolha é descrita com detalhe em (Paulovich et al., 2010a).

Com a nova formulação, a técnica permite tratar rapidamente conjuntos de dados muito maiores, dando margem à sua utilização para a função de interação com grandes volumes de dados. Em aplicações como a medicina, imagens e volumes são compostos de unidades (pixels e voxels respectivamente) em grande quantidade. Selecionar elementos de interesse está sujeito a um processo muito difícil de pré-segmentação ou a operações também difíceis de definição de funções de transferência complexas.

Em termos de tempos de execução, *PLP* é duas ou três vezes mais rápida que o *LSP* original. A Tabela 3.2 apresenta o tamanho e o número de características de cada conjunto de dados e os tempos de execução para a criação de projeções utilizando o *PLP* e o *LSP* em uma CPU Intel (R) Xeon (R) 2.33 GHz, com 32GB de memória RAM. O procedimento usado no *PLP* para resolver os sistemas lineares e outras medidas necessárias, tais como *clustering*, a consulta dos vizinhos mais próximos, etc, é o mesmo utilizado pelo *LSP* original. Estes tempos de execução foram realizados

duas vezes para cada execução em cada conjunto de dados, ficando o valor médio dos tempos obtidos em execução. Alguns valores para a técnica de LSP não são mostrados pelo fato dela não ser capaz de calcular as projeções de uma quantidade razoável de tempo na máquina utilizada. Isso ilustra o grau de escalabilidade agora possível com a *PLP* sobre a da *LSP*.

Conjunto de dados	Tamanho	Atributos	<i>LSP</i>	<i>PLP</i>
Brain Image (Gabor)	64,435	30	16,868,994	33,064
Brain Image (Intensity)	64,435	14	3,523,094	26,504
Head Image (Gabor)	919,584	30	-	2,555,702
Papaya Volume	222,548	5	14,072,763	107,122
Head Volume	1,114,112	14	-	2,358,941
Head Volume (Watershed)	65,279	9	5,441,198	19,756
VisContest 2008	576,600	10	-	1,660,832

Tabela 3.2: Comparação entre a *LSP* e a *PLP*. Tempo de execução em segundos (Paulovich et al., 2010a).

3.3.4 PLMP

Ao contrário de outras técnicas, a *Part-linear Multidimensional Projection* ou *PLMP* (Paulovich et al., 2010b) requer somente as distâncias de um pequeno conjunto de amostras. A *PLMP* é construída através de um mapeamento linear das coordenadas cartesianas dos elementos representativos, os quais são posicionados inicialmente no espaço visual ($\mathbb{R}^2$). Essa característica faz com que a técnica seja executada mais rápida quando comparada a outros métodos quando grandes conjuntos de dados são processados sendo ainda competitiva em termos de precisão (*stress*).

A ideia principal da técnica é que se existe um mapeamento de um subconjunto de objetos n -dimensionais para o espaço visual, isto é, de $\mathbb{R}^n$ para $\mathbb{R}^p$, então podemos usar essa informação inicial para calcular uma transformação linear $\Phi : \mathbb{R}^m \rightarrow \mathbb{R}^p$ que projete o conjunto completo de uma forma eficiente. A qualidade da projeção final está diretamente relacionada à precisão do mapeamento dos elementos representativos. Portanto, Paulovich et al. (2010b) propõem o uso da técnica *Force Scheme* (Tejada et al., 2003) como o método de alta precisão.

Seja $H = \{h_1, h_2, \dots, h_n\}$ um conjunto de dados com instâncias $h_i \in \mathbb{R}^n$. Supondo que um subconjunto $H' = \{h'_1, h'_2, \dots, h'_k\}$ com $k \ll n$ instâncias de H foi previamente mapeado para $\mathbb{R}^p$ e, considerando a projeção de h'_i em $\mathbb{R}^p$ como $\bar{h}'_i$, o mapeamento linear ideal Φ deveria satisfazer a seguinte equação:

$$\Phi(h_i) = \bar{h}'_i \quad (3.7)$$

Essa equação permite o cálculo de uma aproximação da transformação Φ para cada $i = 1..k$ por meio do produto entre cada fila da matriz representando em Φ e as instâncias em H' . Especificamente, consideremos que o produto entre a primeira fila de Φ e $h'_i, i = 1, \dots, k$ resulta em:

$$\begin{aligned}
\phi_{11}x'_{1,1} + \dots + \phi_{1n}x'_{1,n} &= \bar{x}'_{1,1} \\
\phi_{11}x'_{2,1} + \dots + \phi_{1n}x'_{2,n} &= \bar{x}'_{2,1} \\
&\vdots \\
\phi_{11}x'_{k,1} + \dots + \phi_{1n}x'_{k,n} &= \bar{x}'_{k,1}
\end{aligned} \tag{3.8}$$

onde $\phi_{1i}, i = 1..n$ são entradas na primeira fila da matriz representando Φ , $(x'_{j,1}, \dots, x'_{j,n})$ são as coordenadas de h'_j em $\mathbb{R}^n$ e $\bar{x}'_{j,1}$ é a primeira coordenada de $\bar{h}'_j$ em $\mathbb{R}^p$. O sistema de equações 3.8 origina o sistema linear $\mathbf{L}\phi = b$, onde $\mathbf{L}$ é a matriz $k \times n$ com entradas $x'_{j,1}, \dots, x'_{j,n}$ na fila j , ϕ é a transposta da primeira fila de Φ , e b é o vetor da primeira coordenada de $\bar{h}_j, j = 1, \dots, k$.

Sendo que em geral k é maior que n na maioria de conjuntos de dados, a primeira fila ϕ de Φ pode ser aproximada resolvendo a seguinte equação:

$$\mathbf{L}^T \mathbf{L} \phi = \mathbf{L}^T b \tag{3.9}$$

Assim, a matriz Φ é calculada repetindo o mesmo processo para cada linha de Φ .

O maior custo computacional da *PLMP* está relacionado à posição dos grupos representativos que pode ser feito com qualquer técnica de alta precisão como a *Force Scheme*. A solução da Equação 3.9 não aporta grande custo computacional, dado que pode ser calculado com técnicas de otimização como a Cholesky ou o método de gradiente conjugado.

3.4 Trabalhos baseados em Projeções Multidimensionais

As projeções multidimensionais podem suportar a exploração de volumes multivariados por meio do mapeamento de atributos como pontos em um espaço usualmente bidimensional, assim os *voxels* com propriedades similares são mapeados em áreas vizinhas. Vários autores descrevem soluções para visualização exploratória de volumes que integram *layouts* de projeção ou outras técnicas de visualização multidimensional. Por exemplo, [Chen et al. \(2009\)](#) fornece a interface da Figura 3.5 com vistas em 2D e 3D que são coordenadas para auxiliar a exploração de fibras DTI (Diffusion Tensor Imaging). A representação bidimensional é calculada usando Classical Scaling ([Young e Householder, 1938](#)), o qual preserva as relações espaciais de cada fibra e elimina a poluição visual que é característica na visualização no espaço do objeto.

Outro trabalho com o objetivo de explorar fibras DTI usando técnicas de projeção é o proposto por [Jianu et al. \(2009\)](#). Nesse trabalho, o método usado é baseado em forças ([Eades, 1984](#)) e adota a técnica de aceleração de Chalmers ([Chalmers, 1996](#)) para reduzir os requisitos computacionais e alcançar um desempenho interativo. Adicionalmente, como podemos observar na Figura 3.6, as visualizações são coordenadas com uma árvore de *clustering* hierárquico (dendograma) para ajudar a exploração e o refinamento da seleção.

Seguindo a mesma linha de pesquisa sobre visualização de fibras DTI baseado em projeções multidimensionais, [Poco et al. \(2012\)](#) propõem o uso da técnica de projeção LAMP ([Joia et al.,](#)

Figura 3.5: Interface da ferramenta de visualização de fibras DTI baseado em projeção multidimensional. Extraído de [Chen et al. \(2009\)](#).

Figura 3.6: Visualização de fibras DTI coordenada com outras técnicas de visualização: Projeção (superior direito), *clustering* hierárquico (inferior esquerdo) e codificação de cor (inferior direito). Extraído de [Jianu et al. \(2009\)](#).

2011) para gerar projeções rápidas de grandes conjuntos de dados de fibras, oferecendo funcionalidades de seleção de agrupamentos de fibras similares para uma maior exploração.

(a) Visualização das fibras no espaço do objeto.

(b) Projeção das fibras no espaço de atributos.

Figura 3.7: Exploração de fibras proposta baseada em projeções. Extraído de [Poco et al. \(2012\)](#).

Por outro lado, [Daniels et al. \(2010\)](#) usam projeções para auxiliar a exploração interativa de campos vetoriais, observando que vetores vizinhos em um campo vetorial compartilham características similares, ainda quando estão separados espacialmente. A projeção favorece a percepção dessas relações de vizinhança. Podemos observar no sistema proposto da Figura 3.8 que os usuários podem modificar a localização dos pontos de controle e adicionar uma textura sobre uma tela interativa para melhorar a qualidade da projeção.

Figura 3.8: Sistema de visualização de campos vetoriais apoiado por projeções interativas.

Extraído de [Daniels et al. \(2010\)](#).

Finalmente, um trabalho que compartilha muita similaridade com o nosso trabalho foi proposto por [Guo et al. \(2011b\)](#), o qual propõe o uso de técnicas de projeção incorporado em um *plot* de coordenadas paralelas para definir funções de transferência para posterior *rendering* do volume. Um

exemplo do resultado dessa visualização está na Figura 3.9. Essa abordagem pode ser melhorada para ser escalável e alcançar melhor rendimento com conjuntos de dados de diferentes tamanhos e dimensionalidades (Guo et al., 2011a). A principal limitação encontrada é, novamente, a perda da coerência temporal na visualização dos dados ao longo de vários instantes de tempo, sendo essa a principal diferença com o nosso trabalho.

(a) Visualização no espaço do objeto dos agrupamentos encontrados na projeção embutida nas coordenadas paralelas.

(b) Projeção embutida nas coordenadas paralelas para análise e detecção de agrupamentos no espaço de características.

Figura 3.9: Extraído de Guo et al. (2011b)

3.5 Considerações finais

Nesse capítulo descrevemos as técnicas de projeção multidimensional e a sua possível aplicação à exploração de volumes variantes no tempo. A primeira técnica estudada foi a *Fastmap* a qual consegue projetar separadamente cada instante de tempo, porém, a principal limitação encontrada se traduz ao fato das projeções finais não preservarem a coerência temporal dos dados originais.

Posteriormente foram apresentadas as técnicas LSP e PLP. A primeira delas foi inicialmente aplicada para visualizar conjuntos de dados de texto que normalmente são os conjuntos de dados com um elevado número de dimensões. Para resolver esse problema em uma forma eficiente, observamos que a técnica projeta os dados baseado em vizinhança. O estudo da LSP foi importante porque ela foi a base da PLP, que consegue projetar conjuntos de dados maiores sendo uma técnica de projeção potencialmente útil para explorar volumes. Do mesmo modo que a *Fastmap* ela não consegue preservar a coerência temporal do conjunto de dados.

A terceira técnica de projeção estudada foi a PLMP, a qual foi publicada durante o desenvolvimento desse trabalho e consegue projetar grandes volumes de dados. Dada a característica de projetar dados em fluxo contínuo, as projeções de diferentes instantes de tempo mantêm a coerência temporal do conjunto de dados sempre que a matriz inicial seja mantida ao longo do tempo.

Finalmente estudamos os trabalhos da área de visualização científica que abordam o problema da exploração usando projeções multidimensionais, demonstrando assim que as técnicas de projeção poderiam ser usadas para explorar volumes variantes no tempo.

No próximo capítulo iremos propor algumas modificações para que as técnicas *Fastmap* e PLP consigam manter a coerência temporal nas projeções produzidas.

Projetando dados volumétricos variantes no tempo

4.1 Considerações iniciais

Como discutido no Capítulo 3 existem técnicas que foram criadas para explorar conjuntos de dados compostos por texto (ver Subseção 3.3.2), imagens e músicas (ver Subseção 3.3.3) ou objetos multidimensionais em geral (ver Subseção 3.3.1). Como foi demonstrado, essas técnicas podem ser facilmente usadas ou adaptadas para conjuntos de dados volumétricos que possuem vários atributos por cada instância de dado. A aplicação dessas técnicas de projeção em conjuntos de dados multidimensionais volumétricos variantes no tempo é um campo promissor. Dessa forma, estudamos a adaptação de técnicas de projeção para que possam preservar a coerência temporal do conjunto de dados.

A nossa hipótese para manter a coerência temporal é que a fixação ou projeção em conjunto dos objetos representativos de cada instante de tempo torna estáveis os *layouts* produzidos por meio da projeção de diferentes instantes de tempo. Com esse intuito, estudamos as limitações das técnicas atuais e propomos novas abordagens para manter a coerência temporal do conjunto de dados. Entende-se aqui por coerência temporal, a manutenção da disposição de pontos na projeção em instantes diferentes.

A seguir serão descritas as técnicas de projeção multidimensional propostas para dados variantes no tempo: a *Fastmap** é apresentada na seção 4.2 e a *RLNP* na seção 4.3. Por fim, a Seção 4.4 fornece os resultados obtidos com as novas abordagens propostas.

4.2 *Fastmap** para dados variantes no tempo

No Capítulo 3 descrevemos a técnica *Fastmap* proposta por Faloutsos e Lin (1995). Contudo, uma limitação dessa técnica é que não preserva a coerência temporal do conjunto de dados. Assim, nessa seção apresentamos a nossa nova implementação que tem a característica de poder ser aplicada em fluxo contínuo (ou *streaming*, do inglês), permitindo a análise visual de instâncias com características similares em conjuntos de dados grandes sem consumo alto de memória.

4.2.1 Abordagem proposta

O ponto de partida desta nova abordagem trabalha com os pivôs, os quais são usados para delimitar os hiperplanos que definirão os valores dos pontos projetados em cada recursão. A hipótese sobre a nossa abordagem é que fixando os pivôs ao longo de diferentes instantes de tempo, podemos garantir a coerência temporal das projeções resultantes.

Embora a hipótese possa garantir a coerência temporal como será visto nessa subseção, a implementação continua recursiva, precisando projetar todos os pontos das recursões anteriores para a projeção de um ponto em uma recursão arbitrária, não sendo portanto, possível a aplicação em fluxo contínuo. Por isso, iremos propor uma solução iterativa que consiste na aplicação da *Lei dos Cossenos* descrita na Seção 3.3.1 pela Equação 3.1 por cada instância.

Com essas duas modificações na técnica original será possível a aplicação da *Fastmap* com conjuntos de dados que variam no tempo. Como o algoritmo original foi alterado, a nova técnica proposta será denominada *Fastmap**. A seguir, explicamos a primeira modificação que é a demonstração da hipótese de fixar os pontos de controle para garantir a coerência temporal.

4.2.2 Seleção dos pivôs

Para manter a coerência temporal do conjunto de dados nas projeções produzidas pela *Fastmap** realizamos a seleção dos pivôs do conjunto de dados completo. Uma limitação de aplicar essa abordagem é a grande quantidade de cálculos de distância a serem executadas.

Com o objetivo de manter o custo computacional baixo ($O(n)$), a heurística de busca dos pivôs itera somente três vezes por recursão no conjunto de dados e, no final, nem todos os pontos são corretamente projetados com esses pontos de controle. Nos nossos testes encontramos que a quantidade desses pontos é desprezável, não afetando a qualidade da projeção. Ilustramos como acontece esse problema na Figura 4.1

Observamos que esse problema não causa muito impacto no *stress* da projeção dos instantes de tempo separadamente, uma vez que a quantidade de pontos que não caem dentro da linha definida pelos pivôs é muito baixa. Nos testes efetuados, a quantidade de pontos fora desta linha não passou dos 0,002% do total dos pontos em qualquer instante de tempo.

Figura 4.1: Heurística linear para busca dos pivôs (O_a e O_b). Ela falha quando um objeto P cai fora da região definida pelos pivôs, mas não prejudica a qualidade da projeção.

Com base nisso, podemos selecionar os pivôs em um subconjunto dos pontos e fixá-los para serem usados na projeção de cada instante de tempo. Os critérios propostos para obter esse subconjunto estão baseados na coerência temporal dos dados, que foi explicado no Capítulo 2. Nesse caso, espera-se que todos os instantes de tempo representem a evolução do mesmo fenômeno, assim a variação de um instante de tempo a outro não será muito alta. Concluindo, a variação das propriedades dos pivôs de um instante de tempo para outro também não é muito alta.

Fizemos testes extraindo os pontos de controle de 5 tipos de subconjuntos de dados, os quais descreveremos a seguir:

- **Primeiro:** Subconjunto formado pelos pontos do primeiro instante de tempo;
- **Central:** Subconjunto formado pelos pontos do instante de tempo localizado no meio da simulação;
- **Último:** Subconjunto formado pelos pontos do último instante de tempo;
- **Extremos:** Subconjunto formado pelos pontos do primeiro e último instante de tempo;
- **Todos:** Subconjunto formado pelos pontos do primeiro, último e instante de tempo central;

Como podemos observar na Figura 4.2, o menor valor de *stress* foi produzido pelos subconjuntos *extremos* e *central*. Portanto, usamos arbitrariamente esses conjuntos nos nossos testes da *Fastmap** aqui apresentados.

Para selecionar os pivôs, aplicamos o algoritmo recursivo da *Fastmap*, apresentado na Seção 3.3.1, no subconjunto selecionado. A única diferença é que a saída do algoritmo, dessa vez, será os pivôs do primeiro hiperplano (O_a , O_b) junto com os pivôs do segundo hiperplano (O_i , O_j) incluindo os valores dos pivôs no espaço projetado. É importante a aplicação de todos os passos

Figura 4.2: Valor de *stress* ao longo do tempo das projeções usando os pivôs encontrados nos 5 subconjuntos propostos.

do algoritmo da *Fastmap* porque é necessária a projeção de todos os pontos do primeiro hiperplano para obter os pivôs do segundo hiperplano.

Uma vez escolhido o subconjunto de pontos de onde vamos seleccionar os pontos de controle, podemos aplicar sobre ele a técnica *Fastmap* original para obter os pontos de controle a serem usados na próxima etapa, isto é, na projeção de todos os pontos do conjunto de dados, a ser descrita na subsecção a seguir.

4.2.3 Implementação iterativa

Uma limitação da implementação original da *Fastmap* é que, para encontrar os valores da segunda recursão em diante é necessário buscar novos pivôs em função dos valores projetados nas recursões anteriores. Como foi proposto na subsecção anterior, na *Fastmap** são encontrados os pivôs de um pequeno subconjunto representativo, sendo assim possível implementar a *Lei dos Cossenos* de uma forma iterativa.

Para projetar cada ponto separadamente sem precisar executar as recursões completas foi implementada a projeção por vetores, executando iterativamente a lei dos cossenos para cada dimensão. Essa implementação iterativa permite a projeção em *streaming* e a coerência temporal das projeções resultantes. O Algoritmo 4.1 detalha o resultado das modificações. Sendo que as distâncias usadas nas linhas 7, 8 e 9 foram descritas na Equação 3.4.

Finalmente, mostramos na Figura 4.3 os diagramas de blocos da *Fastmap* (4.3(a)), a etapa de pré-processamento da *Fastmap** (4.3(b)), e a etapa de projeção (4.3(c)).

Algoritmo 4.1: Projeção de um ponto com *Fastmap****Entrada:** V : Vetor de pontos a ser projetado O_a, O_b : Pivôs do primeiro hiperplano O_i, O_j : Pivôs do segundo hiperplano O'_a, O'_b : Posição dos pivôs do primeiro hiperplano no espaço projetado O'_i, O'_j : Posição dos pivôs do segundo hiperplano no espaço projetado $d(P, Q)$: Dissimilaridade usada para comparação dos pontos P e Q **Saída:** V' : Projeção do ponto**1 para** $d = 1$ **até** k **faça****2 se** $d=1$ **então****3** $dist_{O_a-V} \leftarrow d(O_a, V)$ **4** $dist_{O_a-O_b} \leftarrow d(O_a, O_b)$ **5** $dist_{O_b-V} \leftarrow d(O_b, V)$ **6 senão****7** $dist_{O_a-V} \leftarrow \sqrt{|(d(O_a, V))^2 - (O'_{a_{d-1}} - V'_{d-1})^2|}$ **8** $dist_{O_a-O_b} \leftarrow \sqrt{|(d(O_a, O_b))^2 - (O'_{a_{d-1}} - O'_{b_{d-1}})^2|}$ **9** $dist_{O_b-V} \leftarrow \sqrt{|(d(O_b, V))^2 - (O'_{b_{d-1}} - V'_{d-1})^2|}$ **10 fim****11** $V'_d \leftarrow \frac{(dist_{O_a-V})^2 + (dist_{O_a-O_b})^2 - (dist_{O_b-V})^2}{2dist_{O_a-O_b}}$ **12 fim****13 retorna** V'

(a) Diagrama de blocos da *Fastmap*. (b) Diagrama de blocos da etapa de pré-processamento da *Fastmap**.(c) Diagrama de blocos da etapa de projeção da *Fastmap**, sendo ela aplicada iterativamente ou em fluxo contínuo sobre todos os pontos.**Figura 4.3:** Diagramas de blocos da *Fastmap* (4.3(a)), o pré-processamento da *Fastmap** (4.3(b)) e a projeção da *Fastmap** (4.3(c)).

4.2.4 Complexidade Computacional

O custo computacional da *Fastmap** continua sendo linear, comparável com a *Fastmap* original. O diferencial está no tempo de execução. Uma vez que a busca dos pivôs é executado somente uma vez como uma etapa de pré-processamento, a *Fastmap** é 3 vezes mais rápida do que a *Fastmap*. Adicionalmente, como a *Fastmap** tem a característica de projetar um vetor por vez, ela pode ser paralelizada em CPU, chegando a executar 11 vezes mais rápido do que a *Fastmap* original.

Na Seção a seguir apresentamos a nossa segunda proposta para projeção de dados volumétricos variantes no tempo.

4.3 Recursive Laplacian-based Neiboorhood Projection

Na literatura encontramos somente uma técnica de projeção multidimensional que preserva a coerência temporal do conjunto de dados, que é a *PLMP* (Paulovich et al., 2010b). Essa técnica e a *Fastmap** possuem características e limitações similares. Dentre essas limitações podemos identificar duas principais: produzem projeções suavizadas e são não adequadas para dados com alta dimensionalidade.

A primeira limitação pode prejudicar o processo de exploração uma vez que mistura agrupamentos similares dentro da projeção. A segunda limitação não permite que conjuntos de dados com muitos atributos derivados possam ser explorados com projeções multidimensionais.

Dadas essas duas limitações, adaptamos a técnica *LSP* usando uma abordagem similar à proposta por Paulovich et al. (2011), permitindo assim a sua aplicação em conjuntos volumétricos variantes no tempo.

4.3.1 Abordagem proposta

Como foi visto na Seção 3.3.2, a técnica *LSP* busca preservar relações de vizinhança entre objetos multidimensionais no espaço projetado, ao contrário de preservar relações de distância como a *Fastmap*. Assim, um conjunto de pontos similares no espaço de características será posicionado na mesma vizinhança no plano de projeção.

A *LSP* se resume em duas etapas principais: na primeira etapa, um conjunto de pontos chamados “pontos de controle” são selecionados e projetados usando relações de distância, para depois, na segunda etapa, projetar os pontos restantes por meio da solução de um sistema linear baseado em relações de vizinhança no espaço original. Uma limitação dessa técnica é o reduzido número de pontos que podem ser incluídos no sistema linear.

Já a *PLP* aborda a limitação do número de pontos que podem ser projetados adicionando um passo chamado de *sampling* aos dois passos anteriores. Com esse novo passo o algoritmo passa a dividir o problema em pequenos subproblemas onde a *LSP* pode ser aplicada. O conjunto das subprojeções resultantes vão ser corretamente mapeadas, uma vez que os pontos de controle de cada amostra vão ser projetados juntos usando uma abordagem *MDS* global.

Essa subdivisão do conjunto em pequenas amostras permite a sua aplicação em conjuntos maiores como os dados volumétricos, que são de interesse nesse trabalho de mestrado. Uma limitação é o fato da *PLP*, do mesmo modo que a *Fastmap*, não preservar a coerência temporal dos dados variantes no tempo.

A nossa proposta é aplicar recursivamente a idéia de criar pequenos agrupamentos dos dados baseados em vizinhança no espaço original, para assim poder abranger todos os instantes de tempo do conjunto. Adicionalmente definimos os conceitos de *pontos de controle locais* e *pontos de controle temporais*, que são respectivamente, os pontos de controle de cada agrupamento encon-

trado sendo reconhecidos como locais quando descrevem pontos de um instante de tempo único, e temporais quando descrevem pontos de vários instantes de tempo simultaneamente.

Dada a interpretação recursiva que podemos observar na Figura 4.4, denominamos a nossa nova abordagem *Recursive Laplacian-based Neiboorhood Projection* ou simplesmente *RLNP* e a dividimos em duas etapas principais, sendo a primeira delas o pré-processamento e a outra a solução do sistema linear, as quais serão descritas nas Subseções 4.3.2 e 4.3.3, respectivamente. Adicionalmente descreveremos, na Subseção 4.3.4, a redução dos dados que pode ser usada em alguns conjuntos com o objetivo de acelerar a projeção de novos instantes de tempo em sequência.

Figura 4.4: Execução das recursões da *RLNP*. A primeira recursão é similar à executada pela *PLP* onde extraímos pontos de controle locais, na segunda recursão são extraídas relações de vizinhança dos pontos de controle locais e são identificados os pontos de controle temporais, sendo estes finalmente projetados usando uma técnica *MDS*. As regiões (a) e (b) indicam onde são usadas técnicas de *MDS* na *PLP* e *RLNP*, respectivamente.

4.3.2 Pré-processamento

A *RLNP* requer que todos os agrupamentos e os pontos de controle locais estejam previamente calculados. Por isso, nessa etapa processamos iterativamente todos os instantes de tempo salvando arquivos auxiliares que descrevem os agrupamentos e pontos de controle encontrados.

Para encontrar os agrupamentos usamos o algoritmo *bisecting k-means* (Steinbach et al., 2000). O algoritmo funciona dividindo os objetos multidimensionais em hiper-esferas. Desse modo, cada objeto se encontra mais próximo ao centro da hiper-esfera a qual pertence do que aos centros das outras hiper-esferas. Essa característica é benéfica para uma projeção baseada em vizinhança como a *LSP*, *PLP* e a nossa abordagem *RLNP*. No final, cada hiper-esfera é considerada um agrupamento e os pontos de controle são os pontos mais próximos ao centroide da hiper-esfera ou agrupamento.

Esse processo é aplicado tanto na *LSP* quanto na *PLP*. No primeiro caso os pontos de controle são projetados usando um método *MDS* (Cox e Cox, 2000) e no segundo caso é executado um mecanismo iterativo de projeção e reposicionamento dos pontos de controle por parte do usuário. A seguir descrevemos como usamos os agrupamentos e pontos de controle no processo recursivo da *RLNP*.

(a) Primeira recursão e pontos de controle *locais*

A primeira recursão da abordagem proposta consiste em agrupar os pontos de cada instante de tempo. Nessa recursão, os pontos de controle são denominados “locais” dado que servem como guia para a projeção do restante dos pontos em um instante de tempo arbitrário.

Uma diferença com as técnicas anteriores é que nessa etapa não é realizada a projeção dos pontos de controle. Esses pontos serão projetados na segunda recursão com o intuito de preservar a coerência temporal do conjunto de dados.

(b) Segunda recursão e pontos de controle *temporais*

Na segunda recursão, os pontos a serem projetados são os pontos de controle locais de todos os instantes de tempo em conjunto, para isso, aplicamos a mesma idéia da primeira recursão criando agrupamentos e selecionando novos pontos de controle, os quais denominamos “temporais”, uma vez que esses objetos garantem a coerência temporal nas projeções geradas pela *RLNP*.

Uma vez selecionados os novos pontos de controle temporais, eles são projetados em conjunto usando uma técnica *MDS* para serem retornados aos agrupamentos iniciais de cada instante de tempo, sendo agora considerados como pontos de controle temporais projetados.

4.3.3 Solução do sistema linear

Uma vez que sabemos como executar as recursões, projetamos os pontos por meio da construção de um sistema Laplaciano similar ao da *LSP*, usando a mesma abordagem usada na *PLP* para resolver um sistema linear por agrupamento.

Uma vantagem da nossa técnica é que nem todos os agrupamentos vão ser projetados ao mesmo tempo. Agrupamentos somente são projetados quando for necessário projetar o instante de tempo ao qual pertencem.

4.3.4 Codificação incremental

Uma limitação de aplicar várias vezes o algoritmo *bisecting k-means* é o custo computacional alto¹. Parte da nossa abordagem é realizar uma redução dos pontos que não sofrem muita variação de um instante de tempo para outro. Assim, podemos aproveitar os pontos “estáticos” previamente projetados no instante de tempo anterior (D_{ant}) no atual (D_{atu}).

Essa codificação consiste na definição de um limiar l , onde os pontos com uma variação menor ao valor de l serão considerados estáticos; caso contrário, os pontos serão reprojados. Para determinar essa variação será usada a mesma função de distância d com que os dados estão sendo comparados no *clustering* e na projeção.

Os pontos que sejam estáticos dado o limiar l serão considerados como o subconjunto $D_{atu} \cap D_{ant}$, sendo os outros parte do subconjunto $D_{atu} - D_{ant}$. Assim, dada uma exploração sequencial do conjunto de dados completo onde temos a projeção dos dados de D_i em P_i , para atualizar a projeção P_i para P_{i+1} somente precisamos reprojeter os pontos do subconjunto $D_{i+1} - D_i$, reduzindo assim o tempo de projeção da *RLNP*.

4.3.5 Complexidade Computacional

A *RLNP* é composta por duas etapas principais: pré-processamento e projeção, sendo o pré-processamento executado somente uma vez sobre o conjunto de dados inteiro e a projeção é calculada uma vez por instante de tempo.

A etapa de pré-processamento possui uma complexidade computacional composta de $O(t * aL + aT + MDS)$, onde t é o número de instantes de tempo, aL e aT são as complexidades de criar os agrupamentos nos instantes de tempo locais e temporais, e MDS é a complexidade da técnica escolhida para projetar os pontos de controle temporais. O custo computacional de calcular os agrupamentos é de $n\sqrt{n}$, sendo n o número de objetos por instante de tempo. Assim a complexidade da etapa de pré-processamento é $O(t * n\sqrt{n} + n\sqrt{n} + MDS) = O(tn\sqrt{n})$. Na maioria dos nossos testes usamos a *Fastmap* como técnica *MDS* para projetar os pontos de controle temporais. Nesse caso a complexidade computacional é $O(tn\sqrt{n} + n) = O(tn\sqrt{n})$.

Como o conjunto de dados foi pré-processado previamente, a etapa da projeção tem complexidade computacional de $O(n\sqrt{n})$. (ver Seção 3.3.2 e 3.3.3).

4.4 Resultados e Avaliação das Técnicas propostas

Para avaliar as abordagens propostas nesse capítulo, identificamos quatro propriedades que são interessantes no trabalho com projeções de volumes multidimensionais variantes no tempo. Essas propriedades são: tempo de processamento, *stress*, *layout* gerado e suporte para alta dimensionalidade.

¹ $O(nc)$, sendo n o número de elementos e c o número de agrupamentos desejados, considerando $c = \sqrt{n}$.

A importância do tempo de processamento de uma técnica de projeção multidimensional depende diretamente da aplicação onde vai ser usada. Em algumas aplicações onde o usuário gera diferentes projeções usando diferentes atributos, o tempo de resposta precisa ser muito baixo mas, se o usuário somente vai carregar e visualizar os dados já projetados, a técnica não precisa ter um tempo de resposta muito baixo.

A segunda propriedade é o *stress* que mede quão efetiva foi a projeção em termos de preservação da dissimilaridade entre os pontos, tomando como referência os dados originais e a projeção resultante. O valor do *stress* encontra-se em uma faixa de valores de $[0, 1]$, onde o valor do *stress* 0 significa que a projeção conseguiu preservar muito bem as dissimilaridades, enquanto um valor de 1 quer dizer que a projeção não as preservou. A medida de *stress* vem se tornando padrão para comparar diferentes tipos de projeções. Nessa seção vemos que a terceira propriedade, o *layout* gerado pela projeção, não mantém relação direta com o valor do *stress*. Às vezes o *layout* gerado ajuda a identificar melhor as estruturas internas do volume explorado, mesmo tendo obtido um *stress* alto.

Por último, consideramos a “alta dimensionalidade” do conjunto de dados como uma última propriedade para comparar as nossas abordagens, principalmente a *RLNP* que, sendo derivada da *LSP* e da *PLP* possui a capacidade de projetar conjuntos de dados com muitas dimensões com um valor de *stress* menor do que as anteriores.

A seguir, descrevemos os resultados obtidos nesse capítulo. Nos nossos testes usamos as projeções que mantêm a coerência temporal do conjunto de dados (*PLMP*, *Fastmap** e *RLNP*). Queremos destacar que avaliamos as três primeiras propriedades das nossas abordagens usando os conjuntos de dados descritos na seção 3.2. A última propriedade é explorada com outros conjuntos de dados uma vez que não encontramos conjuntos de dados da área de visualização volumétrica com alta dimensionalidade.

4.4.1 Tempo

Na Tabela 4.1 detalhamos os tempos de pré-processamento e projeção de um instante de tempo usando as abordagens propostas comparadas com as técnicas existentes (*PLMP*, *Fastmap* e *PLP*). Foram usados os três conjuntos de dados descritos na Seção 3.2 escolhendo arbitrariamente os instantes de tempo 20 para o *VisContest 2004*, 90 para o *VisContest 2008* e 01 no caso do *VisContest 2011*. Nesse caso os melhores resultados em incremento da velocidade de projeção foram obtidos com o conjunto do *VisContest 2004*.

Os testes foram executados em uma máquina com processador Intel® Core™2 Quad CPU Q9650 com quatro processadores com 3.00GHz de frequência cada um e 8GB de memória RAM.

Comparamos a *Fastmap* com a implementação sequencial da *Fastmap** e com uma implementação paralelizada em CPU da *Fastmap** (mCPU *Fastmap** na tabela) e podemos ver que além de manter a coerência temporal do conjunto de dados a *Fastmap** chega a ser quase 6 vezes mais

Técnica	VisContest 2004		VisContest 2008		VisContest 2011	
	$pre(t)$	$proj(t)$	$pre(t)$	$proj(t)$	$pre(t)$	$proj(t)$
<i>PLMP</i>	0,58s	0,296s	0,812s	0,064s	0,041s	0,014s
<i>Fastmap</i>		1,926s		1,488s		0,194s
<i>Fastmap*</i>	1.691s	0.327s	1.387s	0.395s	0.177s	0.043s
mCPU <i>Fastmap*</i>	1.589s	0.13s	1.457s	0.176s	0.190s	0.027s
<i>PLP</i>		56.13s		301.84s		3.503s
<i>RLNP</i>	12.575s	13.205s	29.272s	220.18s	18.037s	12.153s

Tabela 4.1: Tempos de execução das técnicas originais e modificadas. $pre(t)$ é o tempo de pré-processamento e $proj(t)$ é o tempo que demora para completar a execução da projeção. As técnicas originais não precisam o tempo de pré-processamento. mCPU *Fastmap** é a *Fastmap** paralelizada em CPU.

rápida que a *Fastmap* original, enquanto a implementação paralelizada em CPU chegou ser quase 15 vezes mais rápida do que a *Fastmap* original.

Uma segunda comparação que podemos observar na tabela é o ganho em velocidade de projeção da *RLNP* comparada com a *PLP*, uma vez que são as projeções com a capacidade de projetar dados volumétricos baseados em vizinhança. O tempo de projeção da *RLNP* chegou ser quase 7 vezes mais rápido do que o tempo de projeção da *PLP*.

Na comparação das nossas abordagens com a *PLMP* podemos observar que a versão paralelizada em CPU da *RLNP* chegou a ser duas vezes mais rápida no conjunto de dados do *VisContest 2004*. Mas em geral, a *PLMP* demonstrou ser uma técnica mais rápida tanto no pré-processamento, quanto no tempo de projeção.

Finalmente, podemos observar que os tempos de execução da *PLP* e da *RLNP* são muito altos com o conjunto de dados do *VisContest 2008*, não sendo possível a aplicação dessas técnicas em ferramentas interativas.

4.4.2 Stress

Um dos objetivos das projeções multidimensionais é a redução da diferença dentre a distância dos pontos no espaço original e no espaço projetado. Para esse objetivo usa-se a métrica do *stress* (Ingram et al., 2009) que está na Equação 4.1:

$$stress = \sqrt{\frac{\sum_{ij} (d_{ij} - d'_{ij})^2}{\sum_{ij} d_{ij}^2}} \quad (4.1)$$

onde, d é a distância usada no espaço original, d' a distância usada no espaço projetado, i o ID do objeto e j a dimensão do atributo.

Resumimos os resultados em termos de *stress* na Figura 4.5, onde avaliamos o valor obtido pelas projeções *Fastmap**, *RLNP* e *PLMP* projetando três instantes de tempo diferentes dos três conjuntos de dados usados nesse trabalho.

(a) Valor de *stress* ao longo do tempo usando o conjunto de dados “VisContest 2004”.

(b) Valor de *stress* ao longo do tempo usando o conjunto de dados “VisContest 2008”.

(c) Valor de *stress* ao longo do tempo usando o conjunto de dados “VisContest 2011”.

Figura 4.5: Valor de *stress* das projeções *Fastmap*, *Fastmap**, *PLP*, *RLNP* e *PLMP* usando diferentes conjuntos de dados.

Na Figura 4.5(a) estão os resultados da projeção do conjunto de dados do “VisContest 2004” onde observamos que os resultados da *Fastmap** e a *RLNP* são muito próximos de 0 enquanto os valores da *PLMP* são próximos a 0,4.

Observamos na Figura 4.5(b), que os melhores resultados do conjunto de dados “VisContest 2008” em termos de *stress* da projeção foram obtidos pela *Fastmap**, seguido pela *PLMP* e finalmente a *RLNP*. Observamos também que os resultados são muito similares e a diferença dentre os valores máximo e mínimo obtidos estão em torno a 0,015.

Por outro lado, a Figura 4.5(c) mostra os resultados das projeções dos dados do “VisContest 2011” que está composto por dados de maior quantidade dimensões. Observamos que a projeção que atingiu melhores resultados foi a *RLNP* e que a *Fastmap**, junto com a *PLMP*, obtiveram resultados similares. Esse resultado sugere que a *RLNP* pode conseguir melhores resultados em dados de alta dimensionalidade.

Resumimos os dados das Figuras 4.5(a), 4.5(b) e 4.5(c) na Tabela 4.2. Podemos observar que os resultados das técnicas propostas são comparáveis com os obtidos pela *PLMP*, sendo melhores em alguns casos.

Conjunto de Dados	<i>Fastmap*</i>	<i>RLNP</i>	<i>PLMP</i>
VisContest 2004 (t=12)	$3,93E - 09$	0,016490323	0,3453625
VisContest 2004 (t=24)	$4,40E - 09$	0,011871725	0,40194863
VisContest 2004 (t=36)	$5,62E - 09$	0,014642373	0,3356113
VisContest 2008 (t=40)	0,004308424	0,007838739	0,007070283
VisContest 2008 (t=100)	0,005870595	0,017687509	0,013870152
VisContest 2008 (t=160)	0,009910159	0,020783341	0,018278432
VisContest 2011 (t=8)	0,5842294	0,36870223	0,598145
VisContest 2011 (t=15)	0,60382056	0,35293133	0,67759186
VisContest 2011 (t=22)	0,56639427	0,29616678	0,50004995

Tabela 4.2: *Stress* obtido pelas projeções dos conjuntos de dados do VisContest usando as técnicas *Fastmap**, *RLNP* e *PLMP*.

4.4.3 Layout gerado

Na figura 4.6 apresentamos as projeções geradas pela *PLMP*, *Fastmap** e *RLNP* utilizando os conjuntos de dados da Seção 3.2. Podemos observar que as projeções produzidas pelas técnicas *PLMP* e *Fastmap** são mais suavizadas quando comparadas com as projeções produzidas pela *RLNP*. Isso acontece porque a *RLNP* projeta os pontos tentando preservar relações de vizinhança, em consequência, podemos observar melhor alguns agrupamentos nos dados projetados.

A seguir analisamos os *layouts* de cada conjunto de dados separadamente.

No caso do conjunto de dados do VisContest 2004, escolhemos a projeção produzida pela *RLNP* por ser uma projeção melhor agrupada e por não ser muito suavizada. Na Figura 4.7 mostramos os resultados da análise visual feita no espaço de atributos (Figura 4.7(a)) e o resultado da visualização no espaço do objeto (Figura 4.7(b)). No espaço de atributos entendemos que os dados projetados estão distribuídos em duas direções principais representadas pelas linhas vermelha e laranja. A linha vermelha denota o sentido da altura das estruturas. Para entender essa relação (*altura-estrutura*), colorimos a linha vermelha de modo que a altura da estrutura é diretamente proporcional à intensidade da cor da linha vermelha. Por último, a linha laranja denota a distância da estrutura em relação ao centro do furacão. Nesse caso, a intensidade da linha laranja indica quão perto do centro está localizada cada estrutura.

Figura 4.6: *Layouts* produzidos pelas técnicas *PLMP*, *Fastmap** e *RLNP* quando usadas para projetar os instantes de tempo 20 do *VisContest 2004*, 90 do *VisContest 2008* e 01 do *VisContest 2011*.

Para o segundo conjunto de dados (*VisContest 2008*), selecionamos a *Fastmap** como técnica de projeção, isso porque os *layouts* gerados pelas três projeções são muito parecidos. Na Figura 4.8 podemos observar 6 agrupamentos selecionados com estruturas diferentes no instante de tempo 90. Para visualizar cada um dos agrupamentos, realizamos a extração da superfície e visualizamos usando a técnica chamada *Surface Rendering* (Tiede et al., 1990). O único objeto que não foi visualizado é o *background*, por ser o complemento dos pontos da explosão. O *background* é selecionado na região delimitada pela cor preta.

Do mesmo modo que no segundo conjunto de dados, escolhemos a o resultado da *RLNP* no conjunto do *VisContest 2011*, uma vez que os dados são melhor distribuídos no plano da projeção e que nessa projeção foi possível encontrar os vórtices produzidos na simulação.

(a) Projeção *RLNP* do instante de tempo 20 do conjunto de dados *VisContest 2004* com os agrupamentos selecionados. (b) Visualização do espaço do objeto dos agrupamentos selecionados na projeção.

Figura 4.7: Layout da projeção produzida pela *RLNP*. As áreas limitadas pelos círculos amarelo e vermelho mostram um exemplo de estruturas encontradas no interior do furacão.

Figura 4.8: Seleção de 6 agrupamentos na projeção *Fastmap** do conjunto de dados do *VisContest 2008* junto com a visualização no espaço de objeto de 5 deles.

A Figura 4.9 mostra uma seleção na projeção das *streamlines* (Figura 4.9(a)) junto com os vórtices visualizados no espaço do objeto (Figura 4.9(b)).

(a) Projeção das *streamlines* com a seleção dos pontos que representam os vórtices. (b) Visualização no espaço do objeto das *streamlines* com os vórtices realçados.

Figura 4.9: *Layout* da projeção *RLNP* do conjunto de dados do *VisContest 2011*.

4.4.4 Alta dimensionalidade

Investigamos a qualidade dos *layouts* gerados com a *RLNP* e a *Fastmap** em função da dimensionalidade. Para isso empregamos uma coleção de texto composta por 574 artigos científicos (com título, resumo e referências) de três áreas diferentes: Raciocínio Baseado em Casos (do inglês *Case Based Reasoning* - CBR), Programação Lógica Indutiva (do inglês *Inductive Logic Programming* - ILP), e Recuperação de informação (do inglês *Information Retrieval* - IR).

Os artigos de CBR e ILP foram obtidos manualmente das séries *Lecture Notes on Artificial Intelligence* (LNAI), e os da área de ILP foram obtidos da Internet. Para a nossa análise, foram geradas várias representações ‘bag-of-words’ do conjunto de dados variando o número de características, gerando vários conjuntos de dados de 118, 144, 283, 307, 641, 2.548, 2.591, 3.000, 18.694 e 22.313 dimensões. É importante lembrar que representações com mais de 600 características definem espaços de dados muito esparsos. Nesses casos, a *PLMP* não pode ser usada, dado que o número de dimensões é maior ao número de elementos (Paulovich et al., 2010b).

A Figura 4.10 mostra os valores *stress* calculados tanto para *Fastmap** quanto para *RLNP*. Podemos observar que os valores produzidos pela *RLNP*, comparados com os valores produzidos pela *Fastmap** são menores e aumentam mais lentamente enquanto a dimensionalidade aumenta.

As novas técnicas obtiveram bons resultados quando foram aplicadas a conjuntos de dados de altas dimensionalidades. Acreditamos que essa característica, junto com o suporte da variação dos dados ao longo do tempo, fazem que as técnicas propostas criem novas possibilidades de estudos de dados científicos.

Figura 4.10: Comparação dos valores de stress dos *layouts* produzidos pelas *Fastmap** e *RLNP* em função à dimensionalidade dos dados.

4.5 Considerações finais

Nesse capítulo apresentamos duas novas abordagens para projetar vários instantes de tempo de volumes multidimensionais visando manter a coerência temporal nos *layouts* finais. As abordagens estão baseadas na hipótese proposta no início do capítulo que consiste na fixação ou projeção em conjunto dos objetos representativos de cada instante de tempo para tornar estáveis os *layouts* produzidos por meio da projeção de diferentes instantes de tempo.

Segundo essa hipótese, fixamos os pivôs da técnica *Fastmap* por meio da separação em duas etapas principais. A primeira delas é a seleção dos objetos mais distantes denominados pivôs, processo que não precisa ser aplicado a todos os instantes de tempo como passo prévio à etapa de projeção. Porém, ele pode ser aplicado a um subconjunto de dados arbitrário que, nos nossos testes, foi o instante de tempo central. A segunda etapa é a aplicação da *lei dos cossenos* de uma forma iterativa para encontrar as posições no espaço projetado. Tanto a divisão da técnica em duas etapas quanto a implementação iterativa, diminuem o tempo de processamento e facilitam a paralelização do algoritmo, aumentando a velocidade da técnica até em 15 vezes.

Baseados na mesma hipótese foi proposta uma segunda abordagem de projeção baseada em vizinhança. Adaptamos a técnica *LSP* aplicando uma estratégia similar à apresentada por [Paulovich et al. \(2011\)](#), onde aplicamos recursivamente a idéia de criar agrupamentos e selecionar pontos de controle locais em cada instante de tempo (primeira recursão) e pontos de controle temporais na segunda recursão. Produzindo assim uma técnica de projeção multidimensional temporal que, ao contrário da *Fastmap** e da *PLMP*, busca preservar as relações de vizinhança.

Testamos as novas abordagens segundo 4 propriedades. Essas propriedades demonstraram a importância de ter várias opções de projeções multidimensionais que preservem a coerência temporal dos conjuntos de dados uma vez que nem sempre o *layout* produzido pela projeção é a mais útil para o problema estudado.

Finalmente, realizamos um estudo sobre o desempenho das projeções multidimensionais temporais quando lidam com espaços esparsos ou de alta dimensionalidade. Nesse sentido, observamos que a *RLNP* herda as propriedades da *LSP*, convertendo a *RLNP* em uma técnica promissora para lidar com conjuntos de dados temporais de muitas dimensões.

Percebemos que as projeções de dados volumétricos multivariados não sempre facilitam a análise de agrupamentos por possuir características bastante similares. Outra limitação encontrada é a perda de controle dos valores dos atributos, os quais podem ser necessários na exploração do volume. A seguir apresentamos o nosso estudo dessas limitações e as nossas abordagens propostas.

Melhorando a exploração através de projeções

5.1 Considerações iniciais

No capítulo 4 propomos duas técnicas de projeção multidimensional que, ao contrário da maioria das técnicas, conseguem manter a coerência temporal dos dados ao longo de vários instantes de tempo. Isso torna possível a exploração de volumes variantes no tempo baseada na análise no espaço de características.

Contudo, identificamos alguns problemas que dificultam a exploração do conjunto de dados quando usamos técnicas de projeção multidimensional. O primeiro problema é que, dada a natureza contínua dos dados obtidos por simulações de fenômenos naturais, as projeções resultantes se apresentam suavizadas entre agrupamentos, diminuindo assim a possibilidade de separação de vários agrupamentos na projeção. O segundo problema identificado é que, quando exploramos o espaço de características através de projeções multidimensionais, perdemos a habilidade de trabalhar com alguns atributos específicos. Um exemplo recorrente nos conjuntos de dados usados é a necessidade de identificar regiões no volume com maior concentração de uma determinada grandeza.

Nesse capítulo propomos duas abordagens para explorar os dados volumétricos projetados considerando essas duas dificuldades. A primeira abordagem é descrita na Seção 5.2 e explora a separação de agrupamentos escondidos nas projeções, com base em técnicas de seleção de atributos e reprojeção de subconjuntos de dados. A segunda abordagem proposta é detalhada na Seção 5.3 e

busca dar mais expressividade à projeção quando a análise dos atributos é importante no processo de exploração.

5.2 *Cluttering* de agrupamentos

Explorado na Seção 4.4.3, o *layout* gerado por técnicas de projeção multidimensional baseadas em distância apresentam um *layout* mais suavizado quando comparadas com técnicas baseadas em vizinhança, mesmo assim, essas últimas não resolvem o problema de separar efetivamente os agrupamentos. Podemos observar um exemplo desse problema na Figura 5.1 onde foram identificadas várias estruturas internas no volume posicionadas muito próximas umas das outras. Um detalhe importante que gerou alguma confusão em outros trabalhos (Paulovich et al., 2010b) é o fato de existirem três estruturas no volume (delimitadas nas cores amarela, azul e laranja) que estão posicionadas bem próximas ao *background* (delimitada com a cor preta). Quando essa região é selecionada e visualizada usando técnicas de extração de superfícies, pode ser interpretada de maneira errônea como o frente da explosão, sendo essa uma característica de bastante interesse para resolver as questões sobre o conjunto de dados.

Figura 5.1: Várias estruturas internas do volume são identificadas no espaço de atributos. Essas estruturas são agrupadas em regiões muito próximas limitando a descoberta de novas regiões de interesse.

Em geral, verificamos que uma projeção de um conjunto de dados de característica multi-dimensional produz um *layout* suavizado, ou apresenta pontos espalhados no meio dos poucos

agrupamentos identificados visivelmente na projeção. Esse problema surge como consequência da natureza contínua dos dados por tratar-se da simulação de fenômenos naturais. Essa questão foi abordada parcialmente por [Guo et al. \(2011b\)](#), sendo aplicados pesos diferentes nos atributos para produzir projeções com *layouts* diferentes. Uma limitação dessa abordagem é que a capacidade de gerar melhores *layouts* recai na habilidade do usuário em identificar os pesos específicos para cada atributo.

Diante disso, na próxima seção descrevemos uma solução para o problema propondo uma nova abordagem de seleção de atributos relevantes e não relevantes visando gerar melhores *layouts* por meio da reprojeção de agrupamentos que poderiam conter vários subagrupamentos.

5.2.1 Descrição da proposta: Reprojeção de agrupamentos

Uma dificuldade de se trabalhar com dados multivariados é selecionar o conjunto de atributos que melhor descrevem um problema particular. Em alguns casos vários atributos são correlacionados virando assim pesos das funções de distância, sendo isso prejudicial para o resultado do *layout* final da projeção.

Uma abordagem para solucionar esse problema é o uso de algoritmos de seleção de atributos, visando escolher um subconjunto de características que melhor classifiquem um conjunto de objetos multivariados como uma classe unária.

Portanto, como o nosso objetivo é a redução da suavização entre agrupamentos para melhor exploração de estruturas contidas nos dados, baseamos a nossa abordagem na hipótese de que, descartando o subconjunto de características que melhor descrevam um agrupamento, reduzimos a coesão dos pontos produzindo um *layout* com agrupamentos mais separados uns dos outros.

Para comprovar a validade da hipótese, detalhamos a nossa abordagem na Figura 5.2, na qual selecionamos um subconjunto de pontos multivariados que poderiam conter vários agrupamentos de pontos muito próximos. Para tentar separar os subagrupamentos, aplicamos um algoritmo de seleção de atributos para reprojetar posteriormente os pontos, descartando os atributos não selecionados pelo algoritmo.

Encontramos, nos nossos experimentos, que diferentes algoritmos de seleção de atributos produzem diferentes conjuntos de atributos relevantes, sendo esses resultados nem sempre úteis para a separação dos subagrupamentos. Por isso, limitamos o nosso estudo às técnicas que demonstraram melhores resultados com os conjuntos de dados testados.

Uma vez selecionados os atributos relevantes, esses são descartados para reprojetar a seleção usando os atributos não relevantes, com o intuito de espalhar os pontos para explorar possíveis subagrupamentos. Como estamos trabalhando com dados volumétricos variantes no tempo, temos que assegurar a coerência temporal na reprojeção e, para isso, usamos a técnica *Fastmap** apresentada na Seção 4.2.

Figura 5.2: Passos da abordagem proposta para solucionar o problema da separação de subagrupamentos descartando os atributos selecionados por um algoritmo de seleção de atributos.

5.2.2 Seleção de atributos relevantes

O uso de algoritmos de seleção de subconjuntos de atributos é uma boa opção quando trabalhamos com dados da mesma classe. Embora possa parecer que a seleção de atributos promove a perda de informação, isso não acontece quando existem características irrelevantes no conjunto total de atributos (Xing e Karp, 2001). Essas características irrelevantes quase não contém informação útil para uma tarefa específica ou não descrevem bem os dados reduzindo a qualidade da classificação e do agrupamento.

Embora esses dados irrelevantes possam ser eliminados imediatamente usando sentido comum ou conhecimento básico do domínio, a seleção do melhor subconjunto de características requer frequentemente uma abordagem sistemática. A abordagem ideal para seleção de atributos é tentar todos os subconjuntos possíveis como entrada do *Algoritmo de Mineração de Dados* (AMD) e escolher o subconjunto que produz os melhores resultados. A vantagem desse método é produzir o melhor subconjunto de atributos que descrevem o problema. A questão está no fato de existirem

2^d subconjuntos possíveis para d atributos, empecilho para a abordagem. Para tentar solucionar esse problema existem três tipos de abordagens de seleção de atributos: as embutidas, as baseadas em filtros e os *wrappers*.

Abordagem embutida: Nessa abordagem a seleção de atributos acontece naturalmente como parte do AMD, onde o algoritmo decide quais atributos vão ser usados e quais vão ser ignorados;

Abordagem filtro: As características são selecionadas antes da execução do AMD usando alguma abordagem independente da tarefa de mineração de dados;

Abordagem *Wrappers*: Esses métodos usam o AMD como uma caixa preta para encontrar o melhor subconjunto de atributos de uma forma muito similar à abordagem ideal, e sem avaliar todos os subconjuntos possíveis.

Como nesse trabalho focamos na visualização dos agrupamentos por meio de técnicas de projeção multidimensional, o nosso AMD será a visualização no espaço de características. Portanto, limitaremos o nosso estudo à aplicação das abordagens baseadas em filtro e *wrappers*, descartando as técnicas embutidas, uma vez que são específicas para o AMD selecionado.

Na figura 5.3 podemos distinguir facilmente a diferença entre as abordagens filtro e *wrapper*. A primeira abordagem (Figura 5.3(a)) é comparada com um pré-processamento nos dados, sendo as características agrupadas conforme propriedades que derivam em boas características, como ortogonalidade e alto conteúdo de informação (John et al., 1994), para agrupá-las, algumas técnicas baseadas nessa abordagem usam um algoritmo classificador para estimar o mérito de um conjunto de atributos. A vantagem dessa abordagem é a velocidade, porém os resultados nem sempre são os melhores, uma vez que as características são selecionadas independentemente do AMD.

Por outro lado, a abordagem *wrapper* da Figura 5.3(b) trata a questão de encontrar um subconjunto de características como um problema de busca, percorrendo o espaço de soluções utilizando busca exaustiva (largura ou profundidade) ou heurísticas não determinísticas (procuram o máximo global). A maior limitação de usar a abordagem *wrapper* é o alto custo computacional exigido, dado que um algoritmo classificador é executado para cada solução candidata. Para resolver esse problema reduzimos a quantidade dos pontos executando um *sampling* randômico dos pontos selecionados antes do passo de seleção de atributos.

Para a etapa de seleção de atributos usamos a API (do inglês *Application Programming Interface*) do WEKA (Hall et al., 2009), a qual incorpora todos os algoritmos necessários para realizar essa tarefa.

(a) Abordagem filtro. A avaliação do subconjunto gerado é feita somente no final do processo.

(b) Abordagem wrapper. Utiliza-se um algoritmo de classificação para avaliar o subconjunto de atributos selecionados.

Figura 5.3: Esquema das abordagens filtro (Figura 5.3(a)) e wrapper (Figura 5.3(b)).
Extraído de John et al. (1994).

5.2.3 Exemplos e Resultados

Nessa subseção apresentamos dois testes realizados dentro da reprojeção de agrupamentos. Para esses testes foram usados os dados da explosão, nos quais podemos encontrar várias estruturas internas escondidas pelo *cluttering* dos agrupamentos projetados (Figura 5.1).

Para esses experimentos foram usadas as classes do Weka *CfsSubsetEval* e *ClassifierSubsetEval* baseadas na abordagem filtro e a *WrapperSubsetEval* baseada na abordagem wrapper. A classe *CfsSubsetEval* analisa o valor de um subconjunto de atributos junto com o grau de redundância entre eles. A *ClassifierSubsetEval* avalia subconjuntos de atributos de dados de treinamento utilizando um classificador para estimar o mérito de um conjunto de atributos. Os classificadores usados foram *GaussianProcesses*, *IsotonicRegression*, *LeastMedSq*, *MultilayerPerceptron*, *RBFNetwork*, *LinearNNSearch* e *KStar*. Por último, o *WrapperSubsetEval* avalia conjuntos de atributos usando um esquema de aprendizagem. Através da validação cruzada para estimar a precisão do sistema de aprendizagem para um conjunto de atributos.

Em cada um dos experimentos, separamos os resultados em dois grupos: melhores e não satisfatórios. Cada resultado foi avaliado visualmente determinando como melhor resultado a reprojeção do subconjunto de dados que conseguiu gerar dois ou mais agrupamentos de pontos. Caso contrário, o resultado foi classificado como não satisfatório.

(a) Experimento 1

O objetivo do primeiro experimento é analisar um pequeno agrupamento na projeção buscando encontrar novos subagrupamentos. Na Figura 5.4 visualizamos o agrupamento selecionado em uma área vermelha (Figura 5.4(a)) e a visualização da seleção no espaço do objeto (Figura 5.4(b)).

(a) Seleção do agrupamento no espaço de características.

(b) Visualização dos pontos agrupados no espaço do objeto.

Figura 5.4: Experimento 1: Seleção e visualização de um pequeno agrupamento na projeção.

Os resultados da aplicação dos algoritmos de seleção de atributos estão nas Figuras 5.5 e 5.6. A Figura 5.5 mostra os melhores resultados obtidos aplicando-se as classes *CfsSubsetEval*, *WrapperSubsetEval* e *ClassifierSubsetEval* com os classificadores *LeastMedSq*, *MultilayerPerceptron* e *IBk*. Outros resultados não satisfatórios estão na Figura 5.6.

(a) Reprojeção dos pontos usando somente os melhores atributos obtidos.

(b) Reprojeção dos pontos descartando os melhores atributos selecionados pelo algoritmo.

Figura 5.5: Melhores resultados obtidos na reprojeção dos pontos do subagrupamento selecionado. As três abordagens de seleção de atributos conseguiram gerar um melhor *layout* na segunda projeção.

(a) Reprojeção com atributos relevantes obtidos usando o classificador *GaussianProcesses*. (b) Reprojeção descartando atributos relevantes.

(c) Reprojeção com atributos relevantes obtidos usando o classificador *IsotonicRegression*. (d) Reprojeção descartando atributos relevantes.

(e) Reprojeção com atributos relevantes obtidos usando o classificador *KStar*. (f) Reprojeção descartando atributos relevantes.

Figura 5.6: Resultados não satisfatórios obtidos pela seleção de atributos usando a classe *ClassifierSubsetEval*.

(b) Experimento 2

Nesse segundo agrupamento exploramos a nossa abordagem com um agrupamento que possui mais de uma estrutura interna. Para isso, reprojetaamos um área com quatro subagrupamentos, sendo essa área observada na Figura 5.7 a qual possui a projeção original e a visualização da seleção a ser reprojetaada.

Figura 5.7: Experimento 2: Seleção e visualização de um pequeno agrupamento na projeção.

Os melhores resultados são mostrados na Figura 5.8 e foram obtidos usando a classe *ClassifierSubsetEval*, junto com o classificador *IBk*. Podemos perceber na Figura 5.8(b) que a projeção somente possui dois agrupamentos principais, no qual um deles pertence a uma das estruturas internas esperadas. Outros exemplos não tão bem sucedidos estão na figura 5.9.

Figura 5.8: Melhores resultados obtidos na reprojeção dos pontos do subagrupamento selecionado. As três abordagens de seleção de atributos conseguiram gerar um melhor *layout* na segunda projeção.

Figura 5.9: Resultados não satisfatórios obtidos pela seleção de atributos usando as três abordagens.

Nesses experimentos podemos comprovar a nossa hipótese, na qual, descartando-se o subconjunto de características que melhor descrevam um agrupamento, reduzimos a coesão dos pontos produzindo um *layout* com agrupamentos mais separados uns dos outros. Consideramos importante mencionar que essa hipótese não deve funcionar em todos os casos, especialmente quando trabalhamos com poucas dimensões ou quando os atributos são altamente correlacionados. Outro ponto importante, é o fato de uma técnica de seleção de atributos nem sempre produzir os me-

lhores resultados. Por isso, aconselhamos usar técnicas com baixo custo computacional como as baseadas em filtros ou *wrappers* para poder executar várias opções e possibilitar ao usuário escolher a melhor opção dentre vários resultados obtidos.

A seguir apresentamos a nossa segunda abordagem que visa dar mais expressividade à projeção quando a análise dos atributos é importante no processo de exploração.

5.3 Análise de atributos específicos

Nessa seção falaremos sobre a dificuldade encontrada quando exploramos volumes multivariados procurando conhecer mais detalhes associados a atributos específicos.

Na literatura podemos encontrar vários trabalhos que abordam problemas da área da Visualização Científica e utilizam técnicas de Visualização de Informação para explorar o espaço de características sem perder informação específica dos atributos originais. Alguns exemplos das técnicas de Visualização de Informação utilizadas são Coordenadas Paralelas (Blaas e Post, 2008; Akiba e Ma, 2007; Guo et al., 2011b), *scatter plots* (Gresh et al., 2000; Doleisch et al., 2003; Blaas et al., 2007), multigrafos (Sauber et al., 2006), *clustering* hierárquico (Linsen et al., 2008), *star coordinates* (Kandogan, 2001) histogramas (Wang et al., 2011), dentre outras. Esses trabalhos foram detalhados na Seção 2.7. A principal desvantagem de usar essas técnicas é não simplificar a exploração baseada em similaridade no espaço multidimensional.

Dentre os trabalhos que aliam Visualização Científica e Visualização de Informação, alguns abordam a exploração de dados científicos usando técnicas de projeção multidimensional (ver Seção 3.4) em junção com técnicas de Visualização de Informação, como histograma (Chen et al., 2009), dendograma (Jianu et al., 2009) e coordenadas paralelas (Guo et al., 2011b). Para realizar essa junção entre as diferentes visualizações utiliza-se a coordenação (Eler, 2011), a qual consegue dar ao usuário várias ferramentas para explorar o mesmo problema, porém o usuário precisa conhecer e trabalhar ao mesmo tempo com vários paradigmas de visualização diferentes. A mesma abordagem de coordenação é utilizada nesse trabalho.

Uma solução existente para análise de atributos específicos quando trabalhamos com técnicas de projeção é o mapeamento do valor do atributo em cada ponto projetado usando alguma escala de cor. Assim, o usuário pode perceber onde se encontram os valores altos ou baixos de um atributo no conjunto de dados projetado. Porém, vários pontos são projetados na mesma posição no plano, podendo causar sobreposição e impedindo ao usuário identificar regiões de interesse. Observamos um exemplo dessa limitação na Figura 5.10, onde o atributo H_2 foi mapeado sobre os pontos projetados usando uma escala de cor (azul-verde-vermelho), que interpola os valores em cor entre o valor máximo (vermelho) e o mínimo (azul). A região com valores máximos foi selecionada e os atributos foram visualizados usando um *boxplot* (Frigge et al., 1989), no qual podemos perceber

na região selecionada a presença de pontos onde o valor do atributo H_2 não é máximo. É por isso que procuramos uma solução alternativa para facilitar a análise de atributos específicos sem perder os benefícios das técnicas de projeção multidimensional.

A seguir descreveremos a nossa proposta para resolver o problema apresentado usando uma abordagem híbrida entre *scatter plots* baseados em atributo e projeções multidimensionais.

5.3.1 Descrição da proposta: *Scatter Projections*

Para conseguir uma visualização que simplifique a análise de atributos específicos sem perder a capacidade das projeções multidimensionais de explorar o espaço de atributos baseado na similaridade, propomos o *Scatter Projection* inspirado nos *scatter plots* baseados em atributos e nas projeções multidimensionais. O *Scatter Projection* está formado por um sistema de coordenadas em duas dimensões, onde mapeamos o atributo de interesse no eixo horizontal e uma projeção multidimensional (1D) no eixo vertical. A Figura 5.11 mostra o *Scatter Projection* como parte do sistema de coordenadas proposto.

Formalmente, podemos afirmar que um *Scatter Projection* S é representado por um conjunto de pontos $p(a, v)$, no qual a é o valor de um atributo específico, e v indica a vizinhança a qual pertence o ponto p . Voltando à Figura 5.11, observamos que os quatro pontos de cor azul possuem valores próximos de a e em v por encontrarem-se posicionados próximos uns dos outros, portanto, podemos concluir que os objetos multidimensionais representados nesses pontos são similares no espaço multidimensional e possuem valores baixos do atributo estudado. Um outro caso é representado pelos pontos vermelhos, os quais novamente são similares no espaço multidimensional mas possuem alta variação no atributo estudado. O último caso está representado pelos pontos laranja que provavelmente não são similares no espaço multidimensional mas possuem valores semelhantes (altos) do atributo estudado.

A vantagem dessa proposta é a criação de uma visualização de espaço de atributos na qual se consegue analisar dados por similaridade de acordo com uma magnitude específica.

Na seção seguinte falaremos sobre a utilização da *Fastmap** como técnica de redução da dimensionalidade para ser usada no eixo vertical do *Scatter Projection*.

5.3.2 *Fastmap** aplicado a *Scatter Projections*

Para explorar o espaço de atributos por similaridade no eixo vertical, propomos o uso da *Fastmap**, apresentada no capítulo 4. Uma característica importante da técnica é que o algoritmo da *Fastmap* é composto por recursões (técnica original) ou iterações (técnica proposta nesse trabalho). Na primeira iteração são utilizados dois pivôs interpretados como os objetos mais distantes no espaço d dimensional completo. Na segunda iteração, os pivôs são os objetos mais distantes no espaço $d - 1$ dimensional.

Figura 5.10: Análise da concentração de H_2 no instante 90 do conjunto de dados da explosão. A projeção do mesmo instante de tempo mostra duas seleções feitas por um filtro nos dados (Figura 5.10(a)) e pelo usuário (Figura 5.10(b)) junto com os respectivos *boxplots* (Figuras 5.10(c) e 5.10(d)) para visualizar melhor os valores dos pontos selecionados. As seleções tentam representar os pontos com maior concentração de H_2 . Observamos também que o mapeamento do atributo usando uma escala de cor (azul-verde-vermelho) não foi muito bem sucedida por causa do mapeamento.

Figura 5.11: Esquema de construção do Scatter Projection. O eixo horizontal mapeia o valor de um atributo, enquanto no eixo vertical são mapeados os objetos multidimensionais por similaridade.

Dado que a primeira iteração projeta os pontos baseado no espaço multidimensional completo, podemos aproveitar esses valores no eixo vertical do *Scatter Projection*. Além de poder explorar por similaridade usando os valores obtidos na primeira iteração da *Fastmap**, o *Scatter Projection* consegue manter a coerência temporal dos dados, sempre que o passo de pré-processamento ou seleção dos pivôs seja executado utilizando um conjunto de dados representativo (Subseção 4.2.2).

5.3.3 Exemplos e Resultados

O objetivo desse exemplo é comparar os resultados obtidos por [Linsen et al. \(2009\)](#) com os resultados obtidos pelas *Scatter Projections* tentando reproduzir a mesma experiência, que propõe uma visualização para mostrar em qual parte do volume o composto H_2 é predominante. É importante realçar o interesse no estudo do H_2 na criação de galáxias porque permite o colapso dos gases primordiais provocando a criação de mais estrelas. Os cientistas perceberam algum fenômeno impedindo a produção das estrelas novas e, a hipótese é que a turbulência poderia estar causando esse fenômeno. Porém, não existem visualizações que possam comprovar essa hipótese.

Na Figura 5.12 temos dois *Scatter Projections* construídos com base na densidade de H_2 (Figura 5.12(a)) e a turbulência (Figura 5.12(c)). A análise de atributos específicos começa procurando selecionar uma região ou faixa de valores no eixo do atributo. Como os dois atributos são quantidades, procuramos descartar os valores mínimos aplicando uma seleção (um limiar visual) buscando obter uma estrutura bem definida na visualização do espaço do objeto (Figuras 5.12(b) e 5.12(d)). A escolha de um limiar é um procedimento natural quando trabalhamos com atributos específicos. Uma vez explorado o eixo do atributo, analisamos a área selecionada no eixo da projeção e, percebemos que a seleção do composto H_2 está concentrada no meio do eixo, enquanto a seleção da turbulência está distribuída. Com isso, podemos afirmar que a região onde o com-

posto H_2 é mais predominante está possivelmente rodeada por várias regiões de turbulência com características diferentes.

(a) *Scatter Projection* do atributo H_2 .

(b) Visualização no espaço do objeto da seleção no *Scatter Projection*.

(c) *Scatter Projection* do atributo turbulência

(d) Visualização no espaço do objeto da seleção no *Scatter Projection*.

Figura 5.12: Dois *Scatter Projections* mostrando duas regiões diferentes para análise de atributos específicos.

Na Figura 5.13 comparamos o resultado obtido por Linsen et al. (2009) com o resultado obtido usando *Scatter Projections*. Observamos que os resultados são similares exceto por uma pequena região de H_2 por trás da explosão na Figura 5.13(a). Essa diferença pode ter sido causada pelo algoritmo de agrupamento usado, dado que essa estrutura pode ter características similares à região de H_2 .

Como observado, a vantagem de usar *Scatter Projections* para analisar dados é a capacidade de analisar diretamente a variável de interesse. No exemplo foram utilizados dois *Scatter Projections*

(a) Figura proposta por [Linsen et al. \(2009\)](#), que visa explicar a causa da turbulência (vermelho) se apoiando na densidade do H_2 . Para obter esses valores ele aplicou algoritmos de agrupamento de dados no instante 99.

(b) Visualização das maiores concentrações de turbulência (vermelho) e H_2 (verde) obtidos analisando o instante 99 usando os *Scatter Projections* da Figura 5.12.

Figura 5.13: Comparação dos resultados obtidos por [Linsen et al. \(2009\)](#) e os obtidos usando *Scatter Projections*. Observamos a turbulência encontra-se em volta da região onde o H_2 é mais predominante.

para explorar um problema específico, sem mudar de visualização baseada em projeção de pontos no plano.

A seguir concluímos esse capítulo com algumas considerações sobre o problema estudado e as abordagens propostas.

5.4 Considerações finais

Nesse capítulo apresentamos duas abordagens para melhor exploração de dados volumétricos através de técnicas de projeção multidimensional. Essas abordagens estão baseadas em dois problemas encontrados explorando o volume por meio das técnicas apresentadas no capítulo 4.

O primeiro problema foi denominado “*Cluttering* de agrupamentos”, porque existem vários agrupamentos posicionados muito próximos uns dos outros na projeção. Para tentar separar esses agrupamentos propomos a reprojeção de agrupamentos usando um subconjunto dos atributos do espaço original. Para isso utilizamos técnicas de seleção de atributos e descartamos o subconjunto das características que descrevem o agrupamento como uma classe unária.

A principal vantagem dessa abordagem está na capacidade de separar pontos em uma nova projeção, permitindo assim a melhor exploração das estruturas internas do volume. Porém, a técnica depende da qualidade da seleção de atributos. Observamos que as técnicas baseadas na

abordagem *wrapper* obtiveram melhores resultados por usar um algoritmo de classificação com um custo computacional maior que o das técnicas baseadas em filtros.

Além disso, abordamos um segundo problema que denominamos “Análise de atributos específicos” motivado pela necessidade de focar em características determinadas em algumas etapas da exploração dos dados. Percebermos que o mapeamento de um atributo na projeção usando escalas de cor nem sempre resulta em um guia visual por que existe uma alta sobreposição de pontos na projeção. A nossa proposta visa resolver essa limitação por meio de uma visualização baseada em *scatter plot* e projeção multidimensional. Essa abordagem diminui as limitações de explorar o volume usando projeções, sem perder a capacidade de direcionar a exploração focada em atributos específicos.

É importante esclarecer que, com as duas abordagens propostas nesse capítulo, continuamos explorando os dados volumétricos multidimensionais por meio de técnicas de projeção sem perder a coerência temporal dos dados.

Conclusões

Nesse trabalho propomos algumas técnicas e abordagens visando trabalhar com dados volumétricos multivariados, e também com a questão da variação no tempo. Conforme apresentamos e discutimos, a multidimensionalidade apresenta dificuldades considerando que algumas técnicas não conseguem lidar com espaços de alta dimensionalidade. O tempo também apresenta-se como um problema quando as técnicas não conseguem lidar com a variação dos dados. Visando solucionar tais questões propomos técnicas, baseadas nos conceitos de projeções multidimensionais, para análise de conjuntos volumétricos multivariados que variam com o tempo. As duas técnicas são resultado de uma hipótese proposta de que a fixação ou projeção em conjunto dos objetos representativos de cada instante de tempo torna estáveis os *layouts* produzidos por meio da projeção de diferentes instantes de tempo.

A primeira técnica de projeção para dados volumétricos variantes no tempo proposta foi a *Fastmap**, que está baseada na reformulação da *Fastmap*. A *Fastmap** é composta por duas etapas: pré-processamento e projeção. Essa divisão em duas etapas adiciona três características importantes. A primeira característica é a velocidade, na qual projeções de dados são geradas com um custo computacional menor. A segunda característica é o fluxo contínuo ou *streaming*. Uma vez que propusemos uma implementação iterativa sobre cada atributo do vetor, podemos projetar separadamente cada instância dos dados sem precisar guardar informação de instâncias anteriores. A terceira característica é a coerência temporal, sendo que as projeções de diferentes instantes de tempo conseguem manter a coerência dos dados se os dados preprocessados são representativos. Essa última característica é derivada da segunda. É importante lembrar que as projeções *PLMP* e as derivadas da *LSP* não são técnicas de projeção tradicionais, dado que elas tentam adaptar vários objetos multidimensionais a um *layout* produzido por uma outra técnica de projeção multidimensional. Sendo assim, a *Fastmap** é a única técnica de projeção tradicional com a capacidade de

projetar dados em fluxo contínuo. A principal vantagem identificada quando comparada com a *PLMP* é que a *Fastmap** consegue projetar espaços de alta dimensionalidade em fluxo contínuo.

Seguindo a linha da coerência temporal nos *layouts* das projeções produzidas, apresentamos uma segunda técnica de projeção para dados volumétricos multivariados variantes no tempo. Essa abordagem foi denominada *Recursive Laplacian-based Neiboorhood Projection* ou *RLNP* e projeta os dados por similaridade baseado na técnica *LSP*. Com a inclusão de uma hierarquia de recursões, conseguimos projetar conjuntos de dados maiores sempre mantendo a coerência temporal nos dados projetados. A principal vantagem dessa técnica é a capacidade de projetar espaços de alta dimensão com um nível de *stress* baixo. Uma limitação é a complexidade computacional, uma vez que a *RLNP* resolve várias matrizes Laplacianas e executa muitas vezes o algoritmo de agrupamento.

Nos nossos testes, percebemos que as técnicas de projeção ajudam a identificar estruturas internas do volume usando abordagens como *tentativa e erro* na exploração. Algumas dessas estruturas encontram-se posicionadas muito próximas na projeção, o que prejudica a identificação dos agrupamentos. Para facilitar uma melhor identificação e separação de estruturas do volume, propomos uma abordagem denominada *reprojeção de agrupamentos*, na qual usamos algoritmos de seleção de atributos para descartar as características que podem misturar os agrupamentos. Nos experimentos observamos que a separação de agrupamentos é possível com alguns algoritmos de seleção de atributos. Percebemos também que nem todos os algoritmos funcionam em todos os casos. Por isso, recomendamos a escolha de vários algoritmos diferentes baseados na abordagem filtro ou *wrapper* se o método de classificação possui uma complexidade computacional baixa.

Por último propomos uma abordagem baseada em projeções multidimensionais para a análise de atributos específicos, onde foi proposta uma nova visualização denominada *Scatter Projection* que facilita a exploração focando em um atributo específico e na similaridade dos dados entre eles. Com essa visualização o usuário consegue explorar os dados por similaridade sem perder o foco no atributo de importância.

As contribuições deste trabalho podem ser resumidas em duas técnicas de projeção multidimensional, sendo uma delas para projeção em fluxo contínuo e a outra para altas dimensões. Além disso também apresentamos uma nova abordagem para melhor identificação de atributos e uma nova visualização. Todas as técnicas e abordagens propostas nesse trabalho de mestrado servem para explorar grandes conjuntos de dados volumétricos multivariados, mantendo a coerência temporal.

Das técnicas apresentadas nesse trabalho, podem-se identificar alguns trabalhos futuros. A *Fastmap** projeta dados que estão dentro do hiperplano definido pelos pivôs. A técnica pode ser adaptada para projetar dados fora dos hiperplanos por meio da adaptação da *lei dos cossenos* para projeção de um ponto no plano. Assim, a etapa de pré-processamento poderia ter um custo computacional menor ou no melhor dos casos poderia ser eliminada. Por outro lado, como a *RLNP* projeta dados por similaridade, poderia-se explorar a capacidade de modificar o *layout* da projeção para melhor separação dos agrupamentos. Finalmente, a abordagem de reprojeção de

agrupamentos poderia ser utilizada para criação de uma técnica de projeção hierárquica e interativa com o usuário.

Referências Bibliográficas

- AKIBA, H.; MA, K.-L. A tri-space visualization interface for analyzing time-varying multivariate volume data. Eurographics Association, 2007, p. 115–122. [16](#), [69](#)
- AKIBA, H.; MA, K.-L.; CHEN, J. H.; HAWKES, E. R. Visualizing multivariate volume data from turbulent combustion simulations. *Computing in Science and Engineering*, v. 9, n. 2, p. 76–83, 2007. [vii](#), [14](#)
- AKIBA, H.; MA, K.-L.; CLYNE, J. End-to-end data reduction and hardware accelerated rendering techniques for visualizing time-varying non-uniform grid volume data. 2005, p. 31–39. [11](#)
- ANAGNOSTOU, K.; ATHERTON, T. J.; WATERFALL, A. E. 4d volume rendering with the shear warp factorisation. New York, NY, USA: ACM, 2000, p. 129–137. [12](#)
- BERNARDON, F. F.; CALLAHAN, S. P.; COMBA, J. A. L. D.; SILVA, C. T. An adaptive framework for visualizing unstructured grids with time-varying scalar fields. *Parallel Comput.*, v. 33, n. 6, p. 391–405, 2007. [13](#)
- BINOTTO, B.; COMBA, J. L. D.; FREITAS, C. M. D. Real-time volume rendering of time-varying data using a fragment-shader compression approach. Washington, DC, USA: IEEE Computer Society, 2003, p. 10. [10](#)
- BLAAS, J.; BOTHA, C. P.; POST, F. H. Interactive visualization of multi-field medical data using linked physical and feature-space views. Eurographics Association, 2007, p. 123–130. [16](#), [69](#)
- BLAAS, J.; POST, C. P. B. F. Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. *IEEE Transactions on Visualization and Computer Graphics*, v. 14, n. 6, p. 1436–1451, 2008. [vii](#), [16](#), [17](#), [69](#)
- CALLAHAN, S.; COMBA, J.; SHIRLEY, P.; SILVA, C. Interactive rendering of large unstructured grids using dynamic level-of-detail. 2005a, p. 199 – 206. [13](#)

- CALLAHAN, S.; IKITS, M.; COMBA, J.; SILVA, C. Hardware-assisted visibility sorting for unstructured volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, v. 11, n. 3, p. 285–295, 2005b. [13](#)
- CAO, Y.; WU, G.; WANG, H. A smart compression scheme for gpu-accelerated volume rendering of time-varying data. 2011, p. 205–210. [11](#), [13](#)
- CAO, Y.; XIAO, L.; WANG, H. Hardware-accelerated volume rendering based on dxt compressed datasets. 2010, p. 523–527. [10](#), [13](#)
- CASEY, J. *A treatise on spherical trigonometry: and its application to geodesy and astronomy, with numerous examples*. Hodges, Figgis, & co., 1889. [26](#)
- CHALMERS, M. A linear iteration time layout algorithm for visualising high-dimensional data. Los Alamitos, CA, USA: IEEE Computer Society, 1996, p. 127–ff. [25](#), [33](#)
- CHEN, W.; DING, Z.; ZHANG, S.; MACKAY-BRANDT, A.; CORREIA, S.; QU, H.; CROW, J. A.; TATE, D. F.; YAN, Z.; PENG, Q. A novel interface for interactive exploration of dti fibers. *IEEE Transactions on Visualization and Computer Graphics*, v. 15, p. 1433–1440, 2009. [vii](#), [2](#), [33](#), [34](#), [69](#)
- CIGNONI, P.; DE FLORIANI, L.; MAGILLO, P.; PUPPO, E.; SCOPIGNO, R. Selective refinement queries for volume visualization of unstructured tetrahedral meshes. *IEEE Transactions on Visualization and Computer Graphics*, v. 10, n. 1, p. 29–45, 2004. [14](#)
- CLEVELAND, W. S. *Visualizing data*. Hobart Press, 1993. [16](#)
- COX, T. F.; COX, M. A. A. *Multidimensional scaling*. Second ed. Chapman & Hall/CRC, 2000. [24](#), [30](#), [47](#)
- DANIELS, J.; ANDERSON, E.; NONATO, L.; SILVA, C. Interactive vector field feature identification. *IEEE Transactions on Visualization and Computer Graphics*, v. 16, n. 6, p. 1560–1568, 2010. [viii](#), [2](#), [35](#)
- DOLEISCH, H.; GASSER, M.; HAUSER, H. Interactive feature specification for focus+context visualization of complex simulation data. Eurographics Association, 2003, p. 239–248. [16](#), [69](#)
- DU, Z.; CHIANG, Y.-J.; SHEN, H.-W. Out-of-core volume rendering for time-varying fields using a space-partitioning time (spt) tree. 2009, p. 73–80. [11](#)
- DUNSMUIR, D.; LEE, E.; SHAW, C.; STONE, M.; WOODBURY, R.; DILL, J. A focus + context technique for visualizing a document collection. 2012, p. 1835–1844. [1](#)
- EADES, P. A. A heuristic for graph drawing. 1984, p. 149–160. [33](#)

- ELER, D.; NAKAZAKI, M.; PAULOVICH, F.; SANTOS, D.; OLIVEIRA, M.; NETO, J.; MINGHIM, R. Multidimensional visualization to support analysis of image collections. In: *Computer Graphics and Image Processing, 2008. SIBGRAPI '08. XXI Brazilian Symposium on*, 2008, p. 289–296. [1](#)
- ELER, D. M. *Múltiplas visões coordenadas para exploração de mapas de similaridade*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2011. [69](#)
- ELVINS, T. T. A survey of algorithms for volume visualization. *ACM SIGGRAPH*, v. 26, n. 3, p. 194–201, 1992. [7](#)
- ENGEL, K.; HADWIGER, M.; KNISS, J. M.; LEFOHN, A. E.; SALAMA, C. R.; WEISKOPF, D. Real-time volume graphics. 2004. [vii](#), [6](#)
- FALOUTSOS, C.; LIN, K.-I. Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. *SIGMOD '95*, New York, NY, USA: ACM, 1995, p. 163–174 (*SIGMOD '95*,). [vii](#), [25](#), [26](#), [27](#), [28](#), [40](#)
- FRIGGE, M.; HOAGLIN, D. C.; IGLEWICZ, B. Some implementations of the boxplot. *The American Statistician Association*, v. 43, n. 1, p. 50–54, 1989. [69](#)
- FRUCHTERMAN, T. M. J.; REINGOLD, E. M. Graph drawing by force-directed placement. *Software - Practice and Experience*, v. 21, n. 11, p. 1129–1164, 1991. [24](#)
- GABRIEL, H.; SPILIOPOULOU, M.; STACHTIARI, E.; VAKALI, A. Summarization meets visualization on online social networks. 2011, p. 475–478. [1](#)
- GAO, J.; HUANG, J.; SHEN, H.-W.; KOHL, J. Visibility culling using plenoptic opacity functions for large volume visualization. 2003, p. 341–348. [12](#)
- GAO, J.; SHEN, H.-W.; HUANG, J.; KOHL, J. Visibility culling for time-varying volume rendering using temporal occlusion coherence. 2004, p. 147–154. [12](#)
- GOVE, R.; GRAMSKY, N.; KIRBY, R.; SEFER, E.; SOPAN, A.; DUNNE, C.; SHNEIDERMAN, B.; TAIEB-MAIMON, M. Netvisia: Heat map & matrix visualization of dynamic social network statistics & content. 2011, p. 19–26. [1](#)
- GRESH, D. L.; ROGOWITZ, B. E.; WINSLOW, R. L.; SCOLLAN, D. F.; YUNG, C. K. Weave: a system for visually linking 3-d and statistical visualizations applied to cardiac simulation and measurement data. *IEEE Computer Society*, 2000, p. 489–492, 597. [16](#), [69](#)
- GUO, H.; XIAO, H.; LU, M.; YUAN, X. Scalable multivariate volume visualization and analysis. 2011a, p. 119–120. [36](#)

- GUO, H.; XIAO, H.; YUAN, X. Multi-dimensional transfer function design based on flexible dimension projection embedded in parallel coordinates. 2011b, p. 19–26. viii, 2, 35, 36, 61, 69
- HADWIGER, M.; LAURA, F.; REZK-SALAMA, C.; HOLLT, T.; GEIER, G.; PABEL, T. Interactive volume exploration for feature detection and quantification in industrial ct data. *IEEE Transactions on Visualization and Computer Graphics*, v. 14, n. 6, p. 1507–1514, 2008. vii, 14, 15
- HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, v. 11, n. 1, p. 10–18, 2009. 63
- HETTICH, S.; BAY, S. D. The uci kdd archive. [Online; acessado em 25/06/2012], 1999. Disponível em: <http://kdd.ics.uci.edu> 30
- HU, Z.; ZHANG, N.; ZOU, J.; RONG, J.; GUI, J.; ZHANG, Q.; XIA, D.; ZHENG, H. Region of interest three-dimensional visualization in clinical human ct images sequence. 2010, p. 490–493. 1
- INGRAM, S.; MUNZNER, T.; OLANO, M. Glimmer: Multilevel mds on the gpu. *Visualization and Computer Graphics, IEEE Transactions on*, v. 15, n. 2, p. 249–261, 2009. 50
- INSELBERG, A. The plane with parallel coordinates. *The Visual Computer*, v. 1, n. 2, p. 69–91, 1985. 16
- IVANOVSKA, T.; LINSEN, L. *Visualization in medicine and life sciences*, capítulo. A user-friendly tool for semi-automated segmentation and surface extraction from color volume data using geometric feature space operations Heidelberg, Germany: Springer-Verlag, p. 153–170, 2007. 16
- JANG, Y.; EBERT, D.; GAITHER, K. Time-varying data visualization using functional representations. *IEEE Transactions on Visualization and Computer Graphics*, v. 18, n. 3, p. 421–433, 2012. 15
- JIANU, R.; DEMIRALP, C.; LAIDLAW, D. Exploring 3d dti fiber tracts with linked 2d representations. *IEEE Transactions on Visualization and Computer Graphics*, v. 15, p. 1449–1456, 2009. vii, 33, 34, 69
- JOHN, G. H.; KOHAVI, R.; PFLEGER, K. Irrelevant features and the subset selection problem. Morgan Kaufmann, 1994, p. 121–129. viii, 63, 64
- JOIA, P.; PAULOVICH, F.; COIMBRA, D.; CUMINATO, J.; NONATO, L. Local affine multi-dimensional projection. *IEEE Transactions on Visualization and Computer Graphics*, v. 17, n. 12, p. 2563–2571, 2011. 33

- JOURDAN, F.; MELANÇON, G. Multiscale hybrid mds. Washington, DC, USA: IEEE Computer Society, 2004, p. 388–393. [25](#)
- KANDOGAN, E. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. New York, NY, USA: ACM, 2001, p. 107–116. [17](#), [69](#)
- KRAUS, M.; ERTL, T. Adaptive texture maps. *ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, p. 7–15, 2002. [10](#)
- LI, W.; MUELLER, K.; KAUFMAN, A. Empty space skipping and occlusion clipping for texture-based volume rendering. *IEEE Visualization Conference*, p. 319–324, 2003. [11](#)
- LING, T.; ZHI-YU, Q. An improved fast ray casting volume rendering algorithm of medical image. 2011, p. 109–112. [7](#)
- LINSEN, L.; LONG, T. V.; ROSENTHAL, P. Linking multi-dimensional feature space cluster visualization to surface extraction from multi-field volume data. *IEEE Computer Graphics and Applications*, v. 29, n. 3, p. 85–89, linsenlongrosenthalvcglsmoothvis, 2009. [ix](#), [72](#), [73](#), [74](#)
- LINSEN, L.; LONG, T. V.; ROSENTHAL, P.; ROSSWOG, S. Surface extraction from multi-field particle volume data using multi-dimensional cluster visualization. *IEEE Transactions on Visualization and Computer Graphics*, v. 14, n. 6, p. 1483–1490, 2008. [vii](#), [2](#), [16](#), [18](#), [69](#)
- LINSEN, L.; PASCUCCI, V.; DUCHAINEAU, M. A.; HAMANN, B.; JOY, K. Hierarchical representation of time-varying volume data with 4th-root-of-2 subdivision and quadrilinear b-spline wavelets. *Pacific Conference on Computer Graphics and Applications*, p. 346, 2002. [10](#)
- LUM, E.; MA, K.; CLYNE, J. Texture hardware assisted rendering of time-varying volume data. *Conference on Visualization*, p. 263–270, 2001. [12](#)
- LUO, D.; YANG, J.; KRSTAJIC, M.; RIBARSKY, W.; KEIM, D. Eventriver: Visually exploring text collections with temporal references. *Visualization and Computer Graphics, IEEE Transactions on*, v. 18, n. 1, p. 93–105, 2012. [1](#)
- MA, K.-L. Visualizing time-varying volume data. *Computing in Science and Engineering*, v. 5, p. 34–42, 2003. [9](#), [13](#)
- MA, K. L.; SMITH, D.; SHIH, M. Y.; SHEN, H. W. *Efficient encoding and rendering of time-varying volume data*. Relatório Técnico, Institute for Computer Applications in Science and Engineering, 1998. [11](#)
- MA, K.-L.; STOMPEL, A.; BIELAK, J.; GHATTAS, O.; KIM, E. J. Visualizing large-scale earthquake simulations. In: *In Proceedings of Supercomputing 2003 Conference*, 2003, p. 48–61. [6](#)

- MCCORMICK, B.; DEFANTI, T.; BROWN, M. Visualization in scientific computing. *Computer Graphics Forum*, v. 21, n. 6, 1987. [1](#), [6](#)
- MIRANDA, F.; CELES, W. Accurate volume rendering of unstructured hexahedral meshes. 2011, p. 93–100. [14](#)
- MORRISON, A.; CHALMERS, M. A pivot-based routine for improved parent-finding in hybrid MDS. *Information Visualization*, v. 3, n. 2, p. 109–122, 2004. [25](#)
- MORRISON, A.; ROSS, G.; CHALMERS, M. A hybrid layout algorithm for sub-quadratic multi-dimensional scaling. Washington, DC, USA: IEEE Computer Society, 2002, p. 152. [25](#)
- MOVANIA, M.; ZHAO, F.; CHEONG, L. S.; LIN, F.; QIAN, K.; SEAH, H. S. Gpu-based surface oriented interslice directional interpolation for volume visualization. 2009, p. 1–5. [7](#)
- MUELDER, C.; MA, K.-L. Interactive feature extraction and tracking by utilizing region coherency. 2009, p. 17–24. [15](#)
- MUIGG, P.; HADWIGER, M.; DOLEISCH, H.; GROLLER, E. Interactive volume visualization of general polyhedral grids. *IEEE Transactions on Visualization and Computer Graphics*, v. 17, n. 12, p. 2115–2124, 2011. [14](#)
- PAIVA, A. C.; SEIXAS, R. B.; GATTASS, M. *Introdução à visualização volumétrica*. Relatório Técnico, Departamento de Informática. Pontifícia Universidade Católica de Rio de Janeiro, Rio de Janeiro, RJ, 1999. [vii](#), [6](#), [7](#), [8](#)
- PAULOVICH, F. V.; ELER, D. M.; POCO, J.; BOTHÁ, C. P.; MINGHIM, R.; NONATO, L. G. Piece wise laplacian-based projection for interactive data exploration and organization. *Computer Graphics Forum*, v. 30, n. 3, p. 1091–1100, 2011. [25](#), [29](#), [45](#), [56](#)
- PAULOVICH, F. V.; ELER, D. M.; POCO, J.; NONATO, L. G.; BOTHÁ, C. P.; MINGHIM, R. *A Fast Projection Technique and its Applications to Visualization of Large Dat Sets*. Relatório Técnico 349, Instituto de Ciências Matemáticas e de Computação (ICMC), Universidade de São Paulo (USP), 2010a. [xi](#), [31](#), [32](#)
- PAULOVICH, F. V.; MINGHIM, R. HiPP: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Transactions on Visualization and Computer Graphics*, v. 14, n. 6, p. 1229–1236, 2008. [24](#)
- PAULOVICH, F. V.; NONATO, L. G.; MINGHIM, R.; LEVKOWITZ, H. Least square projection: a fast high precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics*, v. 14, n. 3, p. 564–575, 2008. [vii](#), [25](#), [29](#), [31](#)

- PAULOVICH, F. V.; SILVA, C. T.; NONATO, L. G. Two-phase mapping for projecting massive data sets. *IEEE Transactions on Visualization and Computer Graphics*, v. 16, p. 1281–1290, 2010b. [25](#), [29](#), [32](#), [45](#), [55](#), [60](#)
- PETERSON, E. Time spring layout for visualization of dynamic social networks. 2011, p. 98–104. [1](#)
- POCO, J.; ELER, D. M.; PAULOVICH; MINGHIM, R. Employing 2d projections for fast visual exploration of large fiber tracking data (to appear, eurovis 2012). *EuroVis*, 2012. [vii](#), [33](#), [35](#)
- RAFLA, N. Real-time 3d image visualization system for digital video on a single chip. 2005, p. 908–911. [1](#)
- RESHETOV, A.; SOUPIKOV, A.; HURLEY, J. Multi-level ray tracing algorithm. *ACM Transactions on Graphics*, v. 24, n. 3, p. 1176–1185, 2005. [13](#)
- ROCHA, A.; MIRANDA, F.; CELES, W. Illustrative volume visualization for unstructured meshes based on photic extremum lines. 2011, p. 101–108. [14](#)
- SAMMON, J. W. A nonlinear mapping for data structure analysis. IEEE Computer Society, 1964, p. 401–409. [24](#)
- SAUBER, N.; THEISEL, H.; SEIDEL, H.-P. Multifield-graphs: An approach to visualizing correlations in multifield scalar data. *IEEE Transactions on Visualization and Computer Graphics*, v. 12, n. 5, p. 917–924, 2006. [16](#), [69](#)
- SAVARESI, S. M.; BOLEY, D. L. A comparative analysis on the bisecting k-means and the pddp clustering algorithms. *Intell. Data Anal.*, v. 8, n. 4, p. 345–362, 2004. [31](#)
- SCHNEIDER, J.; WESTERMANN, R. Compression domain volume rendering. Washington, DC, USA: IEEE Computer Society, 2003, p. 39. [12](#)
- SCHROEDER, W.; AVILA, L.; HOFFMAN, W. Visualizing with vtk: a tutorial. *Computer Graphics and Applications, IEEE*, v. 20, n. 5, p. 20–27, 2000. [18](#)
- SHEN, H.; CHIANG, L.; MA, K. A fast volume rendering algorithm for time-varying fields using a time space partitioning (tsp) tree. *IEEE Visualization Conference*, p. 371–378, 1999. [11](#)
- SHEN, H.; JOHNSON, C. Difference volume rendering: A fast volume visualization technique for flow animation. *IEEE Visualization Conference*, 1994. [11](#)
- SORKINE, O.; COHEN-OR, D. Least-squares meshes. *Shape Modeling and Applications, International*, v. 0, p. 191–199, 2004. [30](#)
- STEINBACH, M.; KARYPIS, G.; KUMAR, V. A comparison of document clustering techniques. Boston, MA, 2000, p. 109–111. [31](#), [47](#)

- TEJADA, E.; MINGHIM, R.; NONATO, L. G. On improved projection techniques to support visual exploration of multidimensional data sets. *Information Visualization*, v. 2, p. 218–231, 2003. 2, 32
- THOM, D.; BOSCH, H.; KOCH, S.; WORNER, M.; ERTL, T. Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages. 2012, p. 41–48. 1
- TIEDE, U.; HOEHNE, K. H.; BOMANS, M.; POMMERT, A.; RIEMER, M.; WIEBECKE, G. Surface rendering. *IEEE Comput. Graph. Appl.*, v. 10, n. 2, p. 41–53, 1990.
Disponível em: <http://dx.doi.org/10.1109/38.50672> 53
- VISCONTEST Visualization fusion (atmospheric data). [Online; acessado em 25/06/2012], 2004.
Disponível em: <http://vis.computer.org/vis2004contest/> 22
- VISCONTEST Multifield 3d scalar data (astrophysics turbulence simulation data). [Online; acessado em 25/06/2012], 2008.
Disponível em: <http://viscontest.sdsc.edu/2008/> 22, 23
- VISCONTEST Stability visualization of fluid dynamics. [Online; acessado em 25/06/2012], 2011.
Disponível em: <http://viscontest.sdsc.edu/2011/> 22, 23
- VISCONTEST Future IEEE visualization contest project. [Online; acessado em 25/06/2012], 2012.
Disponível em: <http://viscontest.sdsc.edu/> 22
- WALD, I.; BOULOS, S.; SHIRLEY, P. Ray tracing deformable scenes using dynamic bounding volume hierarchies. *ACM Transactions on Graphics*, v. 26, n. 1, 2007a. 13
- WALD, I.; FRIEDRICH, H.; KNOLL, A.; HANSEN, C. Interactive isosurface ray tracing of time-varying tetrahedral volumes. *IEEE Transactions on Visualization and Computer Graphics*, v. 13, n. 6, p. 1727–1734, 2007b. 13
- WANG, C.; GAO, J.; LI, L.; SHEN, H.-W. A multiresolution volume rendering framework for large-scale time-varying data visualization. 2005, p. 11–223. 12
- WANG, C.; YU, H.; MA, K. Application-driven compression for visualizing large-scale time-varying volume data. *IEEE Computer Graphics and Applications*, v. PP, n. 99, p. 1, 2009. 10, 13
- WANG, K.; YANG, F.; ZUO, W.; DING, N.; ZHANG, H. Effective transfer function for interactive visualization and multivariate volume data. 2011, p. 272–276. 69
- WILHELMS, J.; GELDER, A. V. Multidimensional trees for controlled volume rendering and compression. *Symposium on Volume Visualization*, 1994. 10

- WOODRING, J.; SHEN, H.-W. Multi-variate, time varying, and comparative visualization with contextual cues. *IEEE Transactions on Visualization and Computer Graphics*, v. 12, n. 5, p. 909–916, 2006. [16](#)
- WOODRING, J.; WANG, C.; SHEN, H.-W. High dimensional direct rendering of time-varying volumetric data. Washington, DC, USA: IEEE Computer Society, 2003, p. 55. [10](#)
- XING, E. P.; KARP, R. M. CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics*, v. 17, n. suppl_1, p. S306–315, 2001. [62](#)
- YOUNG, G.; HOUSEHOLDER. Discussion of a set of points in terms of their mutual distances. 1938, p. 19–22. [33](#)
- ZHANG, Q.; EAGLESON, R.; PETERS, T. Gpu-based image manipulation and enhancement techniques for dynamic volumetric medical image visualization. 2007, p. 1168 –1171. [1](#)
- ZHU, J.; LU, L. Perceptual visualization of a music collection. In: *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, 2005, p. 1058 –1061. [1](#)

Apêndice A

Nesse apêndice está o *mini-paper* elaborado para o concurso de visualização *IEEE Visualization Contest 2011*. Que teve alvo a área de dinâmica dos fluidos com o objetivo de elaborar uma visualização que permita explorar a estabilidade em uma simulação da mecânica dos fluidos em relação a diferentes modelos de turbulências.

O *mini-paper* foi elaborado em conjunto com o estudante de mestrado Thiago Reis Santos, o pesquisador de Pós-doutorado Danilo Medeiros Eler e os Professores do Grupo VICG: Dra. Maria Cristina Ferreira de Oliveira, Dra. Rosane Minghim e Dr. Luis Gustavo Nonato.

Finalmente foi aceito como contribuição e será publicado nos jornais eletrônicos da *IEEE VisWeek 2011*.

Vortices Identification Based on Projection of Streamlines

Thiago S. Reis Santos* Christian Jorge Daniel Wong Cruz † Maria Cristina Ferreira de Oliveira ‡
Rosane Minghim § Luis Gustavo Nonato ¶ Danilo Medeiros Eler ||

Instituto de Ciências Matemáticas e Computação - Universidade de São Paulo, Brazil

1 INTRODUCTION

In this work we present a visualization strategy to support identification of vortices in a vector data set. Usually, two methods may be employed for this task: λ_2 [3] or the Q -criterion[2]. However, these methods are not necessarily the best options for vortex identification. Rather than using the original attributes and selecting either of those methods for vortex identification, we change the focus of the problem by applying a transform and conquer approach. We first compute various streamlines from the vector data and extract shape information from them by means of the Discrete Fourier Transform (DFT), building a feature space with the (low frequency) Fourier coefficients. In this way, our new problem is how to localize the vortical streamlines, i.e., the streamlines belonging to vortical regions. We do so employing multidimensional projection techniques, which strive to preserve the relationships of the high-dimensional feature space on the 2D projected space. The underlying idea is to locate vortical streamlines by recognizing the peculiar shapes of groups formed from projecting their distinguishing features.

We have compared different projections techniques for the task and the best results were obtained with the Piecewise Least Square Projection (P-LSP) technique [5]. Once the streamline feature space is projected, one can select groups in the 2D visual space and visualize the selected vortical streamlines in the object space.

The views and linking mechanisms were developed in Java and the streamline visualizations were done using The Visualization Toolkit¹.

2 VISUAL EXPLORATION PROCESS

The data set made available by the Vis Contest 2011 organizers results from a time-varying fluid dynamics simulation of a pump. The simulation includes different attributes: pressure, total pressure, total pressure in stn frame, turbulence kinetic energy, velocity and velocity in stn frame. The simulation dataset consists of 80 time steps, each one comprising 6.7 million nodes and 6.4 million cells. Furthermore, three data sets are available from the same centrifugal pump, by the application of three distinct turbulence models: LES, RANS and the hybrid LES/RANS. We focused our efforts in the LES simulation model.

Each time step is divided in 24 parts. In this work we have focused our analysis in the rotor domain (i.e., all volume elements named with ROTOR.VOL), which consists of 2.24 million nodes.

A major problem in streamline computation regards seed positioning. We compute streamlines with seeds uniformly distributed

throughout the whole volume, using the nodes as seeds. We believe that this approach is able to cover all regions of the volume. As the streamlines tend to cover similar regions, we have used only a subset consisting of 22,478 nodes to generate the same number of streamlines.

The streamline feature space is computed by applying a Fourier transform over the 3D points on the streamline path [7]. The 1D Fourier transform is computed for the X, Y and Z coordinate arrays. We then select only coefficients related to low frequencies, 20 coefficients per coordinate, resulting on a feature vector with 60 coordinate values for each streamline.

The streamline vector space is projected onto a 2D plane, using an adapted version of the PLSP technique. The adaptation introduced fixes certain control points across different time stamps to maintain visual coherence across successive time steps. Users can locate and select in the projection well formed groups of points representing vortical streamlines. As multidimensional projections such as PLSP strive to derive a layout that preserves proximity of highly similar instances (in the multidimensional feature space), vortical streamlines are likely to group in the same projection regions over time. When a neighboring time step is loaded, the object space is updated according to the previous user selection. The nodes corresponding to selected points are then rendered in object space. Linking between the feature and object space representations provides an interaction mechanism that suppresses the need of users handling three dimensional spaces or interacting with complex structures in the volume.

Figure 1 illustrates the exploration process enabled by our linked view approach. Initially, in Figure 1(a) the first time step is loaded and visualizations of both the projected feature space and the fluid are shown. Next, in Figure 1(b), the user selects a few groups in the projection (left image) which are then shown in object space (right image), revealing that the selection comprises groups of vortical streamlines. This same user selection is maintained when the next (second) time step is loaded. As illustrated in Figure 1(c), spatial coherence is preserved between the projections, that is, the vortical streamlines belong to the same selected region.

Figure 2 illustrates a vortex, highlighted by the red polygon, appearing in the sequence comprising time steps 5 to 10. The vortex appears from the inside to the outside part of the rotor blade. Projections and selections corresponding to Figure 2 are shown in the attached pictures.

3 TIME CONSUMPTION

The visualization application was executed on a Intel i7 2.8 GHz computer with 16GB of RAM and an IDE hard disk. Prior to exploration the data set was preprocessed to generate the streamlines for each vertex, with a step of 100. Streamlines were computed employing the vtkStreamTracer class, and computing the 22k streamlines took approximately three hours. We have also preprocessed the projections associated to all (80) time steps, which takes under 18 minutes. The streamlines and the projection relative to one time step (both stored in the hard disk) are loaded during visual exploration in just under a minute. Coordination between projection and object space is done in real time.

*e-mail: thiagors@icmc.usp.br

†e-mail: cwong@icmc.usp.br

‡e-mail: cristina@icmc.usp.br

§e-mail: rminghim@icmc.usp.br

¶e-mail: gnonato@icmc.usp.br

||e-mail: eler@icmc.usp.br

¹vtk.org

(a) Time step 0.

(b) Time step 0 - User selection.

(c) Time Step 1 - Same position of previous user selection.

Figure 1: Identification of vortices: (a) visualizations of the streamline feature space and the object vector space; (b) Selection of well defined groups and visualization of the corresponding streamlines. Selected group corresponds to vortical streamlines; (c) Next time step loaded, maintaining the selection performed in (b), reveals the 2D spatial coherence of the projected streamlines.

4 COMMENTS AND CONCLUSIONS

We initially tried to analyze the relationship of the nodes by projecting a feature space composed by the raw attributes given in the data set. The rationale was to check whether nodes with similar attributes lead to grouping of vortical regions. Results were not satisfactory, that is, apparently vortices are not well grouped by their raw simulated attributes. That first run took the team nearly 20 days, part time, to complete.

The approach described in this work was the best solution we found to enable user controlled vortex detection in a fluid flow data set. The final solution was achieved by the team in one week, also working part time.

We have also investigated other projection techniques to group the streamlines, e.g., Principal Component Analysis [4], Fastmap [1] and Part Linear Multidimensional Projection [6], but PLSP yielded the best results.

We developed a prototype tool for this linked view approach,

Figure 2: Illustration of a vortex appearing from the center to the border of a blade (highlighted in red).

which provides an effective way to select groups of streamlines in 2D space, aiding in the selection and identification of vortices.

5 ACKNOWLEDGEMENTS

This work was supported by the Brazilian research agencies FAPESP and CAPES.

REFERENCES

- [1] C. Faloutsos and K.-I. Lin. Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proc. 1995 ACM SIGMOD Int. Conf. Manage. of Data*, SIGMOD '95, pages 163–174, New York, NY, USA, 1995. ACM.
- [2] G. Haller. An objective definition of a vortex. *J. of Fluid Mech.*, 525:1–26, 2005.
- [3] J. Jeong and F. Hussain. On the identification of a vortex. *J. of Fluid Mech.*, pages 69–94, 285 1995.
- [4] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, NY, USA, 2nd edition, 2002.
- [5] F. V. Paulovich, D. M. Eler, J. Poco, L. G. Nonato, C. P. Botha, and R. Minghim. A Fast Projection Technique and its Applications to Visualization of Large Data Sets. Technical Report 349, Instituto de Ciências Matemáticas e de Computação (ICMC), Universidade de São Paulo (USP), March 2010.
- [6] F. V. Paulovich, C. T. Silva, and L. G. Nonato. Two-phase mapping for projecting massive data sets. *IEEE Trans. on Vis. and Comp. Graph.*, 16:1281–1290, 2010.
- [7] M. A. Pazoti, R. E. Garcia, J. Dalton, C. Pessoa, and O. M. Bruno. Comparison of shape analysis methods for guindardia citricarpa ascospore characterization. *Elect. J. of Biotech.*, 8(3):265–275, 2005.

Apêndice B

Nesse apêndice está o artigo submetido no *20th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2012)*. Que concentra as principais contribuições apresentadas nessa dissertação de mestrado.

O artigo foi elaborado em conjunto com as Professoras do Grupo VICG: Dra. Maria Cristina Ferreira de Oliveira e Dra. Rosane Minghim.

Adapting Multidimensional Projections to Explore Time-Varying Multivariate Volume Data

paper1150

Abstract

Multidimensional Projections (MPs) are becoming an increasingly popular tool for visual data analysis in some areas, such as Scientific Visualization, have evolved to mapping strategies that are fast, precise and capable of handling local and global data features. However, for the purpose of volumetric data sets there are three main problems that are not fully addressed by recent studies. MPs do not provide mechanisms for time-varying data sets, do not support answering attribute-specific questions, and they do not form well defined groups due to the continuous nature of natural phenomena. The projections resulting from multivariate volume data shows a smooth distribution that does not favor the exploration process. In this paper we present a variation of the Fastmap and PLP projection algorithms to give them the power to handle time-varying volumetric data sets. We also propose a new hybrid visualization for analyzing booth similarity and a user-selected focus attributes. The problem of finding groups on projections is handled by reprojecting the volume taking into consideration non-relevant attributes. The proposed approaches improve currently available exploration processes for exploring Volumetric data using MPs.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms

1. Introduction

Volume visualization [Elv92, GW11] deals with methods to explore, analyze and visualize volumetric data, supporting the investigation of dynamical aspects of many critical physical, chemical or biological processes. Volumetric data sets are acquired by sampling, simulation or modeling techniques and are typically represented by three-dimensional regular or irregular meshes. Such spatial representations register observations of multiple scalar and vector variables acquired or computed over time, associated to voxels, cells or nodes. Thus, data consist of a sequence of volumetric meshes each corresponding to a particular time stamp. Understanding and exploring such time-varying multivariate volume data is still a major challenge faced by scientists.

In this paper we investigate the role multidimensional projections (MPs) as tools to assist exploration of time-varying volumetric data with multiple attributes or multivariate volume data. We deal with three critical issues:

1. There are still few projection techniques available that can handle large volume data sets in a time-varying context.

2. Projections offer little support to users trying to establish the role of specific attributes in global behavior.
3. Projections of volume data tend to produce very dense and cluttered layouts, hindering perception of groups of similar elements, which is essential to support investigation of such data.

Addressing these issues requires the ability of deriving 2D projections from time varying multivariate volumetric data in real-time. This scenario imposes the following requirements: (i) speed, i.e., projections of different time-stamps should be obtained at interactive rates; (ii) low memory consumption, as multivariate volumes are typically very large and several must be handled simultaneously; and (iii) preservation of temporal coherence, i.e., projections in all time stamps should preserve global placement and orientation, so that users can relate them.

Fast techniques such as Fastmap [FL95] and PLP (the *Part-Linear Projection*) [PEP*11] might be considered, as they handle high-dimensional data at interactive rates. However, temporal coherence is not guaranteed when projecting a sequence of volumes. We thus modify their original formulation to allow time-varying volume data by producing time-coherent layouts. In order to support the investigation

of the role of specific attributes we introduce a hybrid 2D visualization that combines a one-dimensional projection with a scatter plot. Finally, we propose an attribute selection strategy that results in projections with improved clustering, favoring interpretation and exploration.

2. Related Work

MPs provide support for visual exploration of multivariate volumes by mapping attributes in volume voxels as points in a 2D plane, causing voxels with similar properties to be mapped in nearby areas. Several authors describe exploratory volume visualization solutions that integrate projections or other multidimensional visualizations. For example, Chen et al. [CDZ*09] provide linked 3D and 2D views to assist exploration of Diffusion Tensor Imaging (DTI) fibers. The 2D representation is computed by Classical Scaling [YH38], which preserves the spatial relationships of the fiber tracts and removes the visual clutter characteristic in the 3D visualization.

Jianu et al. [JDL09] also employ projection techniques to assist exploration of DTI Fibers. Additionally, they employ a force direct method [Ead84], adopting Chalmers' [Cha96] acceleration technique to reduce computation requirements and achieve interactive performance. Poco et al. [PEPM12] proposes the use of a MP called LAMP [JPC*11] for fast visualization of large fiber tracking datasets with the selection of groups of bundled fibers for further exploration.

Daniels et al. [IANS10] use multidimensional projections to assist interactive exploration of vector fields, observing that similar features within the vector field share similar neighborhood characteristics, even if they are spatially apart. Users modify the control point locations and design a texture over an interactive canvas to improve projection quality.

Recent work by Santos et al. [SWdO*11] describes a visualization strategy to support identification of vortices in vector data. Authors compute streamlines from velocity data, extract shape information as features and project them using the PLSP projection [PEP*10]. Groups of projected streamlines may be selected, and streamline visualization is done with the Visualization Toolkit [SAH00].

These previous methodologies, however, do not address the problem of temporal coherence. To the best of our knowledge, the only multidimensional projection capable of properly handling high-dimensional data while preserving temporal coherence is the *Part-Linear Multidimensional Projection* (PLMP) [PSN10]. This a fast and precise technique capable of handling massive data and supporting interactive exploration of time-varying volumes. PLMP, however, has no specific interaction strategies for finding groups on the projected data and has presented similar layout to FASTMAP in most cases with large number of points. PLMP also has specific requirements regarding the number

of attributes against the number of items and may not project well if they are not met.

3. Background

The two MPs that we have adapted to account for time coherence and grouping with large datasets are Fastmap and PLP, summarized below.

3.1. Fastmap

Fastmap [FL95] is a fast algorithm for mapping high-dimensional objects (data instances) into a lower k -dimensional space. The mapping is applicable as a projection technique if $k = 2, 3$, and attempts to preserve distances (or dissimilarities) among objects while projecting them onto k mutually orthogonal directions.

The rationale is to recursively project the objects onto k hyperplanes defined by a subset of the dimensions to obtain the object coordinates in each projected dimension.

In each recursion step a projection direction is defined by selecting, within each hyperplane, two pivot objects (O_a and O_b), and then projecting all objects in the line defined by the pivots. The projection of an arbitrary object O_i in each hyperplane is computed using the Cosine Law:

$$d_{b,i}^2 = d_{a,i}^2 + d_{a,b}^2 - 2x_i d_{a,b} \quad (1)$$

where $d_{i,j}$ denotes the distance between any two objects O_i and O_j , x_i is the desired projected coordinate of O_i in the line $\overline{ab}$ that connects the pivots O_a and O_b . Equation 1 may be solved for x_i and the distance between any two projected objects may be computed employing the Euclidean distance:

$$d'_{i,j} = \sqrt{d_{i,j}^2 - (x_i - x_j)^2} \quad (2)$$

Equation 2 allows to project over another line which is orthogonal to the first.

A final issue is how to select the pair of pivots. The ideal choice is to select the two farthest apart data objects, but this is computationally expensive. The following heuristic [FL95], illustrated in Figure 1, provides an alternative. First an arbitrary object A is chosen. Then the object farthest apart from A is found and taken as the first pivot, O_a . Again, the object farthest apart from O_a and found and taken as the second pivot, O_b .

However, the selected pivots are not necessarily good representatives of the entire data set, as a data point P may fall outside the area defined by them, as shown in Figure 1. This affects the method's precision, which may be improved by running further iterations of the heuristic, taking the second

Figure 1: Linear heuristic for finding the pivots (O_a and O_b). It fails when an object P falls outside the parallel region defined by the pivots, as illustrated.

pivot as the initial object. This solution, however, incurs in higher computational cost.

The computational cost of Fastmap is determined by the two steps executed as many times as the number of target dimensions. Choosing the pivots requires three iterations over the data, resulting in complexity $O(3n) = O(n)$. The second step is just an iteration over the data calculating the projected coordinate using the cosine law. Thus, overall cost is $O(k(3n + n)) = O(n)$, since k (1, 2 or 3) is low.

3.2. Piecewise Laplacian-based Projection

The Piecewise Laplacian-based Projection (PLP) [PEP*11] is fast and allows real-time interaction with large data sets. The method comprises three major steps: sampling, building a neighborhood graph and solving a Laplacian linear system.

The sampling step selects a small representative subset from the original data, and may be performed automatically by clustering and taking sample inside the formed clusters. These samples are projected with a *Multidimensional Scaling* (MDS) technique such as the *Least Squares Projection* (LSP) [PNML08]. Then, the neighborhood is defined by a *k-nearest neighbor graph* (*kNNG*), where nodes representing data objects are connected to their *k*-nearest neighbors.

Finally, a Laplacian linear system is defined based on the assumption that each element O_i can be written as a convex combination of its nearest neighbors in the visual space. Under this hypothesis, the projected points for each cluster can be written as:

$$x_i = \sum_{O_{ij} \in kNN(O_i)} \alpha_{ij} (x_{O_{ij}}, y_{O_{ij}}) \quad (3)$$

where $\alpha_{ij} > 0$ and $\sum \alpha_{ij} = 1$.

From Equation 3 a linear system may be assembled as:

$Lx = 0$ (and likewise $Ly = 0$). The coefficients in matrix L are defined as 1 when $i = j$, $-\alpha_{ij} / \sum_{O_{ij} \in kNN(O_i)} \alpha_{ij}$ if $O_{ij} \in kNN(O_i)$ and 0 otherwise. The value of α_{ij} can be the inverse of the distance between objects O_i and O_j or simply equal to one.

To ensure the system has a unique solution, some constraints are imposed based on the control points: $\sqrt{n_i}$ control points are extracted from each cluster C_i , with $|C_i| = n_i$, and projected in the visual space employing a precise (and usually costly) method. Their projected coordinates provide the required constraints for the system.

PLP has cost $O(S + C + P + I)$, where S is the complexity of the initial sampling, C is the complexity of creating the partitions, P is the complexity of projecting the control points, and I is the complexity of projecting each individual cluster. Sampling is $O(n)$, since a subset is chosen randomly from the original data. Bisecting k-means, projection of the control points, and projection of the clusters are each $O(n\sqrt{n})$. Therefore PLP complexity is $O(n + 3 * n\sqrt{n}) = O(n\sqrt{n})$.

4. Methods

In this section we describe how Fastmap and PLP were modified to preserve temporal coherence when projecting time-varying data.

4.1. Fastmap* for time-varying data

As discussed in Section 3.1, Fastmap requires solving Equation 1 to obtain x_i for each object, given a pair of pivots O_a and O_b . Applying the pivot selection heuristic to time-varying data will likely output very different pivot pairs at subsequent time steps, introducing some unexpected variations over time, as observed in Figure 2. In that picture, as well as in the examples given during the explanations of our techniques, we employ the IEEE Visualization 2008 data set, which is based on particle data for simulation of an ionization front. Additional details of the data set are presented in Section 5.

In comparison with our approach they propose traversing the whole data set several times to compute the pivots, resulting in a time-expensive solution that is not interactive.

Our approach to tackle this problem relies on the assumption that time-varying data presents spatial and temporal coherence, i.e., no sudden large variations would occur in a reduced sub-sequence of timesteps. This assumption is likely to hold for time-varying data obtained from a complete simulations of a single phenomenon.

Moreover, coordinate x_i for each point is computed without traversing the data several times. Since pivot selection is independent of the projection step, running more iterations of the heuristic may result in a more representative pair of pivots.

Figure 2: Fastmap layouts obtained for three different timesteps of the VisContest 2008 data. The layout for timestep 90 is rotated as compared with those of timesteps 60 and 120.

Once we have the pivots previously calculated, we can now project each vector separately. In order to do that, we propose an interactive implementation of the *Cosine Law* listed in Procedure 1.

Procedure 1 Projection of a vector using *Fastmap**

Input: V : Vector to be projected.

O_a, O_b : Pivots of the first hyperplane.

O_i, O_j : Pivots of the second hyperplane.

O'_a, O'_b, O'_i, O'_j : Pivots values on the projected space.

$d()$: Dissimilarity to be used.

Output: V' : Vector projected.

1. **For** ($d = 1$) **to** 2 {

a. **If** $d = 1$ {

- $dist_O_a_V \leftarrow d(O_a, V)$
- $dist_O_a_O_b \leftarrow d(O_a, O_b)$
- $dist_O_b_V \leftarrow d(O_b, V)$

b. }

c. **Else** {

- $dist_O_a_V \leftarrow \sqrt{|(d(O_a, V))^2 - (O'_{a_{d-1}} - V'_{d-1})^2|}$
- $dist_O_a_O_b \leftarrow \sqrt{|(d(O_a, O_b))^2 - (O'_{a_{d-1}} - O'_{b_{d-1}})^2|}$
- $dist_O_b_V \leftarrow \sqrt{|(d(O_b, V))^2 - (O'_{b_{d-1}} - V'_{d-1})^2|}$

d. }

e. $V'_d \leftarrow \frac{(dist_O_a_V)^2 + (dist_O_a_O_b)^2 - (dist_O_b_V)^2}{2dist_O_a_O_b}$

2. }

3. **return** V'

4.2. Recursive Laplacian-based Neighborhood Projection

As seen in section 2, in the literature we only find one multi-dimensional projection technique that preserves the temporal coherence of the data set, which is the *PLMP* [PSN10]. This technique and our *Fastmap**, have similar characteristics and limitations. Among those we can identify two main

limitations: they produce smoothed projections and are better for low dimensionality.

The first limitation might hinder the exploration process by mixing similar groupings within the projection. And the second does not allow data sets with large quantities of derived attributes to be explored for multi-dimensional projections.

Given those two limitations, we adapted the LSP using a similar approach to that proposed by [PEP*11], thus allowing its application in time-varying volumetric sets. Our proposal is to recursively apply the idea of creating small clusters of data based on neighborhood in the original space, so we can embrace all the timesteps together. Additionally we define the concept of “local control points” and “temporal control points”, which are respectively the representatives points of each cluster in an arbitrary timestep and temporal when they describe clusters of data spanned through several timesteps simultaneously.

Given the recursive interpretation of this approach, we call it *Recursive Laplacian-based Neighborhood Projection* or simply *RLNP* and it is divided in two main steps: *preprocessing* and *projection*. In the preprocessing step, we cluster each timestep separately and get the local control points which will set up the layout of all the remaining objects of the timestep. Then we cluster again all the local control points together in order to find the temporal control points of the data set. As shown in Procedure 2, we partition the data in each recursion into balanced clusters of similar instances. After that, we project all the temporal control points using a *Multidimensional Scaling (MDS)* technique (i.e. Fastmap). In the projection step, we build a neighborhood graph and solve a Laplacian linear system similar to 3.2 for each recursion, first, to find the position of the local control points on the projected space and then to find the projection of each timestep.

We have investigated the quality of layouts generated with *Recursive Laplacian-based Neighborhood Projection* (from

Procedure 2 Algorithm for extracting control points.**Input:** D : Time-varying data set.**Output:** Control points.

1. For each time step D_t :
 - a. apply the *bisecting k-means* on data volume D_t obtaining clusters C_{ti} .
 - b. For each cluster C_{ti} :
 - i. create the kNN graph for C_{ti} .
 - ii. $CP_{ti} \leftarrow$ randomly selected control points from C_{ti} .
 - c. save the neighborhood information for later and return the control points CP_{ti} .

now on *RLNP*) and Modified FastMap (from now *Fastmap**) as a function of data dimensionality. To test that we have employed a high-dimensional text data set composed by 574 scientific articles (with title, abstract, and references) from three different subjects: case based reasoning (CBR), inductive logic programming (ILP), and information retrieval (IR). The CBR and ILP articles were obtained manually from the Lecture Notes on Artificial Intelligence (LNAI) series, and the ILP ones were retrieved from the Web. Different ‘bag-of-words’ vector representations of the data were generated, varying the number of ‘words’ (features), or dimensions): tests considered 118, 144, 283, 307, 641, 2,548, 2,591, 3,000, 18,694 and 22,313 dimensions. Notice that representations with more than 600 features define very sparse data spaces in this case. Figure 3 shows the stress [IMO09] values computed for the *RLNP* and *Fastmap** methods. One notices that those of *RLNP* are consistently smaller and increase slowly as data dimensionality increases, as compared to *Fastmap**.

4.3. Scatter Projection view

Layouts derived from projections are great tools for visual exploration, as users may interact to identify and analyze groups of similar data instances. However, they fail to reveal relationships amongst data values characterizing attributes. We propose a hybrid visualization to support investigating the role of specific attributes, which combines the intuitiveness of scatter plots with the segmentation capability of projections.

This visualization adopts a 2D Cartesian coordinate layout, with the horizontal axis mapping a user-defined focus attribute (data variable) and the vertical axis mapping a 1-dimensional projection of the data. This projection may be obtained, e.g., by employing a single iteration of Fastmap. The resulting visualization allows focusing on the global similarity between instances while simultaneously investigating the role of a target attribute on this global behavior, as shown in Subsection 5.3.

Figure 3: Comparison of stress values of layouts produced by Fastmap* and RLNP, as a function of data dimensionality.

4.4. Reprojecting user selections

The nature of the data poses additional difficulties to MPS for exploration of scientific volumes. In simulation data, neighboring (i.e., spatially close) instances tend to clutter the visual space because relatively large neighborhoods are deemed considerably more similar than other less populated regions of the data. A projection of the attribute space may reveal no visible groups in these circumstances. Moreover, the large number of voxels and reduced visual space may result in dissimilar voxels, that actually correspond to different spatial structures, to be visually perceived as close.

This is illustrated in Figure 4, which shows views of the VisContest 2008 data at timestep 90: the top image depicts a projection of the attribute space, whereas the bottom image depicts a volume (object space) view of the voxels selected in the projection (selection is indicated by the red rectangle). The selection, apparently comprising a small area, actually includes 42% of the volume voxels. This example illustrates the difficulties in identifying related groups and exploring internal structures within a volume. To assist users in this task, we propose an interaction facility that generates improved projections of a user selection, with better visual separation of the groups within this specific area.

The data attributes certainly play a central role in the projection layout computed by a particular technique. Challenges are analogous to those faced by clustering algorithms due to data sparsity, high-dimensionality and the presence of irrelevant or redundant attributes [XK01]. Attribute (or feature) selection algorithms are typically employed to identify the attributes most relevant for obtaining a good clustering or a projected layout – i.e., that actually separate similar and

dissimilar objects. This has been attempted, for example, by Joia et al. [JGB*11] in the context of an image visual classifier. Authors employ a feature selector to identify the features that best characterize instances in each class, defining a class-specific metric to quantify the similarity of classified instances and obtain a less cluttered projection of the images.

Our scenario is rather different, as we are handling very large and non-classified data. Given a sub-set of user selected voxels over the original projection, we want to generate a less cluttered projection. If the selected voxels are considered as examples from a hypothetical unique class, we can run an attribute selection algorithm to identify the attributes that best characterize this class, the so-called most relevant attributes. The complement set defines the *less-relevant* attributes that are not contributing to class characterization.

Our rationale is to input this sub-set of less-relevant attributes to the projection, to obtain a novel layout of the selection which will be typically less-cluttered. We investigated several attribute selection algorithms in this context. Such algorithms are typically categorized into one of three approaches: filter-based, wrapper-based and embedded approaches. We obtained best results in our experiments with Correlation-based Feature (CFS) Subset Selector [Hal98] (filter-based), Multi-Layer Perceptron [PNSV05] (wrapper-based) and IBK algorithm [AK91] (embedded). The approach may be summarized as:

1. Select a group of voxels in the projection.
2. Apply a feature selector, assuming a single class, to split the attribute set into two subsets of most-relevant and less-relevant.
3. Reproject the selected points using the less-relevant attributes.

5. Experiments and Applications

We now illustrate how the proposed techniques may be employed to assist exploration of time-varying volume data.

5.1. Data sets

We employ as test cases two data sets made available for editions of the *IEEE Visualization Contest*. Both describe multivariate simulations of phenomena with a 3D representation on the visualization space.

Ionization front simulation

This data has been released for the *2008 IEEE Visualization Contest* [WN08], and consists of multifield 3D scalar volumes output by a simulation designed to understand the formation of galaxies, in particular the effect of *shadow instabilities* where radiation ionization fronts scatter around primordial gas. Data is organized in 600x248x248 regular meshes describing 200 time steps. Each cell describes 10

(a) Projection of timestep 90 with a selected group of instances.

(b) Object-space visualization of selected area in timestep 90.

Figure 4: Attribute space (a) and object space (b) views of the VisContest 2008 data at timestep 90. The red area in the projection defines a selection that includes 42% of the volume voxels. In the projection view (a) points are drawn with transparency and regions with higher opacity may indicate relevant structures in the object space. The object space view (b) depicts the selected voxels only.

scalar values, namely: Total particle density, gas temperature, and 8 types of mass abundances (H, H+, He, He+, He++, H-, H2, H2+). Each cell has also a velocity vector, from which additional scalar values may be extracted, namely magnitude, divergence and curl magnitude.

We downsampled each volume by a factor of 4 by randomly selecting one cell from each group of 4x4x4 cells, generating a volume with 576,600 voxels. The temporal series was also reduced to 21 steps, corresponding to time stamps 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 199).

Fluid dynamics simulation

This data was released for the *2011 IEEE Visualization Contest* [IoAM11] and describes a simulation of a centrifugal pump described using three models, namely RANS, LES and a Hybrid LES/RANS, where each data set comprises one

full rotation of the centrifugal pump described over 80 time steps. Each time step is described by an irregular volume mesh where each cell has 6 associated attributes. The scalar variables are pressure (static pressure), total pressure (static pressure plus kinetic energy of the relative velocity in pressure units), total pressure in stn frame and turbulence kinetic energy (static pressure plus kinetic energy of the absolute velocity); the vector variables are velocity (in the relative system) and velocity in stn frame (in the absolute system). The meshes actually describe several parts, or domains, but we focused specifically on the rotor domain.

We computed 22,478 streamlines from an equal number of random cells, from which features were extracted after applying the Fourier Transform. The resulting data has 39 time-stamped volumes, each with 22,478 streamlines described by 57 features. The data preprocessing is detailed elsewhere [SWdO*11].

5.2. Comparison RLNP

We compare layouts obtained with *RLNP*, the original *PLP* and the *PLMP* techniques for the VisContest 2011 data. Since *PLP* does not handle the temporal variation, all timesteps were projected simultaneously in order to achieve temporal coherence, and observation is from a single step at a time. *Stress Curves* are shown in Figure 5

The *PLP* layout shown is a zoomed-in (by a factor of 18) view of the original, which groups 89% of the data voxels in the central region, and spreads the remaining voxels around the main cluster. This is very ineffective for user exploration. Similarly, the output of *PLMP* shows no visible clusters of similar instances. In contrast, the layout produced by *RLNP* shows many small clusters. It has the lowest stress value and the best *stress curve*.

5.3. Scatter Projection to answer specific questions

We illustrate how the *Scatter Projection* visualization helps answering a question posed at the 2008 Visualization Contest about the star formation simulation. The question is: *H₂ enables primeval gas clouds to collapse and form the first stars before galaxies later coalesce. Where is H₂ most prevalent in the simulation?*

The horizontal axis maps the target attribute *H₂ Mass Abundance*, and the vertical axis maps a 1D projection of the data obtained considering only attributes *density* and *temperature* in the *Scatter Projection* view shown in Figure 6(a) (left). The top right view shows a volume rendering of the points selected in the *Scatter Projection*. One observes that higher values of *H₂* are concentrated in the middle of the projection axis. Such values appear in the volume view concentrated in the middle and in the back of the advancing ionization front.

The *Scatter Projection* shown in Figure 6(b) maps *turbulence* as the focus attribute in horizontal axis, and the same

Figure 5: Projected layouts of the VisContest 2011 data at timestep 28 and their respective Stress Curves, for (a) *PLP* (stress 0.7383191), (b) *PLMP* (stress 0.146437) and (c) *RLNP* (0.11202002).

projection described in the previous figure in the vertical axis. Unlike *H₂*, higher turbulence values are not concentrated in a particular region of the projection axis, reflecting the prevalence of *H₂* in this instance of the simulation. The volume view reveals that high turbulence values occur in the middle of the two regions depicting high *H₂* values.

These findings are in accordance with those reported by Linsen et al. [LLR09] in their discussion on the prevalence of *H₂*. In this experiment we use the same timestep 99 in order to compare our results.

(a) H_2 Mass Abundance. Scatter Projection and object-space visualization.

(b) Turbulence. Scatter Projection and object-space visualization.

Figure 6: (top) Scatter Projection visualization of H_2 Mass Abundance versus 1D projection of density and temperature; and (bottom) Turbulence versus 1D projection of density and temperature. (VisContest 2008 data).

5.4. Further exploring user selections

Let us consider Figure 4 and the selection shown. Figure 7 depicts the results of reprojecting that selection: in the projection in (a) only the sub-set of attributes deemed as most relevant by the attribute selection algorithm – in this example the *Multilayer Perceptron* – were employed; in the projections in (b) and (c) only the sub-set of attributes deemed as less relevant were employed, following the approach described in Section 4.4. Corresponding volume views are shown on the right. As expected, clutter is reduced in the layout obtained by projecting the less relevant attributes, clearly indicating the presence of two groups. Figures 7(b) and 7(c) show a user selection of elements in each group, and the right view shows the corresponding object-space volume views of the selected voxels. Notice that the first group, in (b), cor-

responds to the explosion front, whereas the second one, in (c), corresponds to its body.

5.5. Execution times

The execution times of the *Fastmap** and *RLNP*, relative to their original formulations, are detailed in Table 1, for the data sets described in Section 5.1. Since we introduced the Pivot and the Control Points extractors in the modified Fastmap and PLP as a preprocessing step, we show preprocessing and projection timings for both.

The total time for projecting a time-varying data set with *Fastmap** or *RLNP* is given by $(\text{preprocessing time}) + (\text{number of time steps}) \times (\text{projection time})$. For *RLNP* the total time to project is given by $(\text{number of time steps}) \times [(\text{preprocessing time}) + (\text{projection time})]$.

Technique	VisContest 2004		VisContest 2008		VisContest 2011	
	$pre(t)$	$proj(t)$	$pre(t)$	$proj(t)$	$pre(t)$	$proj(t)$
PLMP	0,58s	0,296s	0,812s	0,064s	0,041s	0,014s
Fastmap		1,926s		1,488s		0,194s
Fastmap*	1,691s	0,327s	1,387s	0,395s	0,177s	0,043s
mCPU Fastmap*	1,589s	0,13s	1,457s	0,176s	0,190s	0,027s
PLP		56,13s		301,84s		3,503s
RLNP	12,575s	13,205s	29,272s	220,18s	18,037s	12,153s

Table 1: Execution times of original and modified techniques. $pre(t)$ and $proj(t)$ are the execution times for preprocessing and projection steps, respectively. The original techniques do not require the preprocessing step. mCPU Fastmap* is a multi-threaded CPU implementation of Fastmap*.

Figure 7: Using relevant and non-relevant attributes to reproject a data selection at timestep 90 (VisContest 2008 data). (a) Projection obtained using the attributes identified as relevant by the Multilayer Perceptron feature selector. (b) and (c) show projections of less relevant attributes. Each depicts a selection of one of the two visible clusters.

6. Conclusion and Future Work

We have introduced novel strategies for exploring multivariate time-varying volumes based on multidimensional projections.

Because volume data is typically related with time-

varying simulations, we addressed the issue of preserving temporal coherence in projecting a sequence of volumes derived from a simulation, reformulating the original Fastmap and PLP algorithms to preserve temporal coherence when projecting time-varying data. Fastmap was modified with a pre-processing step that selects pivots from a representative sample of the whole data set. PLP was modified by introducing a hierarchy that enables selecting representative control points, again from the whole temporal data, that enable producing coherent projections over time while showing better *Stress* than other time-varying projections.

Regarding interaction capabilities, we introduce projection-based functionalities to assist users in tasks that require answering questions on how specific data attributes related with the global behavior of data. We introduced the *Scatter Projection* visualization, a hybrid between a projection and a Scatterplot. To improve user capability of identifying groups, another interaction facility has been introduced that generates a less-cluttered projection of a given user-selection. This is achieved by employing attribute selection algorithms that allow identifying the attributes that improve groups separation.

We are currently investigating how to adapt our approaches to solve other problems of the exploration of multivariate volumes and adapting new *state-of-art* MPs to support the temporal coherence of data.

References

- [AK91] AHA D., KIBLER D.: Instance-based learning algorithms. *Machine Learning* 6 (1991), 37–66. [6](#)
- [CDZ*09] CHEN W., DING Z., ZHANG S., MACKAY-BRANDT A., CORREIA S., QU H., CROW J. A., TATE D. F., YAN Z., PENG Q.: A novel interface for interactive exploration of dti fibers. *IEEE Transactions on Visualization and Computer Graphics* 15 (November 2009), 1433–1440. [2](#)
- [Cha96] CHALMERS M.: A linear iteration time layout algorithm for visualising high-dimensional data. In *Proceedings of the 7th conference on Visualization '96* (Los Alamitos, CA, USA, 1996), IEEE Computer Society Press, pp. 127–ff. [2](#)
- [Ead84] EADES P. A.: A heuristic for graph drawing. In *Congressus Numerantium* (1984), vol. 42, pp. 149–160. [2](#)

- [Elv92] ELVINS T. T.: A survey of algorithms for volume visualization. *SIGGRAPH Comput. Graph.* 26 (August 1992), 194–201. [1](#)
- [FL95] FALOUTSOS C., LIN K.-I.: Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 1995), SIGMOD '95, ACM, pp. 163–174. [1](#), [2](#)
- [GW11] GU Y., WANG C.: Transgraph: Hierarchical exploration of transition relationships in time-varying volumetric data. *IEEE Transactions on Visualization and Computer Graphics* 17 (Dec. 2011), 2015–2024. [1](#)
- [Hal98] HALL M. A.: *Correlation-based Feature Subset Selection for Machine Learning*. PhD thesis, University of Waikato, Hamilton, New Zealand, 1998. [6](#)
- [IANS10] II J. D., ANDERSON E. W., NONATO L. G., SILVA C. T.: Interactive vector field feature identification. *IEEE Trans. Vis. Comput. Graph.* 16, 6 (2010), 1560–1568. [2](#)
- [IMO09] INGRAM S., MUNZNER T., OLANO M.: Glimmer: Multilevel mds on the gpu. *IEEE Trans. Vis. Comput. Graph.* 15, 2 (2009), 249–261. [5](#)
- [IoAM11] INSTITUTE OF APPLIED MECHANICS CLAUSTHAL UNIVERSITY G. D. W.-I. A. L.: Competition data set and description, 2011. In 2011 IEEE Visualization Design Contest. [6](#)
- [JDL09] JIANU R., DEMIRALP C., LAIDLAW D.: Exploring 3d dti fiber tracts with linked 2d representations. *IEEE Transactions on Visualization and Computer Graphics* 15 (November 2009), 1449–1456. [2](#)
- [JGB*11] JOIA P., GOMEZ NIETO E., BOTELHO G., BATISTA NETO J. A., PAIVA A., NONATO L. G.: Projection-based Image Retrieval using Class-Specific Metrics. In *Proceedings SIB-GRAPI* (Maceio, 2011), Lewiner T., Torres R., (Eds.), Conference on Graphics, Patterns and Images, 24. (SIBGRAPI), IEEE Computer Society Conference Publishing Services. [5](#)
- [JPC*11] JOIA P., PAULOVIK F., COIMBRA D., CUMINATO J., NONATO L.: Local affine multidimensional projection. *Visualization and Computer Graphics, IEEE Transactions on* 17, 12 (dec. 2011), 2563–2571. [2](#)
- [LLR09] LINSSEN L., LONG T. V., ROSENTHAL P.: Linking multi-dimensional feature space cluster visualization to surface extraction from multi-field volume data. *IEEE Computer Graphics and Applications* 29, 3 (2009), 85–89. [linsenlongrosenthalvcglsmoothvis](#). [7](#)
- [PEP*10] PAULOVIK F. V., ELER D. M., POCO J., NONATO L. G., BOTH A. C. P., MINGHIM R.: *A Fast Projection Technique and its Applications to Visualization of Large Dat Sets*. Tech. Rep. 349, Instituto de Ciências Matemáticas e de Computação (ICMC), Universidade de São Paulo (USP), March 2010. [2](#)
- [PEP*11] PAULOVIK F. V., ELER D. M., POCO J., BOTH A. C. P., MINGHIM R., NONATO L. G.: Piece wise laplacian-based projection for interactive data exploration and organization. *Comput. Graph. Forum* 30, 3 (2011), 1091–1100. [1](#), [3](#), [4](#)
- [PEPM12] POCO J., ELER D. M., PAULOVIK F., MINGHIM R.: Employing 2d projections for fast visual exploration of large fiber tracking data (to appear). [2](#)
- [PNML08] PAULOVIK F. V., NONATO L. G., MINGHIM R., LEVKOWITZ H.: Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics* 14 (2008), 564–575. [3](#)
- [PNSV05] PANG-NING T., STEINBACH M., VIPIN K.: *Introduction to Data Mining, (First Edition)*, first edit ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. [6](#)
- [PSN10] PAULOVIK F. V., SILVA C. T., NONATO L. G.: Two-phase mapping for projecting massive data sets. *IEEE Transactions on Visualization and Computer Graphics* 16 (2010), 1281–1290. [2](#), [4](#)
- [SAH00] SCHROEDER W., AVILA L., HOFFMAN W.: Visualizing with vtk: a tutorial. *Computer Graphics and Applications, IEEE* 20, 5 (sep/oct 2000), 20–27. [2](#)
- [SWdO*11] SANTOS T. S. R., WONG C., DE OLIVEIRA M. C. F., MINGHIM R., ELER D. M.: 2011. in IEEE. Visualization Contest 2011. Mini-paper contribution. [2](#), [7](#)
- [WN08] WHALEN D., NORMAN M. L.: Competition data set and description, 2008. In 2008 IEEE Visualization Design Contest. [6](#)
- [XK01] XING E. P., KARP R. M.: CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. *Bioinformatics* 17, suppl_1 (June 2001), S306–315. [5](#)
- [YH38] YOUNG G., HOUSEHOLDER: Discussion of a set of points in terms of their mutual distances. In *Psychometrika* (1938), vol. 3, pp. 19–22. [2](#)