Visualização de fibras neurais usando projeções multi-dimensionais

Jorge Luis Poco Medina

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: _____

Assinatura: _____

Visualização de fibras neurais usando projeções multi-dimensionais

Jorge Luis Poco Medina

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

**USP - São Carlos
Agosto/2010**

A meus pais, Jorge e Marina.
A meus avós Herminio, Emilia e Norberta.
E a meu avô Martin Medina que desde
o céu sempre guia meus passos.

Agradecimentos

Esta dissertação é resultado de muito esforço e dedicação, e não poderia deixar de registrar aqui minha sincera gratidão a todos que estiveram ao meu lado durante o desenvolvimento deste trabalho.

Primeiramente a Deus, pois sem sua ajuda, nada teria sido possível e por me dar forças para lutar pelo que acredito.

A minha orientadora, Professora Dra. Rosane Minghim, pelo seu permanente apoio e atenção dispensada no decorrer deste trabalho. À sua disponibilidade irrestrita, sua forma amiga, exigente e crítica, fundamental contribuição no meu crescimento enquanto pesquisador.

A meu colega e agora professor, Fernando Paulovich, pelos conselhos dados no desenvolvimento deste trabalho.

Aos meus pais, pela sólida formação que me foi dada. E a minhas irmãs, pela alegria que me proporcionaram quando eu voltava pra casa.

A Rosario, minha eterna companheira, pela compreensão e amor brindado durante os momentos mais difíceis nestes dois anos.

Aos professores e colegas do grupo de Visualização, Imagens e Computação Gráfica (VICG - ICMC/USP).

A CAPES, pelo apoio financeiro.

Sumário

Agradecimentos	iii
Resumo	x
Abstract	xi
1 Introdução	1
1.1 Considerações Iniciais	1
1.2 A visualização e suas sub-áreas	2
1.3 Contexto e Motivação	5
1.4 Resultados	9
1.5 Organização do trabalho	9
2 Diffusion Tensor Imaging (DTI) e sua visualização	11
2.1 Considerações Iniciais	11
2.2 Conceitos Básicos	11
2.3 Visualização DTI	13
2.3.1 Índices escalares	13
2.3.2 <i>Rendering</i> Volumétrico	14
2.3.3 Tensor Glyphs	16
2.3.4 Rastreo de Fibras (<i>Fiber Tracking</i>)	16
2.4 Agrupamento de Fibras	17
2.4.1 Métodos de Agrupamento	18
2.4.2 Medidas de similaridade	18
2.5 Considerações Finais	20
3 Visualização baseada em Projeções e sua evolução para aplicação em Visualização de Fibras	21
3.1 Considerações Iniciais	21
3.2 Da LSP para a P-LSP	23
3.2.1 <i>Least Square Projection</i> (LSP)	23
3.2.2 <i>Piecewise Least Square Projection</i> (P-LSP)	24
3.3 Visualização de Volumes usando Projeções	25
3.4 Projeções em 3D	27
3.5 Considerações Finais	29

4	Exploração de Fibras Neurais usando Projeções	30
4.1	Considerações Iniciais	30
4.2	Metodologia	32
4.2.1	Conjunto de dados	33
4.2.2	Conversão de DTI para Fibras Neurais	33
4.2.3	Extração de Características das Fibras Neurais	33
4.2.4	Projeção das Fibras Neurais	35
4.3	Resultados das Projeções de Fibras Neurais	35
4.3.1	Comparações da Qualidade das Projeções	36
4.3.2	Comparações em Tempo	38
4.4	Visualização das Fibras como Superfícies	38
4.5	Sistema de Visualização de Fibras	40
4.6	Considerações Finais	40
5	Conclusões e Trabalhos Futuros	45
A	Apêndice A	54
B	Apêndice B	63

Lista de Figuras

1.1	Métodos de Visualização Científica.	3
1.2	Métodos de Visualização de Informação.	4
1.3	Tela principal do sistema WEAVE; no lado superior esquerdo tem-se a visualização 3D; no lado superior direito o <i>scatterplot</i> de duas características; embaixo um histograma (Gresh et al., 2000).	6
1.4	Tela principal do sistema SimVis (Doleisch et al., 2003).	6
1.5	Os quatro tipos de visualizações apresentadas em Blaas et al. (2007).	7
1.6	Tela principal do aplicativo apresentado em Blaas et al. (2008).	8
1.7	Visualização coordenada da árvore de agrupamentos hierárquicos (Linsen et al., 2008a).	8
2.1	Movimento anisotrópico (direita) vs. movimento isotrópico (esquerda). No caso isotrópico a difusão é similar em todas as direções, no caso anisotrópico a difusão é maior em uma direção que nas outras (Berenschot, 2003).	12
2.2	Mapa de coordenadas baricêntricas (Berenschot, 2003).	15
2.3	Visualização de escalares obtidos a partir dos tensores de difusão (Vilanova et al., 2005).	15
2.4	Rendering volumétrico direito (Kindlmann et al., 2000).	15
2.5	Glifos com forma de caixas (esquerda) e elipsóides (direita) (Moberts, 2005).	16
2.6	Rastreio de fibras em um cérebro usando ROIs (Vilanova et al., 2005).	17
2.7	Fibras do cérebro criadas pelo algoritmo de rastreio de fibras (Brun et al., 2003).	17
2.8	Uma visão esquemática dos maiores fardos de fibras no cérebro (Brun et al., 2003).	19
3.1	Projeção de uma coleção de 674 documentos. Cada documento contém título, autores e resumo de um trabalho científico publicado em uma de três áreas do conhecimento. A projeção por LSP faz um bom trabalho em separar as áreas e localizar grupos de artigos fortemente correlacionados. Uma técnica de geração automática de tópicos identifica essas três áreas (Paulovich et al., 2008).	24
3.2	Projeções de diferentes conjuntos de características extraídas da mesma imagem. Os pixels e os pontos sobre as projeções são coloridos com base em um algoritmo de agrupamento. As características de Gabor fazem um trabalho melhor em separar o tumor, enquanto que ambos os conjuntos de características separam o fundo do objeto principal (e da sua fronteira) (Paulovich et al., 2010).	26
3.3	Projeções P-LSP dos dados de uma cabeça. As projeções mostram boa separação entre estruturas diferentes (Paulovich et al., 2010).	27

3.4	Projeção, agrupamento e geração de superfícies no espaço visual da LSP 3D para um conjunto de documentos de quatro áreas do conhecimento.	28
4.1	Janela principal da ferramenta desenvolvida por Chen et al. (2009)	31
4.2	Janela principal da ferramenta desenvolvida por Jianu et al. (2009)	32
4.3	Passos principais do processo de transformação das fibras num espaço de dimensionalidade 2D ou 3D.	32
4.4	Duas visões de 250k fibras.	36
4.5	Três visões de um conjunto de 20k fibras.	37
4.6	As fibras usadas para gerar estas imagens foram obtidas da pagina web do projeto (Jianu et al., 2009).	42
4.7	Fibras cerebrais obtidas com dados obtidos de http://www.cad.zju.edu.cn/home/chenwei/interface	43
4.8	20k fibras, a cor das instâncias significam, a classe à que pertencem. Dados obtidos de http://pbc.lrdc.pitt.edu/	44

Lista de Tabelas

2.1	Alguns índices escalares para dados DTI. $\ D\ = \sqrt{D:D}$ é a normal do tensor, $Var(\lambda)$ e $E(\lambda)$ são a variância e o valor esperado dos três autovalores, a_n e <i>normalize weighting factor</i> , $A = D - \langle D \rangle I$ (Vilanova et al., 2005).	14
4.1	Comparação em tempo para a redução de dimensionalidade.	38

Resumo

Neste trabalho apresentamos uma nova abordagem para a exploração de fibras neurais a partir de imagens de tensores de difusão. A estratégia combina técnicas de visualização de informação e visualização científica para obter uma rápida e precisa interpretação das fibras. Para isto fazemos uma transformação das fibras para vetores, e utilizamos uma nova técnica de projeção multi-dimensional (P-LSP), para trabalhar com conjuntos grandes de dados. A exploração do espaço das fibras é feita através desta projeção. Além disso, é apresentada a extensão das técnicas LSP e P-LSP para criar projeções em 3D, assim como estratégias que permitem interagir com pontos em 3D. Outra contribuição deste trabalho é a modificação de um método apresentado para criar superfícies de densidade fechadas sobre pontos esparsos. Esta modificação torna possível criar superfícies sobre conjuntos de pontos maiores com uma qualidade aceitável, o que é utilizado para representar conjunto de fibras como uma superfície. Esta nova abordagem é comparada com trabalhos similares mostrando nossas vantagens em termos de tempo de processamento, qualidade e funcionalidades para analisar esta categoria de dados.

Abstract

This work presents a novel approach for the exploration of neural fibers extracted from Diffusion Tensor Images. The developed strategy combines techniques from information and scientific visualization in order to attain a fast and precise interpretation of fiber sets. The approach transforms fibers into vectors from which a new multidimensional projection technique (P-LSP) capable of handling large data sets. The fiber space is explored through the projection. Additionally this work extends P-LSP and LSP projections to 3D, and defines strategies to interact with 3D sparse points. Another contribution of this work is the extension of a method to create close density surfaces over the sparse space generated by the projections. The visualization approach is compared with other similar work showing advantages in processing time, quality and exploration capability to analyze this type of data sets.

Introdução

1.1 Considerações Iniciais

A área de visualização busca criar imagens que transmitam informação sobre dados subjacentes ou processos ([Hansen e Johnson, 2005](#)). Este campo foi adequadamente identificado no *National Science Foundation's Visualization in Scientific Computing Workshop report (1987)* ([McCormick, 1987](#)), que define:

“Visualização é um método computacional. Ele transforma dados simbólicos em geométricos, possibilitando que pesquisadores observem suas simulações e computações. A visualização oferece um método para ver o que não está explicitamente visível. Ela enriquece o processo de análise científica e promove descobertas profundas e não esperadas. Em vários campos a visualização já está revolucionando o jeito como cientistas fazem ciência[...]. O objetivo da visualização é alavancar os métodos científicos existentes provendo novas formas de descoberta por meio de métodos visuais ([McCormick, 1987](#))”

Dentre os temas referentes à visualização, encontrar representação visuais e processos interativos para lidar com problemas onde os itens de dados são complexos (e.g. dados multivariados, vetores, tensores) tem gerado desafios aos pesquisadores da área. Este trabalho integra técnicas de visualização de dados multivariados (em particular mapeamento de pontos no plano) àquelas de visualização de dados volumétricos para obter um arcabouço de interação com dados tensoriais advindos de DTI (*diffusion tensor images*) e fibras neurais.

1.2 A visualização e suas sub-áreas

Entende-se por Visualização o processo de mapeamento de dados e informações em um formato gráfico, através de representações visuais e mecanismos de interação, fazendo uso de suporte computacional e buscando a ampliação da cognição (A.Branco, 2003). O trabalho de Card [Card et al. \(1999\)](#) exprime com concisão essa idéia: “o propósito da visualização é a percepção (*insight*), não figuras”(trad.); sendo que os principais objetivos dessa percepção são a descoberta, a tomada de decisões e o entendimento.

Convencionalmente, a área de Visualização é subdividida em três sub-áreas: visualização científica, visualização de informação e visualização de software (Oliveira e Minghim, 1997). São de especial interesse para este trabalho as técnicas de visualização científica e de informação.

A visualização científica, tradicionalmente definida como a representação visual de dados e informações com características intrinsecamente geométricas, evoluiu concomitantemente com os avanços da informática e da computação gráfica, de onde congrega suas técnicas. Suas aplicações visam, sobretudo, possibilitar a análise visual e a exploração de dados coletados de fenômenos naturais ou obtidos a partir de simulações. Uma de suas características fundamentais é o tratamento de grandes quantidades de dados de natureza m -dimensional, provendo poder analítico para a realização de tarefas que são consideradas complexas de serem realizadas em sua forma original. Como consequência, sua aplicação ocorre de maneira natural nas mais diversas áreas do conhecimento humano (da Silva, 2006). Exemplos típicos de técnicas para visualização de dados científicos são a “extração de superfícies” (Paiva et al., 1999), o “*rendering* volumétrico direto” (Paiva et al., 1999), as linhas de escoamento, etc.. Na Figura 1.1 podemos ver alguns exemplos nesta categoria.

Embora bem definida como disciplina em termos de teoria e algoritmos, a visualização científica ainda lida com vários problemas que demandam mais pesquisa e soluções criativas. De particular interesse para este trabalho estão os problemas de tempo de interação, seleção de elementos de interesse e visualização de ítems complexos, como os tensores de difusão ou as fibras neurais, presentes em conjuntos de dados volumétricos de grande porte (centenas de milhares a milhões de voxels ou pixels), produzidos por equipamentos de ressonância magnética.

Em contrapartida à visualização científica, cuja visão dos dados deve acompanhar a geometria esperada do objeto, a área de *visualização de informação* (Card et al., 1999) trata da visualização de dados abstratos, cujo conteúdo pode variar de coleções de registros a dados não estruturados como documentos e resultados de busca na Web. Em geral esses dados são convertidos em espaços vetoriais (matrizes $\text{ítems} \times \text{atributos}$) de alta dimensionalidade, isto é, um conjunto com grande quantidade de atributos ou dimensões. Alguns exemplos destas técnicas são: *Scatterplots* 2D e 3D, *heat maps* (Witten e Frank, 2002), *height maps*, *table lens* (Rao e Card, 1994), coordenadas paralelas (Inselberg, 1985), *glyphs*, e *radviz* (Hoffman et al., 1999), etc. Na Figura 1.2 podemos ver alguns resultados aplicando estas técnicas.

(a) Mapeamento por cores (Hansen e Johnson, 2005). (b) Mapas de contornos (Hansen e Johnson, 2005). (c) Renderização Volumétrica Direta (Hansen e Johnson, 2005)

(d) Linhas de escoamento (Oliveira e Minghim, 1997)

(e) Visualização de um campo tensorial usando elipses (Westin et al., 1999)

Figura 1.1: Métodos de Visualização Científica.

(a) Scatterplot 2D e 3D (Grinstein et al., 2001).

(b) HeatMaps (Weinstein et al., 1997).

(c) Table Lens (Rao e Card, 1994).

(d) Height Maps de temas de documentos (Grinstein et al., 2001).

(e) Coordenadas Paralelas (Grinstein et al., 2001).

(f) Glyphs, cada atributo é mapeado como uma característica dos rostos (Chernoff (1973).

(g) RadViz (Grinstein et al., 2001).

Figura 1.2: Métodos de Visualização de Informação.

O trabalho por Grinstein et al. (2001) apresenta uma descrição detalhada dos métodos mencionados acima, mas é preferível que o leitor se dirija às referências descritas por cada uma das técnicas para um melhor entendimento das mesmas.

1.3 Contexto e Motivação

Visualização de volumes enfrenta o problema de manipular multi-campos (vários valores naturais - escalares, vetores ou tensores) por voxel assim como um número grande de pontos ou voxels. O processo de definir uma maneira de exploração completa do volume é um dos maiores problemas na visualização. Seleção de regiões ou objetos de interesse assim como a definição de uma função de transferência correta são tarefas que o analista tem que realizar para extrair toda a informação necessária do volume. Devido à habilidade das técnicas de visualização de informação para explorar informação multi-dimensional, esforços vem sendo feitos de forma a empregar tais técnicas na análise das associações entre os dados amostrados e o espaço do objeto (*object space*).

Um dos primeiros exemplos de coordenação explícita entre técnicas de visualização de informação e visualização volumétrica¹ foi o sistema WEAVE (Gresh et al., 2000). O WEAVE fornece uma coordenação entre uma visualização 3D e representações estatísticas. Todas as visões são coordenadas, e as representações estatísticas suportam *linked brushing*². Porém, o espaço do objeto 3D não permite *linked brushing* e seleção. A Figura 1.3 mostra a tela principal do sistema WEAVE.

Um outro arcabouço, conhecido como SimVis (Doleisch et al., 2003), foi desenvolvido com as idéias do WEAVE. Ele combina visões de visualização científica com visões de visualização de informação. Além disso, o trabalho apresenta uma linguagem de definição de características (*feature definition language* - FDL) que formaliza o sub-conjunto de dados que pode ser instanciado pelo *linked brushing*. A Figura 1.4 mostra a tela principal do sistema SimVis.

O trabalho de Blaas et al. (2007) adota a abordagem dos anteriores, combinando os métodos de visualização científica e de informação, mas faz as seguintes contribuições: (1) apóia uma definição visual interativa de projeções 2D arbitrárias de um conjunto de dados n -dimensional; (2) permite *linked brushing* e seleção nas visões da visualização científica; e (3) permite a seleção de características nas visões de visualização de informação para serem coordenadas diretamente nas outras visões. Além disso eles integram técnicas de análise de padrões que podem ser interativamente aplicadas para melhorar o processo de exploração de dados. A Figura 1.5 mostra os quatro tipos de visualizações apresentadas nesse trabalho.

A técnica de coordenadas paralelas é comumente utilizada em visualização de informação. O trabalho de Blaas et al. (2008) apresenta técnicas que melhoram coordenadas paralelas na exploração de grandes volumes de dados variantes no tempo, contendo 10 milhões de pontos por *timestep*.

¹sub-área da visualização científica que trata conjuntos de dados 3D

²Método de interação no qual o usuário pode selecionar um conjunto de itens numa visão e estes são marcados em todas as visões que compartilham os itens

Figura 1.3: Tela principal do sistema WEAVE; no lado superior esquerdo tem-se a visualização 3D; no lado superior direito o *scatterplot* de duas características; embaixo um histograma (Gresh et al., 2000).

Figura 1.4: Tela principal do sistema SimVis (Doleisch et al., 2003).

Figura 1.5: Os quatro tipos de visualizações apresentadas em [Blaas et al. \(2007\)](#).

Nesse trabalho são descritas técnicas de pré-processamento usando quantização e compressão de dados, para uma rápida renderização, e técnicas de *brushing* para uma seleção interativa entre múltiplas visões, incluindo uma visão 3D do espaço do objeto. A Figura 1.6 mostra a tela principal do aplicativo apresentado naquele trabalho.

O trabalho por [Linsen et al. \(2008a\)](#) apresenta uma visualização baseada na extração de superfícies de volumes de dados de partículas multi-campo. Neste trabalho, a exploração no espaço de características é feita por um método de agrupamento hierárquico. No *cluster tree layout* o usuário pode selecionar os agrupamentos de interesse. A seleção de um grupo no espaço de características corresponde a uma superfície no espaço do objeto. Os agrupamentos são representados por discos, onde o raio do disco codifica o tamanho do agrupamento, e sua cor codifica a posição dentro do *radial layout*, (ver Figura 1.7).

O que os trabalhos mencionados acima têm em comum com este trabalho é a integração de técnicas das duas sub-áreas de visualização, uma vez que conjuntos de dados atuais incorporam características geométricas e multivariadas na sua definição. No caso de dados complexos, como os tensoriais, a definição de uma abordagem de exploração baseada em características bem como o tratamento de dados volumosos estão na linha de frente dos problemas que impedem a evolução da área.

Neste trabalho propomos o uso de um tipo de visualização de dados tensoriais da área médica, baseado na criação de fibras neurais a partir de tensores de difusão, acoplado com uma técnica de projeção multi-dimensional 2D ou 3D, para apoiar a seleção de partes de interesse no volume, um grande desafio da visualização volumétrica convencional.

Em particular, dentre as técnicas de visualização de dados abstratos, as técnicas de projeção multi-dimensional, ou de posicionamento de pontos no plano, têm sido objeto de estudo e desenvolvimento do grupo de pesquisa em Visualização no ICMC/USP. Dentre os resultados obtidos,

Figura 1.6: Tela principal do aplicativo apresentado em [Blaas et al. \(2008\)](#).

Figura 1.7: Visualização coordenada da árvore de agrupamentos hierárquicos ([Linsen et al., 2008a](#)).

existem trabalhos para o tratamento de dados não estruturados de várias naturezas, como coleções de documentos (Paulovich et al., 2008; Cuadros et al., 2007; Paulovich et al., 2006), coleções de imagens (Eler et al., 2008), e séries temporais (Alencar et al., 2008).

Nesse contexto, o presente trabalho integra as técnicas de projeção, ou mapeamento no plano, às visualizações científicas de dados oriundos da área médica de natureza tensorial. Com o uso da abordagem de projeções como forma de seleção de parte de interesse no volume, foi possível desenvolver uma estratégia de interação com conjuntos de fibras cerebrais semelhantes. Além disso, a extensão de uma das técnicas de projeção para lidar com conjuntos grandes de dados permitiu, neste trabalho, que um volume muito maior de fibras fosse manipulado.

1.4 Resultados

O principal objetivo deste trabalho foi estudar e tratar os problemas de exploração por projeção de dados tensoriais médicos.

Como resultado principal desenvolvemos uma técnica para permitir que, durante a exploração de dados tensoriais médicos, visualizados como fibras representativas de conexões do cérebro, o analista fosse capaz de selecionar estruturas de interesse baseadas em similaridade, em tempo compatível com a análise de conjuntos de milhares de fibras.

Além disso, como resultados adicionais em colaboração com outros membros do grupo de pesquisa, podemos citar:

- Extensão da técnica LSP, gerando a LSP por partes (P-LSP) para permitir trabalhar com um número maior de elementos.
- Extensão das técnicas LSP e P-LSP, que geram convencionalmente mapeamentos para o espaço 2D, para trabalhar em 3D, desenvolvendo também novas formas de interação com projeções em 3D.
- Visualização de conjuntos de fibras similares como um só objeto (usando superfícies de densidade) no espaço 3D.

Um sistema está disponível que incorpora todas essas técnicas e as contribuições permitem levar a análise de fibras neurais geradas a partir de dados tensoriais a uma nova plataforma de desenvolvimento.

1.5 Organização do trabalho

O restante deste texto está dividido da seguinte forma: No Capítulo 2 mostramos um resumo de como são obtidas as imagens tensoriais na área médica, e como elas são visualizadas.

No Capítulo 3 apresentamos uma revisão inicial sobre visualização baseada em projeções multi-dimensionais. No Capítulo 4 é descrita a abordagem usada neste trabalho para a interação de fibras neurais usando projeções multi-dimensionais e sua implementação. Finalmente no Capítulo 5, apresentamos as conclusões e trabalhos futuros.

Diffusion Tensor Imaging (DTI) e sua visualização

2.1 Considerações Iniciais

As imagens por tensores de difusão (*Diffusion Tensor Imaging (DTI)*) permitem ter uma idéia das estruturas presentes em tecidos do corpo humano, representando um tipo de dado que tem despertado grande interesse da comunidade de visualização. Em particular, DTI do cérebro possibilita a observação de fibras e da sua estrutura dentro do cérebro.

Este capítulo é dedicado às imagens geradas por tensores de difusão e as técnicas que são usadas na sua visualização. Na Seção 2.2 apresentamos os conceitos básicos de DTI. Em seguida, na Seção 2.3, apresentamos uma revisão geral das técnicas de visualização de dados tensoriais. Na Seção 2.4 descrevemos brevemente os algoritmos de agrupamento de fibras assim como as diferentes técnicas para se medir a similaridade entre fibras. Finalmente na Seção 2.5 são apresentadas as conclusões e considerações finais do capítulo.

2.2 Conceitos Básicos

Diffusion Tensor Imaging é uma técnica que quantifica a difusão média do movimento das moléculas de água em tecidos biológicos (Le Bihan, 2003). O movimento aleatório das moléculas de água é causado pela energia térmica interna e é conhecido como movimento Browniano (*Brownian*). Certos tecidos limitam o movimento das moléculas de água, reduzindo a distância

que elas viajam. Medindo a direção preferida de difusão é possível reconstruir a estrutura do tecido (Moberts, 2005).

Tecidos que permitem que as moléculas viajem mais facilmente em certa direção do que em outras são chamados anisotrópicos. Um exemplo de tecido anisotrópico é a matéria branca no cérebro. A matéria branca consiste de fibras que conectam a matéria cinzenta (Moberts, 2005). Na matéria branca a água é mais difusa na direção das fibras que na direção perpendicular a elas. Outros tipos de tecidos também apresentam comportamento anisotrópico, como por exemplo o músculo cardíaco, mas neste trabalho só serão considerados DTI do cérebro, pelas suas particularidades. A Figura 2.1 ilustra a diferença entre movimento anisotrópico e isotrópico.

Figura 2.1: Movimento anisotrópico (direita) vs. movimento isotrópico (esquerda). No caso isotrópico a difusão é similar em todas as direções, no caso anisotrópico a difusão é maior em uma direção que nas outras (Berenschot, 2003).

Na difusão anisotrópica, os dados não podem ser simplesmente representados por um escalar. Nesse caso, é necessário o emprego de um tensor simétrico 3×3 :

$$D = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix} \quad (2.1)$$

A diagonalização deste tensor dá três autovalores positivos λ_1 , λ_2 , λ_3 ordenados em ordem decrescente e seus correspondentes autovetores e_1 , e_2 , e_3 . Os autovetores são ortogonais e representam as três principais difusões no voxel. Junto com os autovalores, eles definem a informação completa de um tensor.

Existem diversas técnicas para a visualização de campos tensoriais. A seguir apresentamos algumas delas.

2.3 Visualização DTI

Uma forma de visualização de campos tensoriais que consiga representar a maior quantidade possível de informação se mostra um dos grandes desafios da visualização científica, uma vez que tais dados são multi-valorados e apresentam uma complexa inter-relação entre esses valores, bem como entre os tensores. Nas últimas décadas foram propostas diversas técnicas para visualizar campos tensoriais, sendo a maioria baseada na redução de dimensionalidade para exibir as informações “mais importantes” sobre os dados.

Para se classificar essas técnicas, dois tipos de critérios são normalmente empregados. No primeiro, uma técnica de visualização é classificada de acordo com a dimensão na qual os tensores são representados (por meio de uma redução de dimensionalidade). No outro, as técnicas são classificadas de acordo com a capacidade de apresentar informação local ou global, onde global significa a relação espacial entre os tensores. A seguir, apresentamos algumas técnicas classificadas segundo esses critérios.

2.3.1 Índices escalares

A complexidade dos dados tensoriais requer um esquema complicado para a visualização, porém as pessoas que trabalham com imagens médicas estão acostumadas a trabalhar com imagens escalares em escala de cinza. Os conjuntos de dados escalares, apesar da limitação da informação que eles podem mostrar, podem ser visualizados de forma fácil e clara, levando a uma interpretação rápida e segura.

O desafio de reduzir os tensores a dados escalares inclui o mapeamento da maior quantidade de informação significativa, reduzindo os efeitos do ruído que pode existir. Alguns índices escalares para conjunto de dados DTI são mostrados na Tabela 2.1.

A *Mean diffusivity* (MD), que mede a taxa global de difusão, consiste na média dos autovalores do tensor de difusão e é invariante à rotação. Este escalar é normalmente utilizado na localização de lesões na matéria branca que não são mostrados em outras formas de ressonância magnética.

Antes que o modelo de tensor de difusão seja formulado explicitamente, muitos índices para DTI foram usados, tal como *anisotropic diffusion ratio* (Douek et al., 1991). Infelizmente, esses índices dependem da escolha de um sistema de coordenadas de laboratório e são variantes às rotações. Entre os índices invariantes à rotação, temos o *volume ratio* (VR), *rational anisotropy* (RA) ou *fractional anisotropy* (FA). Note que o RA e o FA podem ser derivadas da norma do tensor e do seu *traço* sem ser necessário o cálculo dos autovalores.

O trabalho de Pierpaoli e Bassar (1996) calculou um índice anisotrópico inter-voxel, o *lattice index* (LI). O LI calcula a média do produto interno entre dois tensores de difusão de voxels vizinhos. Ele é menos sensível a ruído e evita a subestimação da anisotropia quando os voxels vizinhos tem diferentes orientações das fibras.

Tabela 2.1: Alguns índices escalares para dados DTI. $\|D\| = \sqrt{D:D}$ é a normal do tensor, $Var(\lambda)$ e $E(\lambda)$ são a variância e o valor esperado dos três autovalores, a_n e *normalize weighting factor*, $A = D - \langle D \rangle I$ (Vilanova et al., 2005).

Scalar Index	Equations
Mean diffusivity, $\langle D \rangle$	$\frac{D_{xx} + D_{yy} + D_{zz}}{3}, \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}, Trace(D)$
54 Volume radio, VR	$\frac{\lambda_1 \lambda_2 \lambda_3}{(\frac{\lambda_1 + \lambda_2 + \lambda_3}{3})^3}, 27 \frac{Determinante(D)}{Trace(D)^3}$
3 Fractional anisotropic, FA	$\sqrt{\frac{3}{2} \frac{D - \langle D \rangle I}{\ D\ }}$
3 Rational anisotropic, RA	$\frac{\ D - \langle D \rangle I\ }{\ \langle D \rangle I\ }, \frac{\sqrt{Var(\lambda)}}{E(\lambda)}$
54 Lattice index, LI	$\sum_{n=1}^N a_n \left(\frac{3}{8} \frac{\sqrt{A:A_n}}{\sqrt{D:D_n}} + \frac{3}{4} \frac{A:A_n}{\sqrt{D:D_n} \sqrt{D_n:D_n}} \right)$
66 Linear anisotropic, c_l	$\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3}$
Planar anisotropic, c_p	$\frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3}$
Isotropy c_s	$\frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}$

Como os tensores são reduzidos a um escalar, os índices FA, RA e LI não indicam a variação direcional. Por exemplo, tensores representados por elipsóides de formatos completamente distintos podem apresentar o mesmo valor FA. Foram propostas medidas baseadas na geometria da difusão (Pierpaoli e Basser, 1996) como, por exemplo, *linear anisotropy* (c_l), *planar anisotropy* (c_p) e *spherical anisotropy* (c_s), onde a soma desses valores resulta em um. Assim, estas três métricas parametrizam um espaço baricêntrico no qual as três formas extremas (linear, planar e esférica) são os cantos de um triângulo, como mostra a Figura 2.2.

A Figura 2.3 compara qualitativamente algumas das métricas descritas acima.

2.3.2 Rendering Volumétrico

O *Rendering* volumétrico direto é uma técnica onde uma representação geométrica não é explicitamente criada. Ao invés disso, uma função de transferência é usada mapeando os tensores em características visuais como cor, opacidade e tonalidade. O desafio nesse caso é a definição da função de transferência, já que algumas características são mais importantes que outras e não devem ser escondidas por outras de menor importância. Um dos grandes problemas do *rendering* volumétrico é seu custo computacional, além de ser limitado em termos de interação com o usuário. Na Figura 2.4 mostramos um exemplo deste tipo de visualização (Moberts, 2005).

Figura 2.2: Mapa de coordenadas baricêntricas (Berenschot, 2003).

Figura 2.3: Visualização de escalares obtidos a partir dos tensores de difusão (Vilanova et al., 2005).

Figura 2.4: Rendering volumétrico direto (Kindlmann et al., 2000).

2.3.3 Tensor Glyphs

Um glifo (*glyph*) é um objeto geométrico com forma e comprimento definidos pelo tensor. A orientação do glifo é determinada pelo autovetor principal, e a forma pelos autovalores. Glifos podem ter formas básicas como caixas ou elipsóides ou formas mais complexas como os *superquadric tensor glyphs* (Kindlmann, 2004). Os glifos podem ser usados em 2D e 3D, mas por causa da oclusão eles são mais usados em pequenas regiões 2D. Na Figura 2.5 dois tipos de glifos são apresentados e coloridos de acordo com o índice escalar FA (Moberts, 2005).

Figura 2.5: Glifos com forma de caixas (esquerda) e elipsóides (direita) (Moberts, 2005).

2.3.4 Rastreamento de Fibras (*Fiber Tracking*)

O *Rastreamento de Fibras* configura um dos métodos de visualização mais promissores para dados DTI. Na literatura este método é chamado também de *tractography*. O objetivo principal desta técnica é a reconstrução 3D de trajetórias nos dados discretos DTI. Existem dois tipos de algoritmos de rastreamento de fibras: propagação de linhas e minimização de energia (Mori e van Zijl, 2002). A propagação de linhas assume que a maior difusão em um voxel está alinhada com a orientação dos axônios dos neurônios na matéria branca. Começando em um ponto a linha é propagada através do volume na direção do principal autovetor. O método de minimização de energia, por outro lado, procura pelo caminho energeticamente mais favorável entre dois pontos (Moberts, 2005). Existe uma analogia direta entre as *streamlines* e as estruturas a serem visualizadas (por exemplo, as fibras). O rastreamento das *Streamlines* está baseado na resolução da seguinte equação:

$$p(t) = \int_0^t v(p(s)) ds \quad (2.2)$$

onde $p(t)$ é o *streamline* gerado e v corresponde ao campo vetorial gerado pelo principal autovetor e_1 . $p(0)$ é o conjunto de pontos iniciais da curva de integração.

A técnica de rastreamento de fibras tem três passos principais: definição dos pontos iniciais, ou *seed points*, a integração, e a definição do critério de parada. Os *seed points* são geralmente definidos pelo usuário: o usuário define uma ou mais regiões de interesse (ROI). O interior das regiões de

interesse é amostrado e as amostras são usadas como sementes (ver Figura 2.6). A Equação (2.2) é então resolvida por métodos de integração numérica como o de *Euler* ou o de *Runge-Kutta* de segunda e terceira ordem. O critério de parada evita o cálculo das *streamlines* onde o campo vetorial não está robustamente definido.

Figura 2.6: Rastreio de fibras em um cérebro usando ROIs (Vilanova et al., 2005).

2.4 Agrupamento de Fibras

Os algoritmos de rastreio de fibras vistos na seção anterior produzem um conjunto de fibras (ver a Figura 2.7). Nesta seção mostraremos diferentes métodos disponíveis na literatura sobre “medidas de similaridade” entre essas fibras assim como “métodos de agrupamento” para conjuntos de fibras.

Figura 2.7: Fibras do cérebro criadas pelo algoritmo de rastreio de fibras (Brun et al., 2003).

2.4.1 Métodos de Agrupamento

Um algoritmo de agrupamento impõe um tipo de classificação nos dados de entrada. Esta classificação pode ser feita de várias formas. Um algoritmo de agrupamento de **partição** produz uma divisão simples dos dados de entrada [Corouge et al. \(2004\)](#), já um algoritmo **hierárquico** cria divisões aninhadas [Zhang e Laidlaw \(2002\)](#).

Nos algoritmos baseados em grafos, os objetos que vão ser agrupados representam os nós de um grafo não orientado, e as arestas representam a relação entre os nós. Esta relação pode ser baseada na similaridade ou dissimilaridade entre os nós. [Brun et al. \(2003\)](#) usa a técnica, chamada *Laplacian eigenmaps*, para o agrupamento de fibras. No trabalho de [Brun et al. \(2004\)](#) o método utilizado para o agrupamento de fibras é o *normalized cuts*.

2.4.2 Medidas de similaridade

A maioria dos algoritmos de agrupamento define os conjuntos de itens de acordo com algum critério de similaridade. Assim, é preciso existir uma medida de similaridade ou dissimilaridade entre os objetos.

Não há uma maneira padrão de se calcular a proximidade entre pares de fibras. Calcular a proximidade entre um par de *pontos* é relativamente fácil; uma distância Euclidiana nos dá uma indicação da similaridade. Mas as fibras são representadas como uma lista de pontos na qual cada uma tem um comprimento diferente. A seguir apresentaremos algumas formas de cálculo de similaridade entre fibras.

Nas seguintes equações, $\|\cdot\|$ representa a norma Euclidiana.

A **distância dos pontos mais próximos** ([Corouge et al., 2004](#)), fornece informação de dissimilaridade apenas de um par de fibras, aquelas que estão mais próximas entre os conjuntos de pontos de cada fibra. Esta medida é representada por d_c e pode ser calculada pela Equação (2.3).

$$d_c(F_i, F_j) = \min_{p_k \in F_i, p_l \in F_j} \|p_k, p_l\| \quad (2.3)$$

A **média dos pontos mais próximos** [Corouge et al. \(2004\)](#) dá uma informação mais global sobre a dissimilaridade de um par de fibras. Cada ponto de uma fibra é mapeado ao ponto mais próximo da outra fibra, a distância entre este par de fibras é definida como a média das distâncias calculadas anteriormente. Na Equação 2.5 a definição matemática desta medida é apresentada.

$$d_M(F_i, F_j) = \text{mean}(d_m(F_i, F_j), d_m(F_j, F_i)) \quad (2.4)$$

$$\text{com, } d_m(F_i, F_j) = \text{mean} \min_{p_l \in F_i} \min_{p_k \in F_j} \|p_k - p_l\| \quad (2.5)$$

Outra maneira de medir a proximidade entre fibras é com a **distância de Hausdorff** (Corouge et al., 2004). Essa distância é muito conservadora. Duas fibras são consideradas semelhantes só se todas as distâncias entre os pontos mais próximos forem muito pequenas. A distância de Hausdorff é definida como a máxima distância entre dois pares de pontos das fibras. A distância de Hausdorff é apresentada na equação abaixo.

$$d_H(F_i, F_j) = \max(d_h(F_i, F_j), d_h(F_j, F_i)) \quad (2.6)$$

$$\text{com } d_h = (F_i, F_j) = \max_{p_k \in F_i} \min_{p_l \in F_j} \|p_k - p_l\| \quad (2.7)$$

Brun et al. (2003) definem a **distância dos pontos finais**. Essa distância se baseia na idéia de que duas fibras com pontos finais semelhantes deveriam ser consideradas semelhantes. Por exemplo, na Figura 2.8 nós queremos que a fibra com pontos finais (A, A') tenha uma alta similaridade com a fibra com pontos finais (B, B') . No entanto a fibra (C, C') deveria ser considerada dissimilar às fibras (A, A') e (B, B') embora elas compartilhem um ponto final. Para tal, definimos $f_{i,1}$ e $f_{i,end}$ como a primeira e última coordenadas da i -ésima fibra, e calculamos:

$$f_i = (f_{i,1}, f_{i,end}) \quad (2.8)$$

$$\tilde{f}_i = (f_{i,end}, f_{i,1}) \quad (2.9)$$

$$d_E = \begin{cases} 0 & \text{if } i = j \\ \exp(-\frac{\|f_i - f_j\|^2}{2\sigma^2}) + \exp(-\frac{\|f_i - \tilde{f}_j\|^2}{2\sigma^2}) & \text{if } i \neq j \end{cases} \quad (2.10)$$

Figura 2.8: Uma visão esquemática dos maiores fardos de fibras no cérebro (Brun et al., 2003).

O trabalho de Zhang e Laidlaw (2002) define a similaridade entre duas fibras como mostra a Equação ((2.11)). Nesta equação s parametriza o comprimento do arco da fibra curta, e $dist(s)$ é a distância mais curta (Euclidiana) desde a posição s da fibra mais curta até a fibra mais longa. T_t é o umbral, e apenas as distâncias que sejam maiores que o umbral vão contribuir à distancia. Esta medida é chamada **distância sobre o umbral**.

$$d_t = \frac{\int_s \max(\text{dist}(s) - T_t, 0) ds}{\int_s \max(\frac{\text{dist}(s) - T_i}{|\text{dist}(s) - T_i|}, 0) ds} \quad (2.11)$$

O trabalho de Brun et al. (2004) **mapeia as fibras de alta dimensão a uma dimensão menor** e usa um *kernel* para comparar as fibras neste espaço. Primeiro cada fibra é mapeada em um espaço Euclideano de 9 dimensões. Após isso, a média dos pontos da fibra m e a matriz de covariância C são calculadas. Além disso $G = \sqrt{C}$ é encontrado para se evitar um comportamento não-linear. Finalmente a fibra é descrita como:

$$\Phi(F) = (m_x, m_y, m_z, g_{xx}, g_{xy}, g_{xz}, g_{yy}, g_{yz}, g_{zz})$$

E a medida de similaridade é definida por:

$$d_k(\Phi(F_i), \Phi(F_j)) = \exp\left(-\frac{\|\Phi(F_i) - \Phi(F_j)\|^2}{2\sigma^2}\right) \quad (2.12)$$

onde o parametro σ ajusta a sensibilidade da função de similaridade.

2.5 Considerações Finais

Neste capítulo foram apresentados os dados tensoriais gerados por DTI e as técnicas atuais para sua visualização. A técnica mais utilizada com este tipo de dados são o rastreo de fibras, porque eles permitem uma visualização global dos dados, além que dá informação da conectividade das áreas do cérebro.

Também foram apresentadas estratégias de agrupamento de fibras; dentro desta categoria mostramos métodos de agrupamento e técnicas para medir a similaridade entre fibras.

No próximo capítulo descreveremos os métodos de visualização baseada em projeções, e no Capítulo 4 mostraremos como os métodos de projeção são utilizados com as fibras geradas dos dados tensoriais (ver Seção 2.3.4) que é um dos objetivos do nosso trabalho.

Visualização baseada em Projeções e sua evolução para aplicação em Visualização de Fibras

3.1 Considerações Iniciais

Como foi apresentado na introdução deste trabalho, um dos aportes deste projeto foi integrar técnicas de visualização científica com as de visualização de informação, especificamente com as técnicas baseada em projeções, para obtenção de uma estratégia efetiva de exploração de dados volumétricos complexos.

Considere um conjunto de dados D contendo m instâncias representadas por vetores de características num espaço n -dimensional. Uma técnica de projeção multi-dimensional, ou técnica de posicionamento de pontos, pode ser definida por uma função que mapeia cada instância $d_i \in D$ em um elemento gráfico (ex.: ponto, círculo ou outro ícone) em um espaço visual (1D, 2D, ou 3D). A posição relativa dos elementos no *layout* final reflete algum tipo de relacionamento, por exemplo, similaridade entre os itens de dados ([Tejada et al., 2003](#)). Usuários exploram este tipo de visualização através da localização de grupos de instâncias que se formam no *layout*, e que sejam de interesse para sua investigação dos dados.

Dois exemplos clássicos de projeções que podem ser usadas para esse fim são a *Sammon's Mapping* ([Sammon, 1964](#)) e o *Classical Multidimensional Scaling (MDS)* ([Cox e Cox, 2000](#)). O primeiro define uma função-custo com base na diferença entre as distâncias no espaço de características e as distâncias no espaço visual e minimiza tal função. Na segunda, uma matriz de dis-

tâncias entre todos os pares de elementos do conjunto é definida e uma decomposição espectral é aplicada para recuperar as coordenadas cartesianas dos elementos no espaço visual. Apesar dessas serem as técnicas com resultados mais precisos na literatura de projeções, seu custo computacional ($O(n^2)$) impossibilita o uso para conjuntos de dados volumosos.

Uma outra estratégia para posicionamento de pontos no espaço visual, a *Force-Directed Placement (FDP)* (Fruchterman e Reingold, 1991), foi inicialmente concebida para o desenho de grafos. No modelo FDP, cada instância é modelada como uma partícula sobre o efeito de forças de atração ou repulsão em relação às demais partículas. Como toda instância afeta todas as outras, sua complexidade computacional é $O(n^2)$. Para reduzir esta complexidade, Chalmers (Chalmers, 1996) define uma iteração linear onde as forças são determinadas usando amostras ao invés de todas as instâncias de dados. Cada iteração é acelerada, mas ainda n iterações são necessárias, o que não reduz, em última instância, a complexidade computacional do algoritmo. Essa complexidade é posteriormente reduzida no trabalho de Morrison et al. (2002) e Morrison e Chalmers (2004), que criam um modelo híbrido que aproxima a técnica de Chalmers. Nesse modelo, primeiro uma amostra é projetada e as demais são posicionadas por meio de uma interpolação. Isso reduz a complexidade para $O(n \sqrt[4]{n})$. Jourdan e Melançon (2004) otimizam ainda mais o processo, criando uma técnica $O(n \log n)$. Apesar de realmente produzirem aumento de eficiência, essas abordagens mais rápidas perdem muito em precisão no que se refere ao posicionamento dos pontos (instâncias) na vizinhança dos mais similares, um objetivo importante deste tipo de mapeamento.

O nosso grupo de pesquisa desenvolveu recentemente uma outra abordagem, chamada *Least Square Projection (LSP)* (Paulovich et al., 2008) (ver Subseção 3.2.1), que tende a produzir resultados muito bons quanto à preservação de vizinhanças, melhores que as demais quando o número de dimensões (atributos) do conjunto é grande. Apesar do seu tempo de processamento ser razoável, a LSP não consegue, na sua definição inicial, lidar com conjuntos de dados grandes. Parte deste trabalho foi apoiar o desenvolvimento de uma nova técnica, a *Piecewise Least Square Projection (P-LSP)* (ver Subseção 3.2.2), uma extensão da LSP que permite tratar dados em grande quantidade. A próxima Seção apresenta a LSP e sua extensão P-LSP. Exemplos de sua utilização para a aplicação proposta neste trabalho também são apresentados.

Este capítulo é dedicado à apresentação das principais técnicas de projeção multidimensional envolvidas neste trabalho. Na Seção 3.2 mostramos a técnica *Least Square Projection (LSP)* e como ela evoluiu para a técnica *Piecewise Least Square Projection (P-LSP)* lidando com as desvantagens da LSP. Na Seção 3.3 apresentamos alguns resultados obtidos na aplicação das projeções multi-dimensionais na exploração de imagens e volumes da área médica. A Seção 3.4 descreve a extensão da técnica LSP para 3D de forma a melhorar a precisão de discriminação da técnica e as técnicas de interação que foram desenvolvidas para sua exploração. Todas essas extensões fazem parte dos resultados deste mestrado.

Finalmente na Seção 3.5 são apresentadas as conclusões e considerações finais do capítulo.

3.2 Da LSP para a P-LSP

3.2.1 Least Square Projection (LSP)

A técnica *Least Square Projection* 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 visual, determinados, por exemplo, pela distância Euclidiana. Matematicamente, seja $N_i = \{p_{i_1}, \dots, p_{i_{k_i}}\}$ o conjunto dos k_i vizinhos mais próximo dos p_i (assumindo uma medida de distância definida em D), com $(x_{p_{i_j}}, y_{p_{i_j}})$ as coordenadas cartesianas de cada elemento $p_{i_j} \in N_i$ mapeado para $\mathbb{R}^2$. As coordenadas bi-dimensionais de p_i podem ser calculadas por:

$$(x_{p_i}, y_{p_i}) = \sum_{p_{i_j} \in N_i} \alpha_{ij} (x_{p_{i_j}}, y_{p_{i_j}}) \quad (3.1)$$

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

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

$$Lx = 0; \quad Ly = 0 \quad (3.2)$$

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 (3.1).

Os conjuntos N_i definem um *Nearest Neighbor Graph (NNG)* de D , isto é, um grafo conectando cada elemento 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 conexos do NNG (Sorkine e Cohen-Or, 2004). Assim, de forma a se obter uma única solução não trivial para os sistemas (Eq 3.2), o NNG deve ter apenas um componente conexo, o que pode ser garantido através da adição de novas arestas ligando componentes possivelmente desconexos do NNG.

LSP adiciona informação geométrica ao mapa através da adição de *pontos de controle* como restrições aos sistemas (Eq. 3.2). 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 de aplicação da LSP para conjunto de documentos é dado na Figura 3.1. No caso de documentos o espaço de características é extraído por um processo de pré-processamento onde os textos individuais são processados e termos relevantes sofrem um processo de contagem. No caso da Figura 3.1, o mapeamento separou adequadamente as três áreas do conhecimento das quais tratava o conjunto de trabalhos científicos objeto da visualização.

Figura 3.1: Projeção de uma coleção de 674 documentos. Cada documento contém título, autores e resumo de um trabalho científico publicado em uma de três áreas do conhecimento. A projeção por LSP faz um bom trabalho em separar as áreas e localizar grupos de artigos fortemente correlacionados. Uma técnica de geração automática de tópicos identifica essas três áreas (Paulovich et al., 2008).

Os maiores problemas com essa formulação da LSP são: o tempo de processamento para localização dos k vizinhos mais próximos e o tempo de processamento para solução de grandes sistemas lineares quando o conjunto de dados é muito grande.

Para solucionar estes problemas, foi formulada a P-LSP, brevemente descrita abaixo.

3.2.2 *Piecewise Least Square Projection (P-LSP)*

A idéia da P-LSP é particionar o conjunto de dados original em subconjuntos menores nos quais a LSP original possa ser eficientemente empregada.

Para o particionamento dos dados é usado o algoritmo *bisecting k-means* (Steinbach et al., 2000), que desempenha tão bem quanto outros métodos matematicamente bem formados como o PDDP (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 e posicionando 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 num trabalho do grupo em fase de finalização (Paulovich et al., 2010).

O Algoritmo 1 (Paulovich et al., 2010) mostra os passos desta técnica.

Algorithm 1 Piecewise Least Square Projection (P-LSP) (Paulovich et al., 2010).

Require: D : data set to be projected.**Ensure:** the projection of D .

```

1: procedure P-LSP( $D$ )
2:   Create the partitions  $P = \{P_1, \dots, P_k\}$  using the bisecting k-means algorithm, where  $k = \sqrt{n}$ .
3:   Set  $C_P = \emptyset$ 
4:   for all  $P_i \in P$  do
5:     Split  $P_i$  into  $n_i = (|P_i|/n) * n^{0.75}$  clusters,  $P_i = \{P_{i_1}, \dots, P_{i_{n_i}}\}$ , using the bisecting k-means algorithm.
6:     Set  $C_{P_i} = \emptyset$ .
7:     for all  $P_{i_j} \in P_i$  do
8:        $C_{P_i} = C_{P_i} \cup \{w_{P_{i_j}}\}$ , where  $w_{P_{i_j}}$  is the medoid of  $P_{i_j}$ .
9:     end for
10:     $C_P = C_P \cup C_{P_i}$ .
11:  end for
12:  Map  $C_P$  using the original LSP.
13:  for all  $P_i \in P$  do
14:    Map the partition  $P_i$  using the original LSP considering the geometry imposed by  $C_{P_i}$ .
15:  end for
16: end procedure

```

Com a nova formulação, a técnica permite tratar rapidamente conjuntos de dados muito maiores, dando margem à sua utilização para uma aplicação inovadora, a de interação com grandes volumes de dados. Em aplicações como as médicas, 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. Utilizar projeções como intermediárias pode apoiar o processo.

3.3 Visualização de Volumes usando Projeções

No contexto da exploração de volumes, se for possível mapear cada voxel (ou pixel) individualmente numa projeção, uma nova forma de seleção de unidades, por coerência de informação, se revela. Através do mapeamento de funções de similaridade entre voxels e entre pixels é possível selecionar seções do volume (ou imagem) que se deseja visualizar ou extrair.

Para ilustrar o processo de extração de características envolvido nessa tarefa, capaz de representar individualmente os elementos do objeto sob estudo, a Figura 3.2 apresenta projeções diferentes para diferentes características de uma imagem (características extraídas a partir da intensidade dos pixels e características de Gabor (Jain e Farrokhnia, 1990; Manjunath e Ma, 1996)). Em seguida é aplicado um processo de agrupamento por *k-means* sobre a projeção (MacQueen, 1967) para

criar grupos. Finalmente, os pontos correspondentes na imagem são coloridos de acordo com a cor do agrupamento a que eles pertencem. Os resultados são apresentados na Figura 3.2. Em ambas as projeções, o fundo, o cérebro e a fronteira entre eles são separados corretamente e ligeiramente melhores para as características de Gabor. Em termos de detecção do tumor, entretanto, as características de Gabor são muito melhores.

Figura 3.2: Projeções de diferentes conjuntos de características extraídas da mesma imagem. Os pixels e os pontos sobre as projeções são coloridos com base em um algoritmo de agrupamento. As características de Gabor fazem um trabalho melhor em separar o tumor, enquanto que ambos os conjuntos de características separam o fundo do objeto principal (e da sua fronteira) (Paulovich et al., 2010).

Um segundo exemplo de aplicação para dados volumétricos, apresentado na Figura 3.3, projeta os voxels de um volume e coordena isso com a visualização do próprio volume de acordo com a seleção do usuário de um grupo coerente na projeção. Para criar essas projeções se usa um subconjunto do volume original de tomografia da cabeça ¹, considerando da fatia 30 à fatia 97. As fatias restantes não possuem voxels com informação, apenas cor de fundo. A amostra final possui mais de um milhão de voxels. Aqui são usadas características de intensidade e seu gradiente, resultando em 14 atributos por voxel. As Figuras 3.3(a) e 3.3(b) mostram seleções do usuário representando uma estrutura interna do cérebro e a parte externa dele, respectivamente. A seleção dos pontos que representam a pele, o crânio e as meninges podem ser vistas na Figura 3.3(c). Finalmente a Figura 3.3(d) mostra todas as partes juntas. A P-LSP foi, portanto, capaz de apoiar a separação de estruturas internas e externas, mesmo aquelas que são muito similares ou próximas umas das outras.

¹disponível em <http://www9.informatik.uni-erlangen.de/External/vollib/>

Figura 3.3: Projeções P-LSP dos dados de uma cabeça. As projeções mostram boa separação entre estruturas diferentes (Paulovich et al., 2010).

3.4 Projeções em 3D

O algoritmo LSP tem sido aplicado com sucesso em vários tipos de dados multi-dimensionais como documentos e imagens (Paulovich et al., 2008; Eler et al., 2008). A exploração em 2D consiste em localizar, selecionar e examinar grupos visíveis. Apesar da efetividade, apresenta o problema de oclusão visual em quantidades grandes de dados ou em dados compactos. Além disso, grupos diferentes tendem a dividir a mesma região.

A proposta original da técnica da LSP resolve um sistema linear para cada coordenada do espaço projetado. A extensão para 3D é trivial, uma vez que se possa obter as coordenadas 3D dos pontos de controle. Escolhendo os pontos de controle e resolvendo o sistema é análogo ao problema para o caso 2D, só é necessário resolver mais um sistema para a terceira coordenada. No Anexo A é apresentado em detalhe o algoritmo da LSP e da P-LSP para o caso de 3D, assim como alguns resultados e evoluções.

Entre as funcionalidades de interação em 3D temos:

- *Representação como superfícies fechadas*: os pontos são agrupados e cada grupo é representado como uma superfície fechada; as superfícies podem ser selecionadas individualmente. Diferentes tipos de superfícies podem ser geradas (veja o Anexo A).
- Uso de diferentes projeções 2D a partir da projeção 3D: isto é feito para reproduzir os agrupamentos visuais em um espaço de manipulação mais fácil. Uma opção natural é fazer uma projeção ortogonal da visão atual 3D. Outras alternativas possíveis são aplicar PCA, MDS ou qualquer outra técnica de projeção para mapear de 3D para 2D.
- Coordenar as diferentes visões geradas, i.e as novas projeções geradas a partir da 3D.
- Combinação de diferentes grupos de superfícies para gerar conjuntos maiores cujos elementos podem ser armazenados para manipulação posterior.

Na Figura 3.4 mostramos o processo de geração de superfícies assim como os diferentes tipos de superfícies fechadas que são utilizadas em nosso sistema. O Anexo A fornece maiores detalhes destas funcionalidades.

Figura 3.4: Projeção, agrupamento e geração de superfícies no espaço visual da LSP 3D para um conjunto de documentos de quatro áreas do conhecimento.

3.5 Considerações Finais

Este capítulo descreveu as estratégias de visualização de volumes baseada em projeções, apresentando a conceituação das técnicas de projeção multi-dimensional, bem como uma discussão dos problemas apresentados quando a quantidade de elementos é muito grande. Também foram descritos dois algoritmos desenvolvidos em nosso grupo de investigação que buscam a preservação de vizinhança e a escalabilidade do algoritmo com conjuntos maiores de dados. Além disso foram apresentadas as projeções em 3D e alguns exemplos de como a utilização de superfícies fechadas sobre conjuntos de pontos permitem uma melhor interação com projeções 3D, as quais foram desenvolvidas neste trabalho de mestrado. Além da técnica de superfícies fechadas, são mencionadas outras estratégias para interagir em 3D. Finalmente foram mostrados alguns exemplos aos quais são aplicadas as técnicas de projeção em exploração de volumes e imagens.

O próximo capítulo é dedicado a apresentar como as técnicas LSP e P-LSP são usadas para a exploração de dados tensoriais da área médica apresentados como fibras neurais (*streamlines*) bem como a implementação envolvida neste trabalho.

Exploração de Fibras Neurais usando Projeções

4.1 Considerações Iniciais

Como foi mencionado no Capítulo 1, existe uma tendência a combinar as técnicas de visualização científica e de informação. Nosso trabalho segue essa linha, especificamente utilizando técnicas de projeção multi-dimensional (ver Capítulo 3) e rastreamento de fibras (ver Seção 2.3.4). O objetivo é utilizar as projeções como apoio para explorar os conjuntos de fibras que são visualizados em 3D.

A idéia principal é aproveitar a utilização da coordenação entre um espaço simplificado de características e a geometria exata em uma visão tradicional 3D, aliada à interação entre elas. Muitas abordagens tem sido usadas para definir uma representação em baixa dimensionalidade de conjuntos de fibras neurais de forma a preservar a vizinhança do espaço em alta dimensionalidade (Tsai et al., 2007). Estas novas formas de representação das fibras são usualmente utilizadas para seu agrupamento (O'Donnell e Westin, 2007). Alguns trabalhos (Chen et al., 2009; Jianu et al., 2009) demonstram a utilidade de representar as fibras em baixa dimensionalidade para melhorar a interatividade, visualização e exploração do conjunto original. As Figuras 4.1, 4.2 mostram as janelas principais das ferramentas desenvolvidas nesses trabalhos.

Um problema importante quando se trabalha com conjuntos de fibras é a quantidade de elementos. Os trabalhos mencionados anteriormente não resolvem este problema, pois os algoritmos utilizados para a redução de dimensionalidade apresentam custos computacionais muito altos pois que torna inviável a sua utilização em conjuntos maiores. Também devemos mencionar que as

Figura 4.1: Janela principal da ferramenta desenvolvida por [Chen et al. \(2009\)](#)

abordagens apresentadas têm como entrada uma matriz de distâncias, sendo quase impossível criar esta matriz para grandes quantidades de fibras devido às restrições de memória em computadores convencionais.

As vantagens de nossa abordagem são:

- A técnica P-LSP, utilizada neste trabalho para realizar projeções, pode lidar com conjuntos grandes de dados. Nesse trabalho só se apresentam resultados com 250k fibras, mas no trabalho ([Paulovich et al., 2010](#)) são apresentadas visualizações com conjuntos de dados ainda maiores.
- Como foi descrito na Seção 3.4 quando o número de elementos mostrados no plano aumenta, existe o problema de oclusão. Para lidar com este problema fizemos a extensão das técnicas LSP e P-LSP para um espaço 3D (ver Seção 3.4) assim como técnicas para interagir com projeções 3D.
- Uma nova abordagem de geração de superfícies sobre os conjuntos de fibras selecionados é apresentada. Este algoritmo é baseado no algoritmo apresentado em [Linsen et al. \(2008b\)](#), mas com uma pequena modificação para tratar grandes quantidades de pontos.

Neste capítulo apresentamos a metodologia utilizada para resolver os problemas mencionados acima. Na Seção 4.2 descrevemos cada um dos algoritmos utilizados nesta abordagem. A Seção 4.3 apresenta alguns resultados das nossas projeções e algumas comparações com trabalhos relacionados. Na Seção 4.4 apresentamos o algoritmo utilizado para visualizar conjuntos de fibras similares como um objeto só utilizando superfícies de densidade.

Figura 4.2: Janela principal da ferramenta desenvolvida por [Jianu et al. \(2009\)](#)

4.2 Metodologia

A Figura 4.3 representa as etapas principais do processo. Primeiro as fibras são convertidas do volume DTI para fibras. O segundo passo é a extração de características para converter as fibras neurais (ver Seção 2.3.4) em vetores. Depois cada um desses vetores é projetado para uma baixa dimensão 2D ou 3D usando a técnica P-LSP (ver Seção 3.2). Cada uma dessas etapas será descrita nas seções seguintes.

Figura 4.3: Passos principais do processo de transformação das fibras num espaço de dimensionalidade 2D ou 3D.

4.2.1 Conjunto de dados

Os dados utilizados neste trabalho são formados por conjuntos de fibras. A notação usada no resto do capítulo é a seguinte. Um ponto é representado por $p_i = \{x_i, y_i, z_i\}$, uma fibra é composta de vários pontos $f_j = \{p_i, i = 1, 2, \dots, m\}$, onde m é o numero de pontos na fibra. Um conjunto de fibras é denotado como $F = \{f_j, j = 1, 2, \dots, n\}$, onde n é o número de fibras.

As imagens mostradas neste trabalho foram geradas utilizando o conjunto de dados obtidos da *2009 Pittsburgh Brain Competition (PBC) Brain Connectivity Challenge* (<http://pbc.lrdc.pitt.edu/>). Para as comparações com outros métodos usamos os dados disponíveis dos trabalhos de [Chen et al. \(2009\)](#) e [Jianu et al. \(2009\)](#).

4.2.2 Conversão de DTI para Fibras Neurais

Este primeiro passo consiste em converter o volume de dados, onde cada voxel contém um tensor de difusão, para fibras neurais. O rastreamento das fibras está baseado na resolução da Equação 2.2. A resolução desta equação é descrita na Seção 2.3.4.

Alguns exemplos mostrados neste capítulo fazem esta etapa de conversão para fibras, mas na maior parte dos exemplos são utilizados diretamente o conjunto de fibras.

4.2.3 Extração de Características das Fibras Neurais

De cada fibra f_j é extraído um conjunto de características estatísticas e de curvatura. Os dois conjuntos são explicados nas sub-seções seguintes. Basicamente o Algoritmo 2 explica o processo de conversão das fibras F para uma matriz de vetores M , onde o α e β são fatores de peso para cada um dos conjuntos de características. Esses valores são definidos pelo usuário.

Algorithm 2 Extração de Características.

Require: F : Conjunto de fibras.

Ensure: M : Matriz de vetores de características.

```

1: procedure EXTRAÇÃO DE CARACTERÍSTICAS( $F$ )
2:   for all  $f_i \in F$  do
3:      $Caracteristicas_{stat} \leftarrow \text{ESTATÍSTICAS}(F)$ 
4:      $Caracteristicas_{curv} \leftarrow \text{CURVATURA}(F)$ 
5:      $M \leftarrow M \cup \{\alpha \cdot Caracteristicas_{stat}, \beta \cdot Caracteristicas_{curv}\}$ 
6:   end for
7: end procedure

```

A saída deste passo é uma matriz, onde cada linha representa uma fibra e o número de colunas representa o número de características extraídas. Em nossos experimentos nós escolhemos 40 características, das quais 10 são estatísticas, e 30 são de curvatura. Finalmente os valores de cada linha são normalizados entre 0 e 1.

(a) Características Estatísticas

As estatísticas usadas neste trabalho são: ponto inicial (p_1), ponto final (p_{last}), centro de massa (p_{cm}) e comprimento ($||f||$). O Algoritmo 3 apresenta de forma exata como cada uma das estatísticas é obtida.

Algorithm 3 Características estatísticas.

Require: F : Conjunto de fibras.

Ensure: M : Matriz de vetores de características.

```

1: procedure ESTATÍSTICAS( $F$ )
2:   for all  $f_i \in F$  do
3:      $Vec \leftarrow \emptyset$ 
4:      $Vec \leftarrow Vec \cup \{p_{1x}, p_{1y}, p_{1z}\}$ 
5:      $Vec \leftarrow Vec \cup \{p_{lastx}, p_{lasty}, p_{lastz}\}$ 
6:     for all  $p_j \in f_i$  do
7:        $p_{cm} \leftarrow p_{cm} + p_j$ 
8:     end for
9:      $p_{cm} \leftarrow \frac{p_{cm}}{||f||}$ 
10:     $Vec \leftarrow Vec \cup \{p_{cmx}, p_{cm_y}, p_{cm_z}\}$ 
11:     $Vec \leftarrow Vec \cup \{||f||\}$ 

12:     $M[i] \leftarrow Vec$ 
13:  end for
14: end procedure

```

(b) Características da Curvatura

A análise discreta de Fourier é muito útil em problemas de extração de características em imagens e análise de formas (Pazoti et al., 2005). Do espectro de Fourier são obtidos coeficientes espectrais, e para isto usamos a Transformada Discreta de Fourier (do inglês *Discrete Fourier Transform* - DFT). Em nossos exemplos o espaço multi-dimensional tem vetores compostos de 30 elementos, extraídos dos coeficientes espectrais de Fourier de cada uma das fibras em 3D. Os coeficientes espectrais são obtidos aplicando a DFT nos pontos que representam o traçado da fibra, i.e. a DFT é aplicada nos três sinais construídos com as coordenadas X, Y e Z. Só a magnitude (parte real) dos coeficientes é considerada para compor os vetores, e só usamos os valores relacionados a altas frequências, pois esses contém informação mais detalhada sobre a forma das fibras (Costa e Cesar, 2000; Bruno et al., 2008). O Algoritmo 4 mostra em detalhes cada um dos passos desta estratégia.

A saída desta etapa gera uma matriz onde cada linha representa uma fibra em um espaço de características. Esta será a entrada para a técnica de projeção em 2D ou 3D.

Algorithm 4 Características de Curvatura.**Require:** F : Conjunto de fibras.**Ensure:** M : Matriz de vetores de características.

```

1: procedure CURVATURA( $F$ )
2:   for all  $f_i \in F$  do
3:      $x = \{P_{j_x} : p_j \in f_i\}$ 
4:      $y = \{P_{j_y} : p_j \in f_i\}$ 
5:      $z = \{P_{j_z} : p_j \in f_i\}$ 
6:      $X \leftarrow \text{real}(FFT(x))$ 
7:      $Y \leftarrow \text{real}(FFT(y))$ 
8:      $Z \leftarrow \text{real}(FFT(z))$ 
9:      $n \leftarrow ||f_i||$  ▷ Numero de pontos.
10:     $Vec \leftarrow X[1, \dots, k]$  ▷ k-valores de altas frequências
11:     $Vec \leftarrow Vec \cup Y[1, \dots, k]$  ▷ k-valores de altas frequências
12:     $Vec \leftarrow Vec \cup Z[1, \dots, k]$  ▷ k-valores de altas frequências
13:     $M[i] \leftarrow Vec$ 
14:   end for
15: end procedure

```

4.2.4 Projeção das Fibras Neurais

O conjunto de vetores de características em alta dimensionalidade que foi extraído da etapa anterior serve de entrada para utilizar a técnica de projeção. Existem diferentes técnicas de projeção multi-dimensional, como foi explicado no Capítulo 3, mas devemos escolher uma técnica que permita trabalhar com um conjunto de dados grande e com um custo computacional baixo. Para fazer esta tarefa utilizamos a técnica P-LSP, que foi explicada na Seção 3.2.2, por que permite tratar rapidamente conjuntos grandes de dados, e por que tem uma boa precisão, preservando na projeção as vizinhanças do conjunto no espaço original. Uma descrição mas detalhada das características desta técnica pode ser lida em [Paulovich et al. \(2010\)](#).

Nos exemplos seguintes também utilizamos as versões 3D das técnicas LSP e P-LSP, que são parte desde trabalho (ver Seção 3.4).

4.3 Resultados das Projeções de Fibras Neurais

Nesta seção mostramos algumas imagens geradas com nosso protótipo, além de comparações com outros trabalhos diretamente relacionados.

A Figura 4.4 apresenta no lado esquerdo o conjunto de 250k fibras visualizadas com tubos em 3D, e no lado direito, a representação das 250k fibras em um espaço 2D usando a metodologia explicada na seção anterior. Este primeiro resultado mostra as dificuldades de uma exploração 3D das fibras devido à grande quantidade de fibras, apresentando o problema de *cluttering*.

Figura 4.4: Duas visões de 250k fibras.

Na Figura 4.5 mostramos um exemplo das fibras renderizadas como linhas. Estas fibras são projetadas num espaço 3D e são utilizadas superfícies para a interação com a projeção 3D. Finalmente, a partir da projeção 3D é aplicada uma projeção ortogonal e criada uma projeção 2D para a interação com as 3 visões. Como é mostrado na Seção 3.4 os pontos em 3D tem uma melhor precisão e permitem descobrir agrupamentos que são sobrepostos nas projeções 2D.

4.3.1 Comparações da Qualidade das Projeções

As comparações foram feitas com os trabalhos de [Chen et al. \(2009\)](#); [Jianu et al. \(2009\)](#). Para a geração destas imagens foram utilizados os conjuntos de dados apresentados naqueles trabalhos.

Para esta primeira comparação utilizamos o conjunto de dados de [Jianu et al. \(2009\)](#). Na Figura 4.6 exibimos 3 visões: a primeira apresenta o conjunto de dados original, que é visualizado de forma tradicional como linhas, e embaixo 2 projeções. A primeira é o resultado da técnica de redução de dimensionalidade do trabalho ([Jianu et al., 2009](#)) e a segunda é a projeção do mesmo conjunto de dados usando a técnica P-LSP. As cores usadas nas três visões estão coordenadas, i.e. primitivas com a mesma cor representam os mesmos elementos. No resultado de [Jianu et al. \(2009\)](#) os pontos têm um posicionamento com forma bem definida e com certa coerência à representação 3D. Nosso resultado mostra um acomodamento menos uniforme, mas isso não afeta a qualidade da projeção. Por exemplo, os grupo de primitivas de cor amarela apresentam um melhor posicionamento em nossa projeção por que mostra grupos bem definidos, o que não acontece com a outra projeção onde é difícil saber onde se encontram os grupos. As primitivas vermelhas em ambas projeções mostra um grupo bem definido, separado dos outros elementos.

(a) Visualização de 20k fibras usando linhas

(b) Projeção em 3D e criação de superfícies para a interação.

(c) Projeção 2D criada a partir da projeção ortogonal da projeção 3D.

Figura 4.5: Três visões de um conjunto de 20k fibras.

Para a segunda comparação usamos o conjunto de dados do trabalho [Chen et al. \(2009\)](#) (ver Figura 4.7). Neste caso nossa projeção apresentou bom resultados. Por exemplo, no grupo azul, em nossa projeção pode-se ver um grupo bem formado, o que não acontece com a outra projeção na

qual os pontos azuis estão espalhados. O efeito pode ser comprovado na visualização 3D onde as fibras de cor azul tem forma e posicionamento semelhante. No caso dos elementos vermelhos, na visão 3D vemos que eles não tem uma forma muito semelhante, mas seu posicionamento espacial é quase o mesmo. É por isso que em nossa projeção elas se encontram agrupadas, o que não acontece com a outra projeção.

As comparações feitas mostram que a qualidade dos resultados da nossa abordagem é semelhante e, em alguns casos melhor, por que mostram grupos melhor definidos. A principal vantagem da nossa estratégia é a possibilidade de utilização para grandes quantidades de dados. Na próxima seção mostramos algumas comparações em tempo para a redução de dimensionalidade.

4.3.2 Comparações em Tempo

Um problema importante a resolver com este trabalho era o tratamento de grandes quantidades de dados. Por isso foi desenvolvida a técnica P-LSP, que pode lidar rapidamente com conjuntos grandes. A tabela 4.1 mostra uma comparação em tempo com o trabalho de [Chen et al. \(2009\)](#). A diferença entre ambas técnicas é grande. No caso de 250k fibras, só foi possível fazer uma redução de dimensionalidade usando a técnica P-LSP e com um tempo 69s.

Numero de fibras	P-LSP (CPU)	Chen et al. (2009) GPU
250000	69,0s	...
4096	1,598s	1125,0s
2048	0,92s	256,0s
1024	0,875s	67,0s
512	0,484s	1,2s

Tabela 4.1: Comparação em tempo para a redução de dimensionalidade.

Finalmente, podemos dizer que nossa estratégia satisfaz nossos objetivos de resolver o problema de lidar com conjuntos grandes de fibras, em tempo baixos, e com uma qualidade das projeções semelhante ou melhor que as outras abordagens. Na próxima seção descrevemos nosso algoritmo para a visualização dos conjuntos de fibras como um objeto através de superfícies de densidade.

4.4 Visualização das Fibras como Superfícies

Recentemente têm sido apresentados muitos algoritmos para a criação de superfícies sobre conjuntos de fibras. Algumas referências são [Merhof et al. \(2009\)](#); [Enders et al. \(2005\)](#); [Chen et al. \(2008\)](#); [Klein et al. \(2007\)](#). Neste trabalho não pretendemos desenvolver uma técnica que tenha resultados muito precisos sobre a forma espacial dos conjuntos de fibras, ou superfícies de melhor qualidade. Nosso principal objetivo é trabalhar com conjuntos grandes e que permitam gerar estas superfícies em tempo real.

Para isto apresentamos uma técnica que não é nova, mas é uma variação do trabalho de [Linsen et al. \(2008b\)](#) para criar superfícies mais rápidas sem sacrificar muito a qualidade da superfície. A técnica foi originalmente utilizada para criar superfícies sobre conjuntos de pontos esparsos. Em nosso caso cada uma das fibras é representada como um conjunto de pontos. Este conjunto de pontos é a entrada no algoritmo de criação de superfícies. Este método consiste em dois passos nos quais o primeiro é transformar o espaço esperso num espaço contínuo e discreto. Este processo de voxelização calcula a dimensão do voxel como o comprimento da maior aresta da árvore de cobertura mínima gerada a partir dos pontos. Este cálculo tem um custo computacional muito caro especialmente neste contexto onde trabalhamos com quantidades de fibras grandes e cada uma das fibras é composta por centenas de pontos. Por isso nós calculamos o comprimento do voxel (Δ) como $\Delta \leftarrow \frac{p_{max} - p_{min}}{dim}$, onde p_{max} e p_{min} são calculados com o *bounding box* dos pontos e dim é definido pelo usuário. Em nossos experimentos usamos $dim = 100$. O Algoritmo 5 descreve cada um dos passos da nossa proposta.

Algorithm 5 Algoritmo para a criação de Superfícies.

Require: F : fibras selecionadas.

Ensure: $Surf$: superfície (malha de triângulos).

```

1: procedure CRIARSUPERFICIE( $F$ )
2:    $P = \{Points \in f_i, \forall f_i \in F\}$ 
3:    $Grid \leftarrow$  CRIARGRID( $P$ ).
4:    $Surf \leftarrow$  MARCHINGCUBES( $Grid, c$ ). ▷ Iso-value.
5: end procedure
```

Require: P : Pontos.

Ensure: $Grid$

```

6: procedure CRIARGRID( $Points$ )
7:    $dim \leftarrow 100$ 
8:    $[p_{min}, p_{max}] =$  BOUNDINGBOX( $Points$ )
9:    $\Delta \leftarrow \frac{p_{max} - p_{min}}{dim}$ 
10:   $Grid \leftarrow$  Tamanho do volume( $dim, dim, dim$ )
11:  for all  $p_i \in P$  do
12:     $C \leftarrow \frac{p_i - p_{min}}{\Delta}$  ▷ Centro da esfera.
13:     $R \leftarrow 5$  ▷ Raio da esfera.
14:    for all  $v_j \in$  SPHERE( $C, R$ ) do
15:       $d \leftarrow$  DIST( $v_j, C$ ) ▷ Distância Euclidiana.
16:       $v_j \leftarrow v_j + (\frac{1-d}{R})^2$ 
17:    end for
18:  end for
19: end procedure
```

A Figura 4.8 mostra a projeção de um conjunto de fibras classificadas (20k fibras) e as superfícies geradas em cada um dos grupos.

4.5 Sistema de Visualização de Fibras

Como resultado também deste trabalho, um sistema foi desenvolvido que compreende todas as características aqui descritas. O sistema é incorporado na forma de módulos adicionados à arquitetura VisPipeline, de visualização de propósito geral, desenvolvida no contexto do grupo de Visualização, Imagens e Computação Gráfica do ICMC-USP. O VisPipeline, por sua vez, incorpora as características de um sistema de visualização baseado em projeções previamente desenvolvido pelo grupo, chamado Pex (*Projection Explorer* (Paulovich et al., 2007)).

Neste trabalho foram desenvolvidos os seguintes módulos:

- Conversão de dados DTI para informações de fibras (*Convert DTI 2 Fibers*).
- Extração de características de fibras (*Feature Extractor*).
- Projeção LSP 3D (*LSP-3D Projection*).
- Projeção P-LSP 3D (*P-LSP-3D Projection*).
- Geração de superfícies envolventes (*Fibers View*).

O formato do arquivo de fibras (TRK) é o mesmo utilizado pelo software TrackVis, a descrição deste formato é definido em <http://www.trackvis.org/docs/?subsect=fileformat>. Um video do sistema é disponibilizado em <http://infoserver.lcad.icmc.usp.br/infovis2/Tools>.

4.6 Considerações Finais

Neste capítulo foi apresentada a abordagem para a visualização de fibras neurais com apoio das projeções multi-dimensionais para a exploração do conjunto de fibras. Primeiro as fibras são transformadas em vetores, os quais contém características sobre a sua curvatura, assim como informação estatística. Isto é feito para evitar a criação de matrizes de distâncias, as quais não são factíveis quando o conjunto de elementos é grande. Depois é utilizada uma técnica de projeção multi-dimensional que reduz a dimensionalidade de grandes conjuntos de dados rapidamente, e com uma boa preservação das vizinhanças do conjunto em alta dimensionalidade. Estas características em conjunto conseguem atingir os objetivos planejados, mostrados nas comparações apresentadas anteriormente.

Adicionalmente mostramos uma variação de um algoritmo de criação de superfícies sobre pontos esparsos para criar superfícies de nível em conjuntos de fibras que são selecionadas por meio da coordenação que existe entre as diferentes visões.

Finalmente esta nova abordagem não está limitada a projeções 2D. Conforme já mencionado na Seção 3.4. Quando trabalhamos com projeções 3D a precisão dos resultados é melhor, e naquela

mesma seção explicamos como interagir em forma conjunta entre 2D e 3D. Ao usar variações e melhoras, a estratégia permite rápida e flexível exploração de conjuntos de fibras. Os Anexos a este trabalho mostram com mais detalhes os resultados das principais contribuições.

(a) Visualização com linhas

(b) Projeção 2D obtida do trabalho [Jianu et al. \(2009\)](#)

(c) Projeção usando a P-LSP

Figura 4.6: As fibras usadas para gerar estas imagens foram obtidas da pagina web do projeto ([Jianu et al., 2009](#)).

(a) Visualização com linhas

(b) Projeção 2D obtida do trabalho [Chen et al. \(2009\)](#)

(c) Projeção usando a P-LSP

Figura 4.7: Fibras cerebrais obtidas com dados obtidos de <http://www.cad.zju.edu.cn/home/chenwei/interface>.

(a) Projeção 2D usando a P-LSP

(b) Visualização das superfícies geradas usando as fibras que tem a mesma cor

Figura 4.8: 20k fibras, a cor das instâncias significam, a classe à que pertencem. Dados obtidos de <http://pbc.lrdc.pitt.edu/>

Conclusões e Trabalhos Futuros

A visualização de imagens e dados gerados por tensores de difusão é uma área que tem alcançado muita importância nos últimos anos. Existem diferentes métodos de visualização deste tipo de dados, e o método mais promissor é o rastreamento de fibras. A seleção de conjunto de fibras é um problema equivalente à definição de funções de transferência no *rendering* volumétrico direto, na qual não existe uma estratégia ótima de realizar a tarefa.

Neste trabalho propusemos uma nova abordagem de visualização e interação com fibras neurais usando projeções multi-dimensionais. Para atingir este objetivo foram apresentadas estratégias para a conversão das fibras em vetores, uma nova técnica de projeção, a P-LSP, extensão das projeções em 3D, e finalmente outra abordagem para visualizar conjuntos de fibras selecionados como um só objeto. O conjunto dessas técnicas consegue realizar o objetivo proposto.

Como parte deste trabalho a técnica LSP foi estendida para a P-LSP, a qual permite lidar com maiores conjuntos de dados. Neste trabalho mostramos os tempos conseguidos para fazer a projeção de até 250k fibras, os quais são muito baixos em comparação às técnicas que utilizam outros trabalhos que também fazem projeção de fibras. A qualidade das projeções geradas usando a P-LSP foi comparável com outras técnicas, e em alguns casos melhor. Estas conclusões foram inferidas fazendo uma análise visual e coordenada entre as projeções e as fibras em sua representação natural.

Outro trabalho que foi resultado desta dissertação é a extensão das técnicas LSP e P-LSP de 2D para 3D, uma extensão que permite ter uma maior precisão nos dados projetados. Ela é detalhada no Anexo A. O desafio nesta extensão foi criar novas técnicas de interação para trabalhar com os pontos em 3D. Uma abordagem foi criar diferentes tipos de superfícies sobre os pontos que foram classificados previamente para selecionar conjuntos de pontos em 3D. Uma outra abordagem

foi gerar, a partir dos pontos em 3D, projeções em 2D utilizando diferentes técnicas (ex: PCA, Fastmap, LSP, etc.) e manter as janelas coordenadas de forma que possamos interagir com as projeções 3D usando todas as técnicas desenvolvidas em 2D.

Um outro resultado deste trabalho foi a criação de superfícies sobre as fibras para representá-las como um só objeto. Esta estratégia tem sido muito usada nos últimos anos, e neste trabalho apresentamos uma variação da abordagem apresentada por [Linsen et al. \(2008b\)](#). Esta pequena variação permite lidar com quantidades maiores de pontos em tempos menores. A qualidade das superfícies geradas foi comparável às geradas usando o algoritmo original para quantidades menores de dados. Na nossa abordagem também foi possível, trabalhar com um número muito maior de fibras do que com outras abordagens.

Não foi possível comparar os resultados de ambos algoritmos com quantidades maiores devido a que o algoritmo original não conseguia trabalhar nestas condições.

Também é importante mencionar que as estratégias apresentadas neste trabalho tem características paralelizáveis, que permitiriam a implementação dos algoritmos em placa gráfica, e assim poder acelerar ainda mais o tempo de projeção assim como a criação das superfícies.

Um outro trabalho futuro seria a utilização de volumes de outras naturezas (e.g. *functional MRI*, tomografias, etc.) e utilizar as projeções para analisar de forma coordenada os diferentes tipos de dados.

Todas as técnicas apresentadas neste trabalho se encontram implementadas na aplicação Vis-Pipeline a qual foi desenvolvida pelo grupo de visualização do ICMC/USP, esta encontra-se disponível em <http://infoserver.lcad.icmc.usp.br/infovis2>.

Referências Bibliográficas

- A.BRANCO, V. M. *Visualização como suporte à exploração de uma base de dados pluviométricos*. Dissertação de Mestrado, ICMC-USP, 2003. [2](#)
- ALENCAR, A. B.; PAULOVICH, F. V.; MINGHIM, R.; FILHO, M. G. A.; OLIVEIRA, M. C. F. Similarity-based visualization of time series collections: an application to analysis of stream-flows). In: *12th International Conference on Information Visualisation - IV*, London-UK: IEEE CS Press, 2008, p. 280–286. [9](#)
- BASSER, P. J.; PIERPAOLI, C. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI. *Journal of magnetic resonance. Series B*, v. 111, p. 209–219, 1996. [14](#)
- BERENSCHOT, G. *Visualization of diffusion tensor imaging*. Dissertação de Mestrado, Eindhoven University of Technology, 2003. [vii](#), [12](#), [15](#)
- BLAAS, J.; BOTHA, C.; POST, F. Extensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. *IEEE Transactions on Visualization and Computer Graphics, Special Issue IEEE Visualization 2008 and IEEE Infovis 2008*, v. 14, n. 6, p. 1229–1236, 2008. [vii](#), [5](#), [8](#)
- BLAAS, J.; BOTHA, C. P.; POST, F. H. Interactive visualization of multi-field medical data using linked physical and feature-space views. In: MÜSETH, K.; YNNERMAN, A.; MÖLLER, T., eds. *Eurovis 2007, Proceedings of the joint Eurographics / IEEE VGTC Symposium on Visualization*, 2007, p. 123–130. [vii](#), [5](#), [7](#)
- BRUN, A.; KNUTSSON, H.; PARK, H. J.; SHENTON, M. E.; WESTIN, C.-F. Clustering fiber tracts using normalized cuts. In: *Seventh International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'04)*, Lecture Notes in Computer Science, Rennes - Saint Malo, France, 2004, p. 368–375 (*Lecture Notes in Computer Science*,). [18](#), [20](#)

- BRUN, A.; PARK, H.-J.; KNUTSSON, H.; WESTIN, C.-F. Coloring of DT-MRI fiber traces using laplacian eigenmaps. In: DIAZ, R. M.; ARENCIBIA, A. Q., eds. *Computer Aided Systems Theory (EUROCAST'03), Lecture Notes in Computer Science 2809*, Las Palmas de Gran Canaria, Spain: Springer Verlag, 2003, p. 564–572. [vii](#), [17](#), [18](#), [19](#)
- BRUNO, O.; NONATO, L.; PAZOTI, M.; BATISTA, J. Topological multi-contour decomposition for image analysis and image retrieval. *Pattern Recognition Letters*, v. 29, p. 1675–1683, 2008. [34](#)
- CARD, S. K.; MACKINLAY, J. D.; SHNEIDERMAN, B. *Readings in information visualization: using vision to think*, cáp. Infosphere, workspace, tools, objects San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., p. 463–464, 1999. [2](#)
- CHALMERS, M. A linear iteration time layout algorithm for visualising high-dimensional data. In: *Proceedings of the IEEE Visualization 1996 (VIS'96)*, Los Alamitos, CA, USA: IEEE Computer Society Press, 1996, p. 127–ff. [22](#)
- CHEN, W.; DING, Z.; ZHANG, S.; MACKAY-BRANDT, A.; CORREIA, S.; QU, H.; CROW, J.; TATE, D.; YAN, Z.; PENG, Q. A Novel Interface for Interactive Exploration of DTI Fibers. *IEEE Transactions on Visualization and Computer Graphics*, v. 15, n. 6, p. 1433–1440, 2009. Disponível em: <http://doi.ieeecomputersociety.org/10.1109/TVCG.2009.112> [viii](#), [30](#), [31](#), [33](#), [36](#), [37](#), [38](#), [43](#)
- CHEN, W.; ZHANG, S.; CORREIA, S.; EBERT, D. S. Abstractive Representation and Exploration of Hierarchically Clustered Diffusion Tensor Fiber Tracts. *Computer Graphics Forum*, v. 27, n. 3, p. 1071–1078, 2008. Disponível em: <http://blackwell-synergy.com/doi/abs/10.1111/j.1467-8659.2008.01244.x> [38](#)
- CHERNOFF, H. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, v. 68, n. 342, p. 361–368, 1973. [4](#)
- COROUGE, I.; GERIG, G.; GOUTTARD, S. Towards a shape model of white matter fiber bundles using diffusion tensor MRI. In: *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, 2004, p. 344–347. [18](#), [19](#)
- COSTA, L. D. F. D.; CESAR, JR., R. M. *Shape analysis and classification: Theory and practice*. Boca Raton, FL, USA: CRC Press, Inc., 2000. [34](#)
- COX, T. F.; COX, M. A. A. *Multidimensional scaling*. Second ed. Chapman & Hall/CRC, 2000. [21](#), [23](#)
- CUADROS, A. M.; PAULOVICH, F. V.; MINGHIM, R.; TELLES, G. P. Point placement by phylogenetic trees and its application to visual analysis of document collections. In: *Proceedings*

- of IEEE Symposium on Visual Analytics Science and Technology - IEEE VAST*, IEEE CS Press, 2007, p. 99–106. [9](#)
- DOLEISCH, H.; GASSER, M.; HAUSER, H. Interactive feature specification for focus+context visualization of complex simulation data. In: *VISSYM '03: Proceedings of the symposium on Data visualisation 2003*, Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2003, p. 239–248. [vii](#), [5](#), [6](#)
- DOUEK, P.; TURNER, R.; PEKAR, J.; PATRONAS, N.; LE BIHAN, D. MR color mapping of myelin fiber orientation. *J Comput Assist Tomogr*, v. 15, p. 923–929, 1991. [13](#)
- ELER, D. M.; NAKAZAKI, M. Y.; PAULOVICH, F. V.; SANTOS, D. P.; OLIVEIRA, M. C. F.; NETO, J. E. B.; MINGHIM, R. Multidimensional visualization to support analysis of image collections (to appear). In: *Proceedings of the XXI Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI*, Campo Grande, Brazil: IEEE CS Press, 2008. [9](#), [27](#)
- ENDERS, F.; SAUBER, N.; MERHOF, D.; HASTREITER, P.; NIMSKY, C.; STAMMINGER, M. Visualization of white matter tracts with wrapped streamlines. In: *Proceedings of IEEE Visualization*, 2005, p. 51–58.
Disponível em: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Visualization+of+White+Matter+Tracts+with+Wrapped+Streamlines#0> [38](#)
- 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.
Disponível em: citeseer.ist.psu.edu/fruchterman91graph.html [22](#)
- 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. In: *VISUALIZATION '00: Proceedings of the 11th IEEE Visualization 2000 Conference (VIS 2000)*, Washington, DC, USA: IEEE Computer Society, 2000. [vii](#), [5](#), [6](#)
- GRINSTEIN, G.; TRUTSCHL, M.; CVEK, U. High-dimensional visualizations. In: *7th Data Mining Conference-KDD 2001*, San Francisco, California, 2001, p. 7–19. [4](#), [5](#)
- HANSEN, C. D.; JOHNSON, C. R., eds. *The visualization handbook*. Orlando, FL, USA: Academic Press, Inc., 2005. [1](#), [3](#)
- HOFFMAN, P.; GRINSTEIN, G.; PINKNEY, D. Dimensional anchors: a graphic primitive for multidimensional multivariate information visualizations. In: *NPIVM '99: Proceedings of the 1999 workshop on new paradigms in information visualization and manipulation in conjunction with the eighth ACM international conference on Information and knowledge management*, New York, NY, USA: ACM, 1999, p. 9–16. [2](#)

- INSELBERG, A. The plane with parallel coordinates. *The Visual Computer*, v. V1, n. 4, p. 69–91, 1985.
Disponível em: <http://dx.doi.org/10.1007/BF01898350> 2
- JAIN, A.; FARROKHNI, F. Unsupervised texture segmentation using gabor filters. In: *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*, 1990, p. 14–19. 25
- JIANU, R.; DEMIRALP, C.; LAIDLAW, D. H. Exploring 3D DTI fiber tracts with linked 2D representations. *IEEE transactions on visualization and computer graphics*, v. 15, n. 6, p. 1449–56, 2009.
Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/19834220> viii, 30, 32, 33, 36, 42
- JOURDAN, F.; MELANÇON, G. Multiscale hybrid mds. In: *IV '04: Proceedings of the Information Visualisation, Eighth International Conference on (IV'04)*, Washington, DC, USA: IEEE Computer Society, 2004, p. 388–393. 22
- KINDLMANN, G. Superquadric tensor glyphs. In: *Proceeding of The Joint Eurographics - IEEE TCVG Symposium on Visualization*, 2004, p. 147–154. 16
- KINDLMANN, G.; WEINSTEIN, D.; HART, D. Strategies for direct volume rendering of diffusion tensor fields. *IEEE Transactions on Visualization and Computer Graphics*, v. 6, n. 2, p. 124–138, 2000. vii, 15
- KLEIN, J.; BITTIHN, P.; LEDOCHOWITSCH, P.; HAHN, H.; KONRAD, O.; REXILIUS, J.; PEITGEN, H. Grid-based spectral fiber clustering. In: *Proceedings of SPIE*, 2007, p. 65091E.
Disponível em: <http://link.aip.org/link/?PSISDG/6509/65091E/1> 38
- LE BIHAN, D. Looking into the functional architecture of the brain with diffusion mri. *Nat Rev Neurosci*, v. 4, n. 6, p. 469–480, 2003.
Disponível em: <http://dx.doi.org/10.1038/nrn1119> 11
- 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, Special Issue IEEE Visualization 2008 and IEEE Infovis 2008*, v. 14, n. 6, p. 1229–1236, 2008a. vii, 7, 8
- LINSEN, L.; VAN LONG, T.; 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. 1483–90, 2008b.
Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/18989000> 31, 39, 46

- MACQUEEN, J. B. Some methods for classification and analysis of multivariate observations. In: CAM, L. M. L.; NEYMAN, J., eds. *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 1967, p. 281–297. 25
- MANJUNATH, B.; MA, W. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 18, n. 8, p. 837–842, 1996. 25
- MCCORMICK, B. H. Visualization in scientific computing. *IEEE Comput. Graph. Appl.*, v. 7, n. 10, p. 69, 1987. 1
- MERHOF, D.; MEISTER, M.; BINGOL, E.; NIMSKY, C.; GREINER, G. Isosurface-Based Generation of Hulls Encompassing Neuronal Pathways. *Stereotact Funct Neurosurg*, v. 87, n. 1, p. 50–60, 2009.
Disponível em: <http://content.karger.com/produktedb/produkte.asp?typ=pdf&file=000195720> 38
- MOBERTS, B. *Hierarchical visualization using fiber clustering*. Dissertação de Mestrado, Technische Universiteit Eindhoven, 2005. vii, 12, 14, 16
- MORI, S.; ZIJL, P. C. M. Fiber tracking: principles and strategies - a technical review. *NMR in Biomedicine*, 2002. 16
- 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. 22
- MORRISON, A.; ROSS, G.; CHALMERS, M. A hybrid layout algorithm for sub-quadratic multidimensional scaling. In: *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, Washington, DC, USA: IEEE Computer Society, 2002, p. 152. 22
- O'DONNELL, L.; WESTIN, C. Automatic tractography segmentation using a high-dimensional white matter atlas. *IEEE Transactions on Medical Imaging*, v. 26, n. 11, p. 1562–1575, 2007. 30
- OLIVEIRA, M. C. F.; MINGHIM, R. Uma introdução à visualização computacional. *Jornadas de Atualização em Informática, XVII Congresso da SBC*, v. 1, p. 85–131, 1997. 2, 3
- PAIVA, A.; SEIXA, R.; GATTAS. *Introdução à visualização volumétrica*. Relatório Técnico, http://www.tecgraf.puc-rio.br/publications/artigo_1999_introducao_visualizacao_volumetrica.pdf; acessada no 1 de Agosto 2009, 1999. 2
- PAULOVICH, F. V.; ELER, D. M.; POCO, J.; NONATO, L. G.; BOTHA, 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), 2010. vii, 24, 25, 26, 27, 31, 35

- PAULOVICH, F. V.; NONATO, L. G.; MINGHIM, R.; LEVKOWITZ, H. Visual mapping of text collections through a fast high precision projection technique. In: *Proceedings of the 10th International Conference on Information Visualisation - IV*, Washington, DC, USA: IEEE Computer Society Press, 2006, p. 282–290. 9
- 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, 9, 22, 24, 27
- PAULOVICH, F. V.; OLIVEIRA, M. C. F.; MINGHIM, R. The projection explorer: A flexible tool for projection-based multidimensional visualization. In: *Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI*, Belo Horizonte, Brazil: IEEE CS Press, 2007, p. 27–36. 40
- PAZOTI, M. A.; GARCIA, R. E.; DALTON, J.; PESSOA, C.; BRUNO, O. M. Comparison of shape analysis methods for guiardia citricarpa ascospore characterization. *Electronic Journal of Biotechnology*, v. 8, n. 3, p. 265–275, 2005. 34
- PIERPAOLI, C.; BASSER, P. J. Toward a quantitative assessment of diffusion anisotropy. *Magn Reson Med*, v. 36, n. 6, p. 893–906, 1996.
Disponível em: <http://view.ncbi.nlm.nih.gov/pubmed/8946355> 13, 14
- RAO, R.; CARD, S. K. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In: *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA: ACM, 1994, p. 318–322. 2, 4
- SAMMON, J. W. A nonlinear mapping for data structure analysis. In: *IEEE Transactions on Computers*, 1964, p. 401–409. 21
- 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. 24
- SILVA, P. M. *Visualização científica de um fenômeno hidrológico na amazônia com variação sazonal*. Dissertação de Mestrado, UNIVERSIDADE UNIVERSIDADE FEDERAL DO RIO DE JANEIRO, 2006. 2
- SORKINE, O.; COHEN-OR, D. Least-squares meshes. In: *Proceedings of Shape Modeling International*, IEEE Computer Society Press, 2004, p. 191–199. 23
- STEINBACH, M.; KARYPIS, G.; KUMAR, V. A comparison of document clustering techniques. In: *Workshop on Text Mining, 6th ACM SIGKDD International Conference on Data Mining (KDD'00)*, Boston, Massachusetts, USA: ACM, 2000, p. 109–110. 24

- TEJADA, E.; MINGHIM, R.; NONATO, L. G. On improved projection techniques to support visual exploration of multidimensional data sets. *Information Visualization*, v. 2, n. 4, p. 218–231, 2003. 21
- TSAI, A.; WESTIN, C.; HERO, A.; WILLSKY, A. Fiber tract clustering on manifolds with dual rooted-graphs. In: *IEEE Conf. Computer Vision and Pattern Recognition*, Citeseer, 2007. 30
- VILANOVA, A.; ZHANG, S.; KINDLMANN, G.; LAIDLAW, D. H. An introduction to visualization of diffusion tensor imaging and its applications. In: *Visualization and Image Processing of Tensor Fields*, Springer-Verlag, [http://www.cs.brown.edu/research/vis/docs/pdf/Vilanova-2005-IVD.pdf\(pdf\)](http://www.cs.brown.edu/research/vis/docs/pdf/Vilanova-2005-IVD.pdf(pdf)) [http://www.cs.brown.edu/research/vis/results/bibtex/Vilanova-2005-IVD.bib\(bibtex: Vilanova-2005-IVD\).](http://www.cs.brown.edu/research/vis/results/bibtex/Vilanova-2005-IVD.bib(bibtex: Vilanova-2005-IVD).), 2005. vii, ix, 14, 15, 17
- WEINSTEIN, J. N.; MYERS, T. G.; O’CONNOR, P. M.; FRIEND, S. H.; FORNACE, A. J.; KOHN, K. W.; FOJO, T.; BATES, S. E.; RUBINSTEIN, L. V.; ANDERSON, N. L.; BUOLAMWINI, J. K.; OSDOL, W. W.; MONKS, A. P.; SCUDIERO, D. A.; SAUSVILLE, E. A.; ZAHAREVITZ, D. W.; BUNOW, B.; VISWANADHAN, V. N.; JOHNSON, G. S.; WITTES, R. E.; PAULL, K. D. An information-intensive approach to the molecular pharmacology of cancer. *Science*, v. 275, p. 343–349, 1997. 4
- WESTIN, C.-F.; MAIER, S. E.; KHIDHIR, B.; EVERETT, P.; JOLESZ, F. A.; KIKINIS, R. Image processing for diffusion tensor magnetic resonance imaging. In: *Medical Image Computing and Computer-Assisted Intervention*, Lecture Notes in Computer Science, 1999, p. 441–452 (*Lecture Notes in Computer Science*,). 3
- WESTIN, C.-F.; PELED, S.; GUDBJARTSSON, H.; KIKINIS, R.; JOLESZ, F. A. Geometrical diffusion measures for MRI from tensor basis analysis. In: *ISMRM ’97*, Vancouver Canada, 1997, p. 1742. 14
- WITTEN, I. H.; FRANK, E. Data mining: practical machine learning tools and techniques with Java implementations. *ACM SIGMOD Record*, v. 31, n. 1, p. 76–77, 2002. 2
- ZHANG, S.; LAIDLAW, D. H. *Hierarchical clustering of streamtubes*. Relatório Técnico CS-02-18, Brown University Computer Science Department, 2002. 18, 19

APÊNDICE

A

Apêndice A

A framework for exploring multidimensional data with 3D projections

Jorge Poco, Fernando Paulovich, Maria Cristina de Oliveira,
Tran Van Long, Paul Rosenthal, Lars Linsen and Rosane Minghim

Abstract—One recurrent way to explore high dimensional spaces is to map them, via projections, to visual (2D or 3D) spaces. Multidimensional projection techniques generate views capable of discriminating data objects based on their similarity relations. Highly correlated elements tend to be placed close to each other, affording intuitive interactive exploration. Exploration functions typically include locating and selecting groups and gradually drilling down to individual objects. Projections on 2D spaces limit grouping and separation capabilities as compared to 3D spaces, which, in turn, make it more difficult to interact with scattered groups of points. In this paper we adapt an existing high-precision projection technique to create 3D data representations, and demonstrate that the third dimension significantly improves projection precision. We then introduce effective ways to explore 3D projections employing enclosing surfaces of data groups and linked views to 2D projections. We demonstrate the applicability of the approach to exploration of multidimensional abstract data and multi-field spatial data from scientific simulations.

Keywords—multidimensional visualization; scientific visualization;

I. INTRODUCTION

Document and image collections, time series, or multiple scalar fields related to a single phenomenon are just a few examples of high-dimensional data. Creating visual representations capable of providing insight into the global behavior of such data is challenging. Typical questions are: Are there well-defined groups of similar objects? How are different groups related? What about instances within a group? Which data features determine the groupings? Seeking answers to those questions requires intuitive visual representations and effective user interaction. Multidimensional projection techniques offer an unifying framework in this scenario. They map data to a low-dimensional space suitable for user interaction, i.e., a 2D or 3D visual space. While 2D maps afford easy interaction, 3D projections decrease information loss allowing better group discrimination. However, 3D projections are more difficult to interact with in everyday applications.

In this paper, we introduce an interaction framework for 3D projections obtained from multidimensional data sources of various types. We depart from two distinct recent contributions. The first one is the *Least Square Projection* technique (LSP) [1]. We extend 2D LSP projections to three dimensions and show, through quantitative and quality measurements, how the additional dimension favors distinction of data groups as compared to 2D.

Our second source of inspiration is an approach to visualize unstructured, point-based, multi-field volume data by Linsen et al. [2]. The authors' driving application is visualizing data from smoothed particle hydrodynamics simulations, where each particle carries multiple properties. They allow user-driven exploration of the feature space by applying a density-based clustering approach and then showing the clusters as density level sets placed in a 3D Star Coordinate layout. Selecting a cluster in feature space induces a specific segmentation of the data objects with respect to their underlying multi-variate function. Thus, surfaces are extracted from the volume based on the segmentation property induced by cluster membership.

We take the opposite perspective and consider the problem of interacting with 3D representations of multidimensional (usually non-spatial) data obtained with projections. 3D interaction in this scenario is difficult because projections typically produce sparse point distributions. Selecting and grabbing are, therefore, hard to accomplish. We handle the problem with interaction mechanisms that rely on enclosing 3D surfaces. They provide a mechanism for exploration tasks that allows the same type of control afforded by a 2D display.

We illustrate the approach with visualizations of document collections, a typical example of abstract high-dimensional data. The same solution copes well with spatial data, as illustrated by its application to particle simulation data as in Linsen et al. [2], [3]. We compare both processes and their results.

II. RELATED WORK

A. Projection-based Information Visualization

Classical Information Visualization relies heavily on visual representations embedded in 2D space. Multiple data dimensions may be anchored on 2D axes layouts, either arranged in parallel, as in Parallel Coordinates [4], or arranged radially, as in RadViz [5] and Star Coordinates [6]

Multidimensional Scaling (MDS) approaches project the data down to a low-dimensional visual space. Known as multidimensional projections, such techniques establish no direct association between the original data dimensions and the 2D axes. A set of points $S = \{p_1, \dots, p_n\}$ in $\mathbb{R}^m$ is mapped into a space $\mathbb{R}^d$, $d \leq m$ based on some criterion, e.g., attempting to preserve the original neighborhood (similarity) relationships. The projected points afford visual representations that reveal groups of similar/dissimilar elements, such as point clouds, 3D surfaces or graphs.

Several classical and novel techniques handle diverse high-dimensional data, e.g., Sammon's Mapping [7], FastMap [8], the Nearest-Neighbor Projection (NNP) [9] and the Least Square Projection (LSP) [1].

LSP has been conceived to handle large data sources characterized by sparse data distributions in high-dimensional spaces. The projection process comprises two steps. First, an *MDS* method is employed to project a number of 'control points' onto the low-dimensional space $\mathbb{R}^d$. Based on these projected points and on neighborhood relationships among the m -dimensional data points a linear system is built and then solved to obtain the projected coordinates of the remaining points.

The control points comprise a sample of S carefully chosen to reflect the data distribution in $\mathbb{R}^m$. The data points are clustered, and the cluster medoids are chosen as the control points. The clusters also define a neighborhood relationship, i.e., a list of neighboring points $V_i \subset S$ for each point $p_i \in S$. A point p_i is placed in the convex hull of V_i by generating a final layout based on local relations in $\mathbb{R}^m$. For each cluster, a nearest neighbor search of its medoid defines its k nearest clusters. When the nearest neighbors of p_i are sought, only the point's own cluster and its nearest clusters are examined. This approximation yields good results at a reasonable cost.

An advantage of LSP (and other multidimensional projections) is that dimensionality of the projected space is just an input parameter. Therefore, mapping to 2D or 3D visual spaces are equally possible. Nonetheless, since the standard display is a point cloud, the lack of a more compact geometry tends to impair user orientation and interaction in 3D. Unlike Scientific Visualization, Information Visualization solutions in general seem to favor 2D interaction over 3D. Some exceptions are briefly reviewed in the following.

B. Information Visualization in 3D

A few classical Information Visualization techniques have been extended to create data representations embedded in 3D. *Viz3D* [10], for example, extends *RadViz* by adding a third dimension represented by an axis orthogonal to the radial axes – the added spatial dimension improves group differentiation. In projection pursuit and grand tours the goal is to select and display a sequence of interesting 2D projections of pairs of data attributes. A 3D projection pursuit [11] and a 3D Grand Tour [12] have been introduced that generate cluster-guided 3D data projections and render the resulting visualizations in a CAVE virtual environment.

Additionally, several contributions on the problem of clustering high-dimensional data rely on visual representations, either in 2D or 3D, to assist or improve clustering decisions [13]. Likewise, visualizing high-dimensional data sometimes relies on clustering the data prior to visualization [14]. Nonetheless, pre-clustering has many subtleties in

this scenario, and data objects may not be easily associated with unique clusters.

Multidimensional projections on 3D spaces offer an interesting alternative to creating 3D 'general-purpose' representations of abstract high-dimensional data. They favor the perception of object similarity both across and within groups and also support visual identification of groups of highly correlated objects and user focusing on groups of interest. Furthermore, we hypothesize that projections on 3D space are likely to improve user perception of data groups, as compared to 2D. Since every dimension 'lost' in the projection process contributes to mixing objects actually unrelated, the added visual dimension may improve precision and enhance user exploration capabilities. To investigate this hypothesis we introduce a third dimension into LSP projections.

III. LEAST SQUARE PROJECTION IN 3D

The original LSP has been successfully applied to various types of multidimensional data, such as document and image collections [1], [15]. Exploration is conducted on a 2D display by locating, selecting and examining visible groups. Despite its effectiveness, visual clutter is severe in larger data sets, mixing different groups and impairing identification of subgroups.

The technique as proposed originally solves a linear system for each coordinate of the projected space. Extension to 3D (or any higher-dimensional space) is trivial as long as one can obtain the 3D coordinates of the control points. Choosing the control points and solving the resulting systems is analogous to the 2D case, only an additional system must be solved for the third coordinate. We modified the original algorithm to generate 3D projections, as follows. For choosing the control points a simple and fast k -means clustering is applied over the vector space. The control points are chosen as the cluster centroids (or medoids) and the cluster organization is then discarded. In order to obtain the 3D positions of the control points, a simplified version of a conventional force-based placement called Force Scheme [9] has been extended to handle 3D coordinates. Neighborhoods for each control point are defined with a k -nearest neighbor search. We refer to the original LSP paper [1] for details relative to each of these steps, as the 3D implementation handles them similarly.

IV. 3D LSP: RESULTS AND EVALUATION

Using LSP to obtain 3D projections improves both neighborhood preservation and identification of highly related data points. We consider the case of visualizing text collections, a typical example of abstract high-dimensional data. Visualization usually requires obtaining a vector space representation of the collection, a process that requires stop-word elimination, stemming (reducing words to their radicals), and term counting and weighting.

We consider initially a collection of 2,841 scientific papers (preserving title, abstract, authors, and affiliations) in 8 areas of knowledge, each represented by a different number of documents. The areas are: case-based reasoning, inductive logic programming, information retrieval, sonification, bibliographic coupling, co-citation analysis, milgrams, and information visualization. The latter has been extracted from the data made available for the IEEE InfoVis 2004 contest (10 years of InfoVis). The remaining articles were partly extracted from an internet repository¹ and partly built up from library and internet searches. After preprocessing, 1,269 terms define the resulting vector space, vector similarity is evaluated with the cosine measure.

The manual pre-classification of the papers in areas is mapped to color in the visual presentations. Although each paper has been assigned to a single class, some class overlap is expected, since many papers share common or related subjects and concepts. We expect this to be revealed in the visualizations, where papers have been placed based on their content similarity and not on their assigned class.

Figure 1 shows both 2D and 3D LSP projections of this corpus. Although it depicts 2D views of the 3D projection, it is noticeable that some groups are better defined, according to their color mapped class. For instance, in Figure 1(b) the separation between red and orange points is better than in the 2D view. One also observes better visual discrimination between yellow and light blue points. In Figure 1(c), which shows a rotation of the projection depicted in Figure 1(b), one also finds groups previously discriminated, such as the seemingly smaller green group and the two darker shades of blue marked on the picture.

We compare the 2D and 3D visual maps using two quality metrics known as *neighborhood hit* and *neighborhood preservation* [1], [16]. Both metrics consider the capability of a projection to preserve the data neighborhoods found in the original space, considering each data point. Neighborhood hit computes the percentage of a point's neighbors that have been human-assigned to its own class and averages the values. The neighborhood preservation metric computes the percentage of a point's neighbors, in the projected space, that belong to the same neighborhood in the original space, and averages the values. Results are shown in Figure 2: the blue curves show a clear improvement in precision in the 3D version, for any number of neighbors considered.

Although the 3D projections do improve group separation as compared to 2D, interaction becomes critical. We introduce a two-part strategy for exploring 3D projections aimed at fostering an interactive analysis process that may be progressively refined to meet user goals. Enclosing surfaces support focusing on groups of interest and coordination amongst 2D and 3D views of the same data helps identifying the relevant groups and drilling-down to sub-groups and

(a) Neighborhood hit metric for 2D and 3D LSP.

(b) Neighborhood preservation metric for 2D and 3D LSP.

Figure 2. Increased precision of 3D LSP as compared to 2D LSP, measured by Neighborhood Hit and Neighborhood Preservation, for the corpus of scientific papers depicted in Figure 1.

individual elements.

V. EFFECTIVE INTERACTION WITH 3D PROJECTIONS

Enclosing surface representations: 3D interaction is a non-trivial issue in visualization, and particularly difficult when the display depicts sparse point glyph representations. A strategy whereby groups of points are enclosed by selectable surfaces may assist refined data exploration. The projected data points are clustered in viewing space and each data cluster is then enclosed by a surface, as illustrated in Figure 3. Selected clusters may be pulled apart from the rest for further individual exploration. We investigate several approaches to generate enclosing surfaces and discuss their applicability in this context.

It is straightforward to obtain surfaces from the convex hull of the points: this solution is fast and produces simple meshes that are easy to manipulate. Figure 3(c) shows the result of applying this approach to the corpus of scientific papers (color still maps paper category). Such surfaces, albeit effective to convey the splitting of the data set, do not preserve the 3D cluster shapes. One loses interpretation capability in situations when cluster shape carries information about its relationships with the other clusters and the global space. However, convex hull surfaces suffice as a mechanism to trim down the presentation to groups of interest, a task particularly common in many applications.

¹ella.slis.indiana.edu/~katy/outgoing/hitcite/{bc,sc,mb,iv}.txt

Figure 1. LSP projections of a collection of scientific papers in 8 different areas of knowledge, showing evidence of better group separation in the 3D projection. Color maps the (human assigned) knowledge area.

An feasible approach to visualize non-convex point clusters is the blobs method by Blinn [17], who associates each point with a radial basis function and defines some parameters to control surface “blobbiness”. Along this line we have implemented a kernel-based approach to generate enclosing surfaces. Let the cluster be described as a set of points $\{p_i = (x_i, y_i, z_i) : i = 1, \dots, m\}$ after projection into 3D space. We consider the function

$$f_h(p) = \sum_{i=1}^m K\left(\frac{p - p_i}{h}\right), p \in \mathbb{R}^3,$$

where K is the kernel function and h is the bandwidth. We apply the marching cubes algorithm [18] to extract and render the boundary surface of the set

$$S(h, c) = \{p \in \mathbb{R}^3 : f_h(p) \geq c\}$$

where c is the iso-value. Parameters h and c are chosen so as to guarantee that $S(h, c)$ is connected and has a volume of minimum extension. In practice, we choose $K(p) = (1 - \|p\|^2)^2$ for $\|p\| \leq 1$ and $K(p) = 0$ otherwise, and the bandwidth h equal to the longest length of the minimum spanning tree of these m points. Let p_i and p_j denote two points, connected by the longest edge of the minimum spanning tree. Then, $(h = \|p_i - p_j\|)$. So, $S(h, c)$ is connected if $[p_i, p_j] = \{\alpha p_i + (1 - \alpha)p_j : 0 \leq \alpha \leq 1\} \subset S(h, c)$. This leads to $f_h(\frac{p_i + p_j}{2}) \geq 2K(\frac{p_i - p_j}{2h})$ and subsequently to the selection of the iso-value $c = K(\frac{p_i - p_j}{2h}) = \frac{9}{16}$. Figure 3(d) shows the enclosing surfaces obtained for the corpus of scientific papers – they follow the original blob model.

The previous solution creates nice renderings, but an adaptive adjustment of the radius of influence is necessary when clusters come close to each other. To overcome this problem, one has to incorporate more geometry information on the point cluster, e.g., by computing Voronoi diagrams [19]. Since their exact computation is not feasible for large point sets, several approximation algorithms exist. Approximation by discrete Voronoi diagrams is a reasonable alternative to

Figure 3. Clustering and enclosing surface generation, in visual space, of the same corpus depicted in Figure 1.

generate distance fields [20], [21]. Our third alternative to construct enclosing surfaces employs a 3D discrete Voronoi diagram computation approach [22] that produces non-convex hulls for non-convex clusters and enclosing surfaces that are arbitrarily close to the point cluster. It is based on a GPU implementation, henceforth it requires availability of the proper hardware.

First, a 3D discrete Voronoi diagram is computed from the points of the target cluster. The discrete distance field that is computed along with the Voronoi tessellation describes the distance to the points of the point cluster. Extracting isosurfaces from the distance field produces an enclosing

surface similar to the one computed above. The hull is then obtained by exploiting the natural neighborhood property induced by the Voronoi tessellation with respect to an appropriate isosurface of the distance field. Where three Voronoi regions come together, we generate a triangle connecting the points of the respective Voronoi regions.

This solution produces a surface that describes a hull of the point cluster, i.e., a triangular mesh whose vertices are a subset of the points within the cluster. This hull is non-convex for non-convex clusters. Results are visually similar to the well-known α -ball approach [23], but it is significantly faster. Figure 4(e) shows the outcome, in comparison to the convex hull and the density surfaces (Figures 4(c) and 4(d)), for a data set of papers (their title, abstract and authors) in four categories. The Voronoi-based non-convex hull surface description is optimal in the sense that it includes the minimal volume.

From an application point of view, one might prefer to render enclosing surfaces rather than hulls, whose points actually belong to the cluster. Starting from the hull, one can generate surfaces that are equidistant to it. We extend the discrete Voronoi diagram computation from point clusters to discrete Voronoi diagram computation from polygonal models in order to compute a distance field to the hull, obtaining a distance field to the hull. Thus, enclosing Voronoi-based surfaces can be generated with any distance to the hull by extracting isosurfaces from the new distance field. The resulting surfaces stick close to the point cluster, as shown in Figure 4(f). In fact, the isovalue (i.e., the distance) can be chosen arbitrarily small without the cluster breaking apart. From Figure 4, we may conclude that non-convex hulls and Voronoi-based surfaces tend to shape up the groups visually more closely to the point data distribution. The preferred view, in most situations, is a matter of user preference.

Coordination of 2D and 3D Views: The ability to coordinate multiple 2D and 3D projections also encourages novel exploration strategies, as illustrated by the examples depicted in Figure 5. Figures 5(a) and 5(b) show how coordinated 2D and 3D projections allow a user to brush the 2D view to highlight groups of interest in both. Users can locate well ‘resolved’ groups that are preserved in both spaces, distinguishing them from those strongly affected by the reduction from 3D to 2D. They may also identify hidden sub-groups. In Figure 5(b) one observes a group of points in the 2D view that apparently also appears in the 3D view. However, a rotation of the 3D projection, shown in the detail, reveals that this group actually separates in two sub-groups.

The converse coordination, from 3D to 2D, illustrated in Figure 5(c), allows relating the location of individual points in 2D to a group observed in 3D. In that picture, the user interacts with an enclosing surface to select a group (selection is indicated by the wire-frame view); selected points (displayed with full opacity) are highlighted in the 2D view. Notice that user selection triggers the display of

the main topic addressed by the documents in the group (text retrieval), a facility offered by the text visualization tool.

Such coordinated multi-space environment supports drilling down in the data in multiple ways. One may select and explore individual groups and clusters, identify new clusters and draw new enclosing surfaces. By selecting a group, users may see its topic and place it apart for further investigation. In further refinements a group may be split into sub-groups, so that users gradually find their way to the documents that deserve further examination.

Figure 5. Coordinating 2D and 3D views for data exploration.

VI. EXPLORING THE FEATURE SPACE OF MULTI-DIMENSIONAL SPATIAL DATA

The same interaction framework applies to multi-field scalar data, as illustrated in Figure 6(a), which depicts visualizations of particle data. This is a time-varying spatial data set with a multidimensional feature space [24], output by the simulation of the propagation of an ionization

Figure 4. Projection, clustering, and surface generation in visual space of a 3D LSP on a data set with papers in four areas of knowledge.

front instability. Although the spatial data is sampled on a volumetric grid, data points are unstructured in multi-dimensional feature space. The data set includes multiple attributes, namely density, temperature, mass abundances of eight chemical species, and velocity. It spans 200 time steps, each with $37 \cdot 10^6$ points, resulting in 1.7 GB of data per time step. For performance purposes, data has been spatially sampled. For an unbiased sampling, random sampling is desirable. The result is a set of points randomly distributed in the volumetric space, where each point carries multiple properties.

The left frame in Figure 6(a) depicts a 3D Star Coordinates view of the data feature space, as in [2]. The data corresponds to one simulation time step. Our visualization, on the right frame, has been created by pre-clustering followed by projection in 3D using LSP – it conveys the same global shape of the data space, but it stresses local cluster formation and separation.

Users may narrow down the examination to particular coherent spatial regions by identifying and delimiting clusters interacting with their enclosing surfaces. This is exemplified in Figure 6(b), left and right frames, which display enclosing density surfaces generated for some 3D clusters, in Star Coordinates and LSP views, respectively. Users interact to define how many clusters and corresponding surfaces they want to extract.

VII. INTERACTIVE 3D EXPLORATION TOOL

The functionalities for multidimensional exploration described were gathered in a Dataflow style system (see Figure 7). It supports, in addition to the aforementioned exploration capabilities:

- selecting from several existing projection techniques to generate 2D views from the original data and perform coordinated 2D-LSP3D interaction;
- generating several projections from 3D to 2D, reproducing currently observed clusters in a more manageable interaction space – this is exemplified in Figure 7. An obvious possibility is to orthogonally project the current 3D view, alternatively one may apply PCA, MDS, or other dimension reduction schemes;
- generating a 2D frame from a current 3D viewing position for further exploration in 2D;
- coordinating every new window generated so that selection of groups affect all visualizations, including the object space view;
- combining groups of individual items and groups of surfaces to generate larger groups whose elements may be saved for further manipulation.

Data transformation functionalities may be accessed just by plugging the proper module chosen from a module directory, depicted in Figure 7. Interaction capabilities are

(a) Placement of particles with original grouping by property

(b) Surface view of particle groups, after clustering in visual space to obtain 30 groups, and zooming in

Figure 6. Visualizations of the feature space of the particle simulation data. Star Coordinates views on the left, LSP views on the right.

added to the viewer modules. Figure 7 depicts various 2D views and one 3D view, plus a visualization in object space, derived for exploring tensor brain data through fiber tracking. The object space visualization shows 19,000 tracks. The LSP 3D projection is generated from features extracted from the tensor set, with dissimilarity computed as Euclidean distances from the feature vectors. 2D projections may be obtained either from the 3D mapping or from the original feature set. By locating groups of neighboring points on a projection the corresponding fibers are highlighted in the object view and their corresponding points are highlighted in all projections currently open. The exploration process is illustrated in the video provided with this article.

VIII. CONCLUSIONS

We discussed how projections in 3D space offer a unifying framework for handling both abstract and spatial high-dimensional data. A recent similarity-based multidimensional projection technique was adapted to generate projections in 3D space, and we showed that the added dimension brings enhanced group differentiation capability to users as compared to 2D displays. We then proposed strategies for proper user interaction with the resulting 3D views. The interaction strategies combine information on post-clustering of the projected data with geometrical information derived

from the clusters to obtain enclosing surfaces. Several approaches were considered to extract surfaces that shape up the partition of the 3D space while retaining interactive rendering rates. Surfaces allow access to groups of elements and support data drilling down. Moreover, they provide an interesting summarization of the data.

An additional interaction facility is the smooth coordination of 2D and 3D representations, that was shown to help identification of subtle data patterns. Coordination circumvents the need for complex 3D manipulations while maintaining the capability of locating highly correlated elements. Different mappings from 3D to 2D obtained with various dimension reduction techniques may provide complementary views to locate groups of interest.

Finally, we argue that interactive 3D projections hold promise as a unifying framework to handle both abstract multidimensional data and multi-field scalar volume data. Two examples illustrate this argument: a visualization of scalar data in feature space of a particles data set, and visualization of particle traces from tensor data. Again, coordination of feature and (spatial) object space views favors cluster interpretation.

Future work includes handling hierarchical access to cluster structure, devising specific coordination schemes between feature and object spaces and implementing suitable summarization strategies to label enclosing surfaces, similarly to topic extraction to label surfaces in document data sets.

REFERENCES

- [1] F. Paulovich, L. Nonato, R. Minghim, and H. Levkowitz, "Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 3, pp. 564–575, May-June 2008.
- [2] L. Linsen, T. V. Long, P. Rosenthal, and S. Rosswog, "Surface extraction from multi-field particle volume data using multi-dimensional cluster visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 6, pp. 1483–1490, Nov.-Dec. 2008.
- [3] P. Rosenthal, T. V. Long, and L. Linsen, "Shadow clustering: Surface extraction from non-equidistantly sampled multi-field 3d scalar data using multi-dimensional cluster visualization," in *Winner of 2008 IEEE Visualization Design Contest, VisWeek 08 Conference Compendium*. IEEE Computer Society, 2008, pp. 64–65.
- [4] A. Inselberg and B. Dimsdale, "Parallel coordinates: a tool for visualizing multi-dimensional geometry," Oct 1990, pp. 361–378.
- [5] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley, "Dna visual and analytic data mining," in *VIS '97: Proceedings of the 8th conference on Visualization '97*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1997, pp. 437–ff.
- [6] E. Kandogan, "Star coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions," in *In Proceedings of the IEEE Information Visualization Symposium, Late Breaking Hot Topics*, 2000, pp. 9–12.

Figure 7. The system: screens illustrate possibilities for 2D-3D exploration of the feature and object spaces of a brain fiber data set.

- [7] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. 18, no. 5, pp. 401–409, 1969.
- [8] C. Faloutsos and K.-I. Lin, "Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *SIGMOD '95: Proceedings of the 1995 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM Press, 1995, pp. 163–174.
- [9] E. Tejada, R. Minghim, and L. G. Nonato, "On improved projection techniques to support visual exploration of multidimensional data sets," *Information Visualization*, vol. 2, no. 4, pp. 218–231, 2003.
- [10] A. O. Artero and M. C. F. de Oliveira, "Viz3d: Effective exploratory visualization of large multidimensional data sets," in *SIBGRAPI '04: Proceedings of the Computer Graphics and Image Processing, XVII Brazilian Symposium*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 340–347.
- [11] G. Nason, "Three-dimensional projection pursuit," *J. Royal Statistical Society, Series C*, vol. 44, pp. 411–430, 1995.
- [12] L. Yang, "3d grand tour for multidimensional data and clusters," in *IDA '99: Proceedings of the Third International Symposium on Advances in Intelligent Data Analysis*. London, UK: Springer-Verlag, 1999, pp. 173–186.
- [13] A. Hinneburg, D. A. Keim, and M. Wawryniuk, "Hd-eye: Visual mining of high-dimensional data," *IEEE Comput. Graph. Appl.*, vol. 19, no. 5, pp. 22–31, 1999.
- [14] K. Boryczko, W. Dzwiniel, and D. A. Yuen, "Clustering revealed in high-resolution simulations and visualization of multi-resolution features in fluid-particle models," *Concurrency and Computation: Practice and Experience*, vol. 15, no. 2, pp. 101–116, 2003.
- [15] D. M. Eler, M. Y. Nakazaki, F. V. Paulovich, D. P. Santos, G. F. Andery, M. C. F. Oliveira, J. B. Neto, and R. Minghim, "Visual analysis of image collections," *The Visual Computer*, vol. 25, no. 10, pp. 923–937, October 2009.
- [16] F. V. Paulovich and R. Minghim, "HiPP: A novel hierarchical point placement strategy and its application to the exploration of document collections," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1229–1236, 2008.
- [17] J. F. Blinn, "A generalization of algebraic surface drawing," *ACM Trans. Graph.*, vol. 1, no. 3, pp. 235–256, 1982.
- [18] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH 1987*. ACM Press, 1987, pp. 163–169.
- [19] G. Voronoi, "Nouvelles applications des parametres continus a la theorie des formes quadratiques," *J. Reine Angew. Math.*, vol. 134, pp. 198–287, 1908.
- [20] A. Meijster, J. Roerdink, and W. H. Hesselink, "A general algorithm for computing distance transforms in linear time," in *Mathematical Morphology and its Applications to Image and Signal Processing*. Kluwer, 2000, pp. 331–340.
- [21] M. Teichmann and S. Teller, "Polygonal approximation of voronoi diagrams of a set of triangles in three dimensions," Tech Rep 766, Lab of Comp. Sci., MIT, Tech. Rep., 1997.
- [22] P. Rosenthal and L. Linsen, "Enclosing surfaces for point clusters using 3d discrete voronoi diagrams," *Computer Graphics Forum, EuroVis 09 Proceedings*, vol. 28, 2009.
- [23] H. Edelsbrunner and E. P. Mücke, "Three-dimensional alpha shapes," *ACM Trans. Graph.*, vol. 13, no. 1, pp. 43–72, 1994.
- [24] D. Whalen and M. L. Norman, "Competition data set and description," in *2008 IEEE Visualization Design Contest*. <http://vis.computer.org/VisWeek2008/vis/contests.html>, 2008.

APÊNDICE

B

Apêndice B

A Fast Projection Technique and its Applications to Visualization of Large Data Sets

Fernando V. Paulovich¹, Danilo M. Eler¹, Jorge Poco¹, Luis G. Nonato¹,

Charl P. Botha² and Rosane Minghim¹

¹ICMC/USP, São Carlos/SP, Brazil

{paulovic,eler,jpocom,gnonato,minghim}@icmc.usp.br

²TU Delft, Delft, The Netherlands

c.p.botha@tudelft.nl

Abstract

Large multivariate data sets still challenge visualization techniques and their interactive capabilities. In this paper we present a new faster formulation of a high precision projection technique that allows much larger data sets to be handled and interacted with. By using this novel approach, named Piecewise Least Square Projection (P-LSP), we provide new and effective ways of exploring the data set through its feature space. The paper shows the use of P-LSP to help data selection by the user during exploration using coherence amongst data items. Examples are shown for images and for volumes as large as 1,000,000 voxels, with as many as 14 variables per voxel, although the technique can also be used for any data set for which a reasonable feature space can be determined.

1 Introduction

Visualizing large multi-variate data sets presents many challenges. Although visualization and exploration techniques have come a long way reducing complexity by sampling and segmenting data sets in various ways, there are many applications for which evaluating the whole data set is necessary, and, for those, current techniques are limited.

For instance, in the case of volumetric data sets, tuning up the parameters for an adequate 3D visualization and progressive exploration may be time consuming and in many cases frustrating, particularly where various scalar fields or fields of mixed nature (scalar, vector, tensor) are present.

Various researchers have spent effort devising interesting ways to deal with multi-valued volumes interactively. Various of the resulting approaches employ techniques derived from multidimensional visualization. They provide a way of accessing the volume through interacting with its feature space whilst simultaneous visualization takes place in object space.

For most of the available techniques, however, there is need to re-sample in lower dimensions and to select or permute attributes in consecutive or side by side visualizations, causing the need for multiple passes through the whole data set and parameter set before the data can be fully analyzed. This paper proposes a novel way to interact with data in feature space that solves some of the drawbacks of the previous approaches.

In order to do that, we have developed a new formulation of a high precision multidimensional projection technique. The original technique, called LSP, is already capable of mapping data sets of many variables with high precision in a 2D visual space, faster than other known projections. The resulting map is a set of points in 2D in which highly related items are projected in the same neighborhood. Ideally, when the preprocessing is adequate and the original multidimensional space allows, there is a formation of groups of highly related individuals and those are separated from one another. These “properties” of a good projection (grouping and separation) can be reasonably measured. Although very precise and faster than other mapping techniques, LSP is noneffective when dealing with massive data sets due to its high processing time. The new approach presented in this work, called *Piecewise Least Square Projection (or P-LSP)*, brings a significant improvement on the processing time while still keeping the precision of the original LSP method.

Due to its effective processing time, P-LSP turns out a viable alternative for problems involving large data visualization and exploration, as it provides a very intuitive mechanism to manipulating volumes indirectly by mapping voxel’s attributes into visual space and then echoing the manipulations to object space. The same process can be applied for images or other spatially sampled data, as is illustrated in this paper. As will be seen, once the feature space is mapped to the visual space (usually the two-dimensional plane), interaction becomes quite natural, enabling to accomplish many different exploratory tasks, such as finding feature patterns of interest.

In the case of very large voxel data sets, we also propose, implement and illustrate a supporting strategy. We first partition the object space through a watershed algorithm, then we project a representative of each watershed basin in feature space to visual space. Interactions with the projections in visual space cause corresponding basins to be rendered back in object space. This results in a very interesting alternative for large data set manipulation.

Therefore, we can summarize the contributions presented in this paper as:

- a fast and high-precision projection technique able to handle large data sets is presented. This new approach is 2 to 3 orders of magnitude faster than its precursor while still preserving the quality of the produced layouts;
- a novel way to interact with data in feature space and assess the results on the object space is provided. This new methodology is particularly interesting to explore and discover patterns on multimodal volumetric data;
- finally, a mechanism based on watershed classification is presented to handle very large volumes, improving interaction substantially.

2 Related Work

Volume visualization is faced with handling both multiple fields (various values of various natures - scalar, vector, tensor) per voxel as well as a large number of data points or voxels. The process of defining proper ways to fully explore a volumetric data set is a major problem for visualization. Selection of regions or objects of interest as well as defining proper transfer functions are part of the tasks the analyst may have to perform in order to extract all the information needed from a volume.

Due to the ability of information visualization techniques to handle multi-dimensional information, there have been efforts to employ them for the analysis of the association between variables sampled in the object space. One of the first examples of explicitly linking information visualization with volume visualization was the WEAVE system [GRW⁺00]. This idea was further expanded upon by Doleisch et al. who formalized the definition of features that could be used in linked 2-D scatterplots to control a linked multi-variate volume visualization [DGH03]. In both these cases, linked 2-D scatterplots are of two discrete features in the feature space. Blaas et al. enabled interactive 2-D projections of arbitrary linear combinations of features in their work [BBP07]. P-LSP takes this a step further by optimizing a 2-D placement of all dataset samples that takes into account distances in the high-dimensional feature space. In other words, points that are highly related in feature space will be projected to the same neighborhood in 2-D. In this way, *all* features are involved in the projection at the same time, not just a subset. The background of this idea is further explained in the following paragraphs.

Some recent developments in projection based visualizations, which is basically an approach to generate data maps by dimensionality reduction to a visual (2D or 3D) space, have improved the precision of these

techniques for abstract data sets with a varied number of attributes. In this work we propose to use an extension of one of those projection techniques as a means to interact with the full set of variables in a complete volume or image, as well as to extend the capabilities of projections techniques for handling larger data sets. We define the concept of a multidimensional projection technique next, and discuss the related work on those.

Consider a data set D containing n data instances represented by feature vectors on a m -dimensional space. A multidimensional projection technique, or point placement technique, can be defined as a function that maps each instance $d_i \in D$ into a graphical element (e.g. point, circle, etc.) embedded on a visual space (1D, 2D, or 3D). On the resulting visual layout, the relative positions of the elements reflect some type of relationship amongst the data instances, such as similarity or neighborhood given by a distance function defined on the m -dimensional space [PM08]. In this way users can employ their visual ability to recognize patterns and structures based on similarity. Exploration is usually driven by the location of groups and sub-groups of elements, that are (ideally) highly correlated if they happen to be ‘projected’ in the same neighborhood.

Two classical examples of projection techniques are *Sammon’s Mapping* [Sam64] and *Classical Scaling* (also referred as *Multidimensional Scaling (MDS)*) [CC00]. The former technique first defines a cost function based on the differences among the distances between the elements in the visual space and the desired distances calculated between the data instances in the m -dimensional space. Then it minimizes this function, employing its gradient in an iterative steepest descent process. In the latter technique, a double-centered distance matrix between all pairs of data instances is defined, and a spectral decomposition is applied to recover the cartesian coordinates of the elements in the visual space. Although these techniques yield highly accurate results in terms of distance preservation, both present high computational complexity, $O(n^2)$, complicating their application on large data sets.

Another well known strategy to create multidimensional projections, originally defined as a graph drawing heuristic, is *Force-Directed Placement (FDP)* [FR91]. In the FDP model, each instance is modeled as a particle and its position is determined as the place where the sum of forces acting over it, generated by all other particles (instances), is zero. These forces are proportional to the difference amongst the desired distances between the visual elements and the current distances. The final layout is obtained changing the position of the particles until the system reaches an equilibrium state. Since each instance is affected by all other instances, one iteration of the FDP model is $O(n^2)$. In order to reduce such complexity, Chalmers [Cha96] defined a linear iteration approach where the forces were determined using samples instead of the whole set of instances. However, as n iterations is needed to produce a final layout, this is still a computationally expensive $O(n^2)$ technique.

Aiming at reducing such complexity, Morrison et. al. [MRC02, MC04] create a *Hybrid Model* approximating the Chalmers technique. In this approach a small sample of the data instances is first projected to the visual space. Then the remaining ones are interpolated considering the most similar sampled instance. It reduces the complexity to $O(n \sqrt[4]{n})$. This process was further optimized by Jourdan and Melançon [JM04] yielding an $O(n \log n)$ technique. Although such approaches really reduces the complexity of the original model, the produced layouts normally present low similarity or neighborhood preservation, specially for high dimensional data sets, due to the employed approximations. The technique proposed here is also based on approximations, however we seek to maintain a balance between the quality of the projections and the final computational complexity.

Recently, a new approach, called *Least Square Projection (LSP)* [PNML08], was presented which tends to produce good results in terms of neighborhood preservation in a reasonable amount of time. In LSP, differentiating it from most alternative techniques, the similarity preservation is pursued on small neighborhoods and between groups of highly related instances, this being a suitable for high-dimensional data sets. However, its application to large data sets is limited due to the employed process to calculate the neighborhoods and the final placement of the visual elements. The technique presented here, named *Piecewise Least Square Projection (P-LSP)*, is an extension of LSP in order to enable the handling of larger data sets. In the following sections, the bottlenecks of the LSP are identified. We show how P-LSP addresses these, thus reducing its computational complexity and running times without sacrificing too much the quality of the resulting layouts.

3 From LSP to P-LSP

This section presents the new mapping technique called *Piecewise Least Square Projection (P-LSP)* method, which can be seen as a refinement of the *Least Square Projection (LSP)* method.

3.1 LSP Overview

The Least Square Projection technique relies on the assumption that each element p_i of a data set D can be written as a convex combination of its nearest neighbors in the mapped domain, as Euclidean plane for example. In more mathematical terms, let $N_i = \{p_{i_1}, \dots, p_{i_{k_i}}\}$ be the set of k_i nearest neighbors of p_i (we are assuming a distance measure is defined in D), and denote by $(x_{p_{i_j}}, y_{p_{i_j}})$ the coordinates of each element $p_{i_j} \in N_i$ when mapped to $\mathbb{R}^2$. Therefore, the two-dimensional coordinates of p_i can be computed

by:

$$(x_{p_i}, y_{p_i}) = \sum_{p_{i_j} \in N_i} \alpha_{ij} (x_{p_{i_j}}, y_{p_{i_j}}) \quad (1)$$

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

Each element in D gives rise to a vectorial equation as described in (1), which can be assembled into two homogeneous linear systems:

$$L\mathbf{x} = 0; \quad L\mathbf{y} = 0 \quad (2)$$

where $\mathbf{x}$ and $\mathbf{y}$ are vectors representing the x and y coordinates of the mapped elements and L the matrix derived from equation (1).

The sets N_i define a *Nearest Neighbors Graph (NNG)* of D , that is, a graph connecting each element in D to its nearest neighbors. It can be shown that the rank of L is $n - q$, where n is the number of elements in D and q is the number of connected components making up the NNG [SCO04]. Thereby, in order to ensure a single non-trivial solution for the linear systems (2), the NNG should have only one connected component, which can be ensured by adding new edges linking disconnected components of the NNG.

The lack of geometric information in (2) might lead to solutions that are difficult to interpret and analyse. LSP deals with this problem by adding *control points* with geometrical information (constraints) to the systems (2). Control points are representative data instances carefully chosen to stand for groups of highly correlated elements. The Cartesian coordinates of control points are obtained through high precision placement techniques such as *Classical Scaling* [CC00]. Since the control points comprise just a fraction of the original data, the high computational cost of conventional “projection” techniques is not a major issue.

The set of n_c control points lead to new linear systems $H\mathbf{x} = \mathbf{b}_x$ and $H\mathbf{y} = \mathbf{b}_y$, where H is a rectangular $n_c \times n$ matrix whose rows have an entry equal 1 in the corresponding control point column and 0 in the remaining entries; $\mathbf{b}_x$, $\mathbf{b}_y$ are the vectors containing the cartesian coordinates of the already mapped control points. The new systems are coupled to (2), resulting in rectangular systems $A\mathbf{x} = \mathbf{b}$, where $A = \begin{pmatrix} L \\ H \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} 0 \\ \mathbf{b}_x \end{pmatrix}$ (the same is valid for y coordinate).

The new full rank rectangular systems are solved in the least-square sense, that is, one finds $\mathbf{x}$ that minimizes $\|A\mathbf{x} - \mathbf{b}\|^2$. In practice, such minimal solution can be obtained by solving the normal equations

$A^T A \mathbf{x} = A^T \mathbf{b}$ (the same for y) by iterative solvers such as conjugate gradient [She94].

The LSP framework described above has three main drawbacks: (1) the NNG construction may generate a graph with multiple components, resulting in rank-deficient normal equations. Therefore, new edges linking the disconnected components must be inserted in order to ensure single solution. However, the new inserted edges might make uncorrelated elements neighbors, negatively impacting the final two-dimensional mapping; (2) nearest neighbors computation is by far the most costly step of LSP, impairing its usage in application involving large data sets; (3) for massive data sets, the systems solution can be memory and time consuming, impairing to use LSP in large scale problems or time varying data sets.

Aiming at overcoming the aforementioned problems, a new framework called *Piecewise Least Square Projection (P-LSP)* is presented, which is fully described on the next sections.

3.2 Piecewise Least Square Projection (P-LSP)

The reasoning behind P-LSP is to partition the original data set in smaller subsets in which the LSP framework can be efficiently applied. However, the data partition strategy gives rise to a series of issues that need to be addressed in order to make the piecewise approach as effective as the original LSP.

A first issue to be considered is how to partition the data set so as to keep similar instances grouped together while still ensuring a balance of the number of elements in each subset. Many different techniques have been proposed to partition data sets taking into account similarity and partition size. In the present work we opt to employ the so-called *bisecting k-means* [SKK00], since in dense data sets such iterative technique behaves quite similarly to other mathematically well founded partition methods, such as PDDP (which is based on SVD decomposition) [SB04], generating well balanced grouping comprised of similar elements. Furthermore, bisecting k-means avoids the burden of building and solving matrix decomposition, increasing its effectiveness in applications involving large data sets. In our implementation the subdivision is carried out until $k = \sqrt{n}$ groups are reached, since it is an upper bound for the number of groups in a data set [PB95], and produces a number of elements LSP can process efficiently.

The coherent mapping of subsets is another issue to be addressed. In fact, if the conventional LSP framework is applied to each subset independently, no guarantee can be given to ensure that correlated groups will be mapped close to each other in the final two-dimensional space. However, by handling the control points properly one can attain a global relationship among groups without losing the local processing benefit. More specifically, the global relation can be built as follows: Let C_{P_i} be a set of control points chosen from a subset P_i of the partition. In the LSP framework the points in P_i will be

mapped to $\mathbb{R}^2$ taken as basis the geometry supplied by the control points in C_{P_i} . Consider now the set $C_P = C_{P_1}, \dots, C_{P_k}$ comprised by the union of the control points picked out from all groups making up the partition. The set C_P can be seen as a new data set containing a fraction of the initial data, thus LSP can efficiently be applied to map the elements of C_P to $\mathbb{R}^2$. Therefore, the geometrical information assigned to the control points of a particular partition group has been computed in harmony with the control points of other groups, reintroducing the global correspondence among elements lost during the partition stage. The global control point mapping ensures that each group P_i will be mapped in accordance with the mapping of other groups, keeping similar groups close and putting apart dissimilar elements. In practical terms, $n_i = (|P_i|/n) * n^{0.75}$ control points ($|P_i|$ is the cardinality of P_i) are picked out in each partition group P_i . This expression for the number of control points has been reached through experimentations that took into account the mapping quality and computational efficiency.

The P-LSP as discussed above is also shown in compact form in Algorithm 1.

4 Results

The original formulation of LSP has been used in mapping data sets of various kinds, such as document collections, time series and sets of images with consistent level of success in terms of neighborhood preservation as well as appropriate group formation and group separation. For those cases, P-LSP has shown to be able to handle much larger collections, maintaining similar precision to the original formulation (see precision analysis at the end of this Section).

In this section we present the P-LSP approach as an alternative to handle exploration of multi-variate data sets of large quantity of units. We do so by demonstrating some results in visual analysis through similarity relationships between pixels (for images) and voxels (for volumes).

These examples show the effectiveness of a projection technique to help users attain insight on how different choices of pixels/voxels representations (features) affect the similarity between them, and how that similarity can support locating regions and items of interest inside the object under study. In addition we seek to demonstrate the performance of P-LSP for large data sets, and how the approximation employed affects the neighborhood precision of the final projection.

We start by presenting some results on pixel projection.

Algorithm 1 Piecewise Least Square Projection (P-LSP).

input: - D : data set to be projected.

output: - the projection of D .

procedure $PLSP(D)$

- 1: Create the partitions $P = \{P_1, \dots, P_k\}$ using the bisecting k-means algorithm, where $k = \sqrt{n}$.
 - 2: Set $C_P = \emptyset$
 - 3:
 - 4: **for all** $P_i \in P$ **do**
 - 5: Split P_i into $n_i = (|P_i|/n) * n^{0.75}$ clusters, $P_i = \{P_{i_1}, \dots, P_{i_{n_i}}\}$, using the bisecting k-means algorithm.
 - 6: Set $C_{P_i} = \emptyset$
 - 7: **for all** $P_{i_j} \subset P_i$ **do**
 - 8: $C_{P_i} = C_{P_i} \cup \{w_{P_{i_j}}\}$, where $w_{P_{i_j}}$ is the medoid of P_{i_j} .
 - 9: **end for**
 - 10: $C_P = C_P \cup C_{P_i}$.
 - 11: **end for**
 - 12:
 - 13: Map C_P using the original LSP.
 - 14:
 - 15: **for all** $P_i \in P$ **do**
 - 16: Map the partition P_i using the original LSP considering the geometry imposed by C_{P_i} .
 - 17: **end for**
-

4.1 Pixel projection

Figure 1 shows the images that are used for the examples in this section. Figure 1(a) shows an axial slice of a brain presenting a tumor, and Figure 1(b) shows a sagittal slice of a (different) head.

In an image, an individual data point is given by a pixel. Before applying P-LSP to explore an image, we have to transform each pixel of an image into a set of features, so that a vector space is built for all pixels. This step is followed by a similarity calculation between the pixels. Here we employ two different approaches. In the first, 12 intensity features are extracted. First, we calculate the mean, entropy and standard deviation for the whole image using a 3x3 mask. Then, for each pixel, we take the mean, variance and standard deviation for each of the three calculations plus intensity from the original image, also using a 3x3 pixel neighborhood. We add to this set the pixel position (x and y coordinates),

Figure 1: Images for the examples. a) Axial slice of a brain b) Sagittal slice of a head.

resulting in a total of 14 features describing a pixel. In the second approach, we extract 30 Gabor features [JF91, MM96] for each pixel, all within a 9×9 pixel neighborhood. The 30 Gabor features are defined by varying the frequency $(0, 2, 4, 8, 16, 32)$ and orientation $(0, \pi/3, \pi/6, \pi/2, 3\pi/4)$ of the filter.

In order to compare the effect of these different feature extractions, first we create features for the image in Figure 1(a) and generate different projections for the resulting data sets, that is, we project the pixels on the 2D visual space using an Euclidean similarity measurement over their features. Then, we apply the k-means clustering algorithm [Mac67] to group the projected points in $\mathbb{R}^2$. Finally, points on the projection and corresponding pixels on the image are colored according to the cluster they belong to. The results are presented in Figure 2. In both projections, the background, the brain and the boundary between them are well separated, with result using Gabor features being a little better. However, in terms of detecting the tumor, Gabor features are far better, and the projection based on them separates well the tumor as a set of points, while the projection using the intensity features fail to produce clear separation of the tumor. Interaction is done on the projection with effects shown on top of the image as color or highlight of some sort.

What these pictures show is that, considering a task (e.g. segmentation or classification) which seeks to identify this kind of tumor on images, employing Gabor features may lead to better results when compared to intensity features. Our method can of course be used with other distance functions that measure other types of distances between the employed feature vectors.

Another application where projections of pixels can be applied is for helping users understand the relationships between the results of a pixel-based image segmentation or classification algorithm and the set of features employed. Figure 3 shows an example of that. First we use the k-means algorithm to segment the image presented on Figure 1(b), considering the Gabor features extracted from it. Using the same features we create a projection, coloring its points according to the segmentation. Observe that there is a

(a) Using intensity features

(b) Using Gabor features.

Figure 2: Projections of different sets of features extracted from the same image. Pixels and corresponding points on the projections are colored the same, based on automatically extracted clusters. It is fast to note that Gabor features separates better the tumor from the brain when compared to intensity features.

match between the groups of points in the projection and the elements defined by the segmentation. That is, there is a clear separation between the colors of each segmented part in the final projection. This is an expected outcome since the k-means segmentation splits the image grouping the most similar pixels, and a projection technique defines visual representations which aims at close placing the most similar elements and far apart the dissimilar ones. In cases such as these, our approach can be used to converge to proper feature extractions and similarity calculations for support to other automatic or interactive tasks.

The similarity of various structures according to the employed feature space can be easily seen in the P-LSP projection. The further points are spatially apart, the less similar they are. The P-LSP projection in fact implicitly optimizes for this. For example, the blue skull in the image is very dissimilar from the green background: In the P-LSP projection, these two colors are separated by large distance.

Just as is the case for images, volumes can be analyzed through projections in terms of their most basic components, the voxels. In the next section we present some results in this regard.

Figure 3: P-LSP projection of the Head image. Colors are based on the segmentation of the image into 10 parts using the k-means algorithm. The projection indicates the similarity relationships between different segments.

4.2 Voxel projection

As for pixels in images, the first step in applying P-LSP for the visual exploration of voxels on volumes is to compose their feature vectors. For data sets with more than one sampled scalar field (or even vector fields for which magnitude bears significance), the values themselves could be used as features. In our first example, the set of features include a voxel position in the volume (x , y and z coordinates), its intensity and its gradient in a $3 \times 3 \times 3$ neighborhood, resulting in 5 different features.

Figure 4 presents an example of a P-LSP projection generated using this type of features extracted from a Papaya volume. A threshold filter was applied to help distinguish the colors of the core and the pulp. In Figure 4(a), three different groups of points, easily noticed on the projection, were manually selected. Figure 4(b) presents the surface fitting of this volume with the voxels colored according to the groups of selected points. The light blue points represent the papaya outer shell, and the beige ones are its inner volume. The dark blue points represent the background voxels – the colors in the volume are not exactly the same as in the projection due to the transparency employed. There is a very clear separation between the outer shell and the inner volume in the projection, indicating that the type of features employed to represent a voxel is suitable to split these parts. We have used the Euclidean distance between vectors to define the dissimilarity amongst voxels. The surface fitting in Figure 4(b) was generated using the DeVIDE visualization software [BP08].

A second example of voxel visualization is presented in Figure 5. To create these projections we use a sub-set of the original Head CT volume¹, considering slices 30 to 97. The remaining slices do not con-

¹The head volume is available at <http://www9.informatik.uni-erlangen.de/External/vollib/>

Figure 4: P-LSP projection from a papaya volume. The projection shows that the employed features separates well the outer shell of the papaya from its inner volume.

tain any voxels with information, they are only background. The resulting sample represents 1,114,112 voxels. Differently from the papaya volume, we use intensity features plus the voxel intensity and its gradient, resulting in 14 features. Figure 5(a) and Figure 5(b) show user selections representing a structure inside the brain and an external part of it, respectively. The selection of points that represent the skin, the cranium, and the meninges can be seen in Figure 5(c). Finally, Figure 5(d) shows all selected parts together. Even for a larger data set, the P-LSP was able to separate well the different structures present in the volume. Additionally, the projections indicate that the set of features employed to represent the voxels deem the internal structure and the external part of the brain very similar since these are closely positioned.

Once there are more points on a large projection than available screen space (we are using a conventional computer monitor), the overlap amongst points is unavoidable, impairing our perception of density in different areas of the projection and of the frontier between them. In order to reduce this problem, we use transparency when drawing the points on the projection presented on Figure 5(d). On this figure, we can verify that the structure inside the brain and its external parts are very dense, and the density of the blue points vary through the projection. It indicates that the employed feature space coupled with the similarity measurement define that the voxels representing the brain parts (in beige and green) are more similar internally than the voxels representing the skin, cranium and meninges (in blue) since they are more dense on the projection. In addition, it is possible to locate groups or sub-groups of high-related voxels on each colored area, which are not visible on the other projections, supporting more refined exploratory tasks.

A different example of applying P-LSP to explore volumetric data sets is shown in Figure 6. In this example we show the result of interacting on the projection of timestep 30 of the Visualization Con-

(a) Selection of an internal structure of the brain.

(b) Selection of the external part of the brain.

(c) Selection representing the skin, cranium and meninges.

(d) All above selections together.

Figure 5: P-LSP projections from a sampled head volume. The projections shows a good separation between the different structures of the head.

Figure 6: Visualization of a timestep of the Visualization 2008 data set. Even for a more complex multimodal volumetric data set, P-LSP groups well the inherent structures.

test 2008 data set [WN08]. This data set represents the simulation of the propagation of an ionization front instability and it includes 10 different attributes, the density, temperature, and mass abundances of eight chemical species in more than 36,000,000 sample points. The huge size impairs conventional exploratory techniques to handle this volume successfully. For performance reasons, we spatially re-sample this data splitting the volume into $4 \times 4 \times 4$ pieces getting the average of their attributes to represent a new point on the sample, thus reducing the sampling to 576,600 points. Figure 6(a) presents the projection of this data set. In this figure, two different groups of points are selected and colored. Based on this selection we generate the surface fitting (Figure 6(b)) presenting the selected voxels. This shows that even for a more complex multimodal volume data set the P-LSP groups well the inherent structures present on the volume.

Although P-LSP can handle large data sets such as these, and transparency helps to overcome the problems related to point cluttering, as the number of projected points increases further, the response time of interacting with the projections becomes too high for real-time applications. In the next section we present a solution to decrease the number of projected points in feature space, whilst still representing and visualizing all voxels in object space.

4.2.1 Watershed-based projection

Our strategy to decrease the number of points to project is to first segment the volume using the watershed algorithm [VS91, HO93], and then to project the generated basins (the segmented parts) instead of the voxels themselves. That is to say that there will be one point per basin in the resulting projection, instead

of one per voxel. We do not make use of the marker-based watershed, or of any other post-watershed merging, so the basins are small parts of the volume containing pixels that belong to similar structures. To extract image features of a basin, we employ it as a mask on the original volume, extracting intensity features for the set of voxels covered by it. Then, the mean, deviation and variance are extracted from the entropy, mean and deviation of this set of voxels, defining the features of a basin.

Figure 7 presents the projections of basins for the head volume. The number of projected instances is reduced from 1,114,112 voxels to 65,279 basins. Figure 7(a) and Figure 7(b) show the projection of the internal structure and external parts of the brain, respectively. Figure 7(c) presents the selection of the basins that mostly compose the skin, cranium and meninges. Finally, Figure 7(d) shows all selections together. In this last figure we employ transparency to reveal the density of the basins on the projection. The projection of the basins, as well as the projection of the voxels (Figure 5(d)), separate well the brain parts from the skin, skull and meninges, defining them on a more compact and dense area on the projection. It indicates that the features extracted from the basins are similar to the features extract from the voxels in terms of the differentiation between the brain on other parts of the head.

Depending on various filtering parameters, watershed basins sometimes group voxels that could be considered unrelated, creating artifacts in the surface reconstruction considering the selection of basins in the projection. This can be clearly seen comparing Figure 5(c) with Figure 7(c). In this case we have performed a slight amount of basin merging before applying the P-LSP. These artifacts can be reduced by applying even less merging. However, it increases the number of instances (basins) to be projected. Therefore, care must be taken in defining the parameters used for the watershed segmentation so as not to generate too few basins, which can impair the quality of the surface reconstruction, or too many basins, which may increase the interaction time with the projection. Still, although the quality of the view is not as sharp as the one produced by projecting all voxels, using the watershed partitioning together with the projection can definitely decrease computational costs and analysis time previous to the final visualization. This approach also represents a good alternative to complex algorithms for complete basin merging that make decisions automatically on what is the best way to combine different basins, leaving that processing out of the initial phases of analysis.

The times taken to run P-LSP on large volumes are quite motivating. The following Section discusses computational costs.

(a) Selection of an internal structure of the brain.

(b) Selection of the external part of the brain.

(c) Selection representing the skin, cranium and meninges.

(d) All above selections together.

Figure 7: P-LSP projection of basins from watershed segmentation of the brain volume. Projecting the basins reduce the number of points from 1,114,112 to 65,279 on the final layout, preserving the outcome on separating the head parts.

4.3 Computational complexity and running time

The P-LSP computational complexity can be described as $O(C + P + I)$, where 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 partition. The complexity of the bisecting k-means algorithm to create $\sqrt{n}$ clusters is $O(n\sqrt{n})$ [TSK05]. The complexity of projecting $n^{0.75}$ control points using LSP is $O(n\sqrt{n})$ – LSP is $O(n^2)$ when conjugated gradient is used to solve the system and $A^T A$ is ill conditioned. Considering that bisecting k-means produce clusters of approximately equal sizes [SKK00], we have $\sqrt{n}$ partitions with $\sqrt{n}$ elements. Using LSP to project all these partitions is $O(n\sqrt{n})$. Therefore the complexity of P-LSP is $O(n\sqrt{n} + n\sqrt{n} + n\sqrt{n}) = O(n\sqrt{n})$.

In terms of running times, P-LSP is two or three orders of magnitude faster than the original LSP considering the data sets used here. Table 1 presents the size and the number of features of each data set, and the running times for creating the projections using the P-LSP and the LSP on a CPU Intel[®] Xeon[®] 2.33GHz, with 32GB of RAM memory. The source code of P-LSP to solve the linear systems and the other necessary steps, such as clustering, nearest neighbors queries, etc., is the same used by the basic LSP. These running times were taken by executing twice each technique for each data set, getting the average value of the obtained running times. Some values for the LSP technique are missing since our implementation was not able to calculate the projections in a reasonable amount of time on the aforementioned machine. This illustrates the degree of scalability now possible with the P-LSP over that of the LSP.

Table 1: Running time in seconds.

Data set	Size	#Features	LSP	P-LSP
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
Vis 2008 contest	576,600	10	—	1,660.832

In the next section we analyze the precision of P-LSP and how little the layout has been affected by the approximations when compared to basic LSP.

Figure 8: Neighborhood preservation comparison between P-LSP, LSP, and the Hybrid model.

4.4 Neighborhood preservation analysis

In order to evaluate the results produced by P-LSP, we reproduce the *Neighborhood Preservation (NP)* measure [PM08], comparing its results with the original LSP and also with the *Hybrid Model* presented in Section 2. The NP of one projected point is obtained by calculating its k -nearest neighbors on the $\mathbb{R}^2$ and verifying the percentage of preservation of the neighborhood defined on the original space $\mathbb{R}^m$. The NP of a projection is given as the average between the NP of all points belonging to it. The results for the Brain Tumor data set (intensity features) is given on Figure 8, with varying the number of nearest neighbors. Similar results are obtained for other data sets.

The attained results are satisfying in that, although somewhat less precise than the ones provided by LSP for small neighborhoods, they still remain excellent - and slightly better for larger neighborhoods. The slight fall in precision can be seen as a result of partitioning the data set and projecting the individual partitions. Since there is not a guarantee that the most similar elements of all instances belong to the same partition – an outcome of any clustering algorithm – some similar instances may belong to different partitions. Although the control points place the most similar partitions close to each other on the final layout, for small neighborhoods some misplacement is expected. This is a problem inherent to any dimension reduction technique. Nevertheless, the results are close to the ones produced by the LSP, and are more precise than the ones produced by the *Hybrid Model*, the other projection technique that took into account optimizations for increase in performance. We believe that the reasonably accuracy penalty is a small price to pay for the orders of magnitude performance increase.

5 Conclusions and Future Work

We have presented the Piecewise LSP, a high precision projection technique that is able to efficiently project high-dimensional samples to 2-D in such a way that their neighborhood relations are maintained. For example, points that are close together in a high-dimensional feature space will be close together in the 2-D visual representation.

The technique is a significant extension of LSP technique [PNML08] in that it is up to three orders of magnitude faster and keeps the quality of the produced layouts in terms of neighborhood preservation, enabling the exploration of large multi-variate data sets. One of the examples shown is a volume with more than a million voxels and 14 features per voxel. Importantly, this technique involves *all* features in the projection, and not just a subset.

Our strategy for projection based exploration in feature space coupled with visualizations in object space is unique in the sense that it can help users locate structures and patterns of interest in full high-dimensional feature volumes, without having to treat or interpret individual attributes or groups of attributes, or to pre-cluster based on attributes.

We have also presented an approach for exploring volume data sets by pre-partitioning them with the watershed algorithm and then projecting the basins using the P-LSP. Although the P-LSP is capable of projecting the multi-variate voxels of a complete volume data set, basin projection offers several advantages in terms of interactivity and clarity of the projection. This strategy supports faster interaction with the volume visualization and works beautifully with the idea of projecting groups of voxels to form groups of basins in visual space.

Additionally, the examples have shown support for users to converge, through iterations, to proper definition of feature sets and similarity measures, suggesting that this approach is also useful for supporting pre-setting of parameters for other automatic and mining tasks, such as segmentation and classification.

As future steps in the development we intend to expand the tool and the techniques to handle vector and tensor data. Due to the various partitions present in the methods disclosed here, it is also possible that adequate parallelization of the code will help improve interactivity further.

It is also suggested by the behavior of the proposed technique that a multi-level combination of basins when partitioning the volume before (or after) projection could possibly help get rid of the artifacts caused by the watershed step.

Acknowledgements

Fruit images were provided by Embrapa - Brazilian Agricultural Research Corporation - São Carlos - Brazil.

References

- [BBP07] Jorik Blaas, Charl P. Botha, and Frits H. Post. Interactive visualization of multi-field medical data using linked physical and feature-space views. In Ken Museth, Anders Ynnerman, and Torsten Möller, editors, *Proc. Eurographics / IEEE-VGTC EuroVis*, pages 123–130, 2007.
- [BP08] C. P. Botha and F. H. Post. Hybrid scheduling in the device dataflow visualisation environment. In H. Hauser, S. Strassburger, and H. Theisel, editors, *Simulation and Visualization*, pages 309–322. SCS Publishing House Erlangen, February 2008.
- [CC00] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall/CRC, second edition, 2000.
- [Cha96] Matthew Chalmers. A linear iteration time layout algorithm for visualising high-dimensional data. In *Proceedings of the IEEE Visualization 1996 (VIS'96)*, pages 127–ff., Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.
- [DGH03] Helmut Doleisch, Martin Gasser, and Helwig Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *VISSYM '03: Proceedings of the symposium on Data visualisation 2003*, pages 239–248. Eurographics Association, 2003.
- [FR91] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.
- [GRW⁺00] D.L. Gresh, B.E. Rogowitz, R.L. Winslow, D.F. Scollan, and C.K. Yung. Weave: a system for visually linking 3-d and statistical visualizations applied to cardiac simulation and measurement data. In *Proceedings of IEEE Visualization 2000*, pages 489–492, 597, 2000.
- [HO93] W. Higgins and E. Ojard. Interactive morphological watershed analysis for 3d medical images. *Computerized Medical Imaging and Graphics*, 17(4/5):387–395, 1993.
- [JF91] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recogn.*, 24(12):1167–1186, 1991.

- [JM04] F. Jourdan and G. Melançon. Multiscale hybrid mds. In *IV '04: Proceedings of the Information Visualisation, Eighth International Conference on (IV'04)*, pages 388–393, Washington, DC, USA, 2004. IEEE Computer Society.
- [Mac67] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [MC04] A. Morrison and M. Chalmers. A pivot-based routine for improved parent-finding in hybrid MDS. *Information Visualization*, 3(2):109–122, 2004.
- [MM96] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.
- [MRC02] Alistair Morrison, Greg Ross, and Matthew Chalmers. A hybrid layout algorithm for sub-quadratic multidimensional scaling. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 152, Washington, DC, USA, 2002. IEEE Computer Society.
- [PB95] N. R. Pal and J. C. Bezdek. On cluster validity for the fuzzy c-means model. *IEEE TFS*, 3(3):370–379, 1995.
- [PM08] Fernando V. Paulovich and Rosane Minghim. HiPP: A novel hierarchical point placement strategy and its application to the exploration of document collections. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1229–1236, 2008.
- [PNML08] F. V. Paulovich, L. G. Nonato, R. Minghim, and H. Levkowitz. Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):564–575, 2008.
- [Sam64] J. W. Sammon. A nonlinear mapping for data structure analysis. In *IEEE Transactions on Computers*, volume C-18, pages 401–409, May 1964.
- [SB04] S. M. Savaresi and D. L. Boley. A comparative analysis on the bisecting k-means and the pddp clustering algorithms. *Intell. Data Anal.*, 8(4):345–362, 2004.
- [SCO04] O. Sorkine and D. Cohen-Or. Least-squares meshes. In *Proceedings of Shape Modeling International*, pages 191–199. IEEE Computer Society Press, 2004.

- [She94] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, University of California at Berkeley, August 1994.
- [SKK00] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *Workshop on Text Mining, 6th ACM SIGKDD International Conference on Data Mining (KDD'00)*, pages 109–110, Boston, Massachusetts, USA, August 2000. ACM.
- [TSK05] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.
- [VS91] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598, June 1991.
- [WN08] D. Whalen and M. L. Norman. Competition data set and description. In *2008 IEEE Visualization Design Contest*. <http://vis.computer.org/VisWeek2008/vis/contests.html>, 2008.