



João Antônio Recio da Paixão

Feature-preserving vector field denoising

Dissertação de Mestrado

Thesis presented to the Postgraduate Program in Mathematics of
the Departamento de Matemática, PUC–Rio as partial fulfillment
of the requirements for the degree of Mestre em Matemática

Advisor : Prof. Hélio Côrtes Vieira Lopes
Co-Advisor: Prof. Thomas Lewiner

Rio de Janeiro
Agosto 2010



João Antônio Recio da Paixão

Feature-preserving vector field denoising

Thesis presented to the Postgraduate Program in Mathematics of the Departamento de Matemática, PUC–Rio as partial fulfillment of the requirements for the degree of Mestre em Matemática. Approved by the following commission:

Prof. Hélio Côrtes Vieira Lopes

Advisor

Departamento de Matemática — PUC–Rio

Prof. Thomas Lewiner

Co-Advisor

Departamento de Matemática — PUC–Rio

Prof. Geovan Tavares dos Santos

Departamento de Matemática — PUC–Rio

Prof. Sinésio Pesco

Departamento de Matemática — PUC–Rio

Prof. Luiz Henrique de Figueiredo

Instituto Nacional de Matemática Pura e Aplicada — IMPA

Prof. José Eugenio Leal

Coordinator of the Centro Técnico Científico — PUC–Rio

Rio de Janeiro — Agosto 27, 2010

All rights reserved.

João Antônio Recio da Paixão

Bachelor of Science in Mathematics from Virginia Tech University.

Bibliographic data

Paixão , João Antônio Recio da

Feature-preserving vector field denoising / João Antônio Recio da Paixão ; advisor: Hélio Côrtes Vieira Lopes; co-advisor: Thomas Lewiner. — 2010.

43 f. : il. ; 30 cm

Dissertação (Mestrado em Matemática)-Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2010.
Inclui bibliografia

1. Matemática – Teses.
 2. Campo Vetorial Discreto.
 3. Remoção de Ruído.
 4. Caminhada Aleatória.
 5. Filtragem.
 6. Topologia de Campos Vetorial.
- I. Lopes, Hélio Côrtes Vieira.
II. Lewiner, Thomas. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Matemática. IV. Título.

CDD: 510

Acknowledgments

I would like to gratefully acknowledge the enthusiasm of my advisors Hélio Lopes and Thomas Lewiner throughout theses last two years. Hélio believed in me from the very beginning, giving me confidence, support, and freedom in this work. Thomas has been there for absolutely everything and in the process become a true friend. I am eternally grateful for my advisors. I am also forever indebted to another great friend and co-author of this dissertation, Renata Nascimento, who help me in every aspect of this work, I simply do not know how to thank her enough. I would like to thank the other co-authors, Marcos Lage, Fabiano Petronetto, Alex Bordignon and my professors Sinésio Pesco and Geovan Tavares for our discussions, their ideas, and their help. I would also like to thank

-CNPq, FAPERJ, and PUC-Rio for their financial support.

-Professor Luis Fernando Alzugarir for the PIV dataset used throughout this work.

-The entire staff at PUC-Rio, especially Kátia, Creuza and Otavio for their assistance with all types of technical problems.

-My colleagues at the mathematics department at PUC-Rio for their care and attention.

-All the teachers I had at Oga Mitá, Santa Teresa, Fieldstone, Pascack Hills, and Virginia Tech throughout the years especially Eric, Angela, Lolla, Marcos, Mrs. Heluk, Mrs. Turner, Mr. Fallon, Mr. Argeski, Mr. Goodman, Mr. Conroy, Mr. Postman, Dr. Brown, Dr. Parry, Dr. Shockley, and Dr. Chapman.

-My students, the people I learned the most from, especially the ones from the Math Emporium and C2 Education.

-My friends for their endless encouragement especially Sam, Henrique, Vitor, Pedro, Mark, Robbie, Dan, Steve, Jason, Vivian, Molly, Rodrigo, Megan, Mario, Jessica, Rudi, Morgan, Gustavo, Felipe, and José.

-My aunts, uncles, and cousins from the Recio and the Paixão families for always being there and the Simões da Paixão family, Trindade Recio family, Figueiredo Santos family, Mougis family, and Tardin family.

-My grandparents, people I truly look up to: Vovó Clara, Vovô Paulo, Vovô Joaquim e Vovó Joaquina.

-My genuine treasures, my sisters, Maria and Fernanda.

-And finally my mom and dad, Antonio and Denise, for simply every single thing in my life that truly matters, thank you.

Abstract

Paixão , João Antônio Recio da; Lopes, Hélio Côrtes Vieira; Lewiner, Thomas. **Feature-preserving vector field denoising.** Rio de Janeiro, 2010. 43p. Dissertação de Mestrado — Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

In recent years, several devices allow to measure real vector fields, leading to a better understanding of fundamental phenomena such as fluid dynamics or brain water movements. This gives vector field visualization and analysis new challenges in many applications in engineering and in medicine. In particular real data is generally corrupted by noise, puzzling the understanding provided by visualization tools. This data needs a denoising step as preprocessing, however usual denoising removes discontinuities and singularities, which are fundamental for vector field analysis. In this dissertation a novel method for vector field denoising based on random walks is proposed which preserves certain discontinuities. It works in a unstructured setting; being fast, simple to implement, and shows a better performance than the traditional Gaussian denoising technique. This dissertation also proposes a semi-automatic vector field denoising methodology, where the user visually controls the filtering scale by validating topological changes caused by classical vector field filtering.

Keywords

Discrete Vector Field. Denoising. Random Walk. Filtering. Vector Field Topology.

Resumo

Paixão , João Antônio Recio da; Lopes, Hélio Côrtes Vieira; Lewiner, Thomas. **Remoção de Ruído em Campo Vetorial.** Rio de Janeiro, 2010. 43p. Dissertação de Mestrado — Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro.

Nos últimos anos, vários mecanismos permitem medir campos vetoriais reais, provendo uma compreensão melhor de fenômenos importantes, tais como dinâmica de fluidos ou movimentos de fluido cerebral. Isso abre um leque de novos desafios a visualização e análise de campos vetoriais em muitas aplicações de engenharia e de medicina por exemplo. Em particular, dados reais são geralmente corrompidos por ruído, dificultando a compreensão na hora da visualização. Esta informação necessita de uma etapa de remoção de ruído como pré-processamento, no entanto remoção de ruído normalmente remove as descontinuidades e singularidades, que são fundamentais para a análise do campo vetorial. Nesta dissertação é proposto um método inovador para remoção de ruído em campo vetorial baseado em caminhadas aleatórias que preservam certas descontinuidades. O método funciona em um ambiente desestruturado, sendo rápido, simples de implementar e mostra um desempenho melhor do que a tradicional técnica Gaussiana de remoção de ruído. Esta tese propõe também uma metodologia semi-automática para remover ruído, onde o usuário controla a escala visual da filtragem, levando em consideração as mudanças topológicas que ocorrem por causa da filtragem.

Palavras-chave

Campo Vetorial Discreto. Remoção de Ruído. Caminhada Aleatória.
Filtragem. Topologia de Campos Vetorial.

Contents

1	Introduction	9
1.1	Motivation and contributions	9
1.2	Related work	10
1.3	Organization	11
2	Vector fields concepts	13
2.1	Continuous vector fields	13
2.2	Discrete vector fields	14
2.3	Detection and classification of singularities	15
3	Denoising by random walks	18
3.1	Random walk	18
3.2	Feature-preserving filtering	19
3.3	Implementation and results	21
4	Topology aware denoising	29
4.1	Methodology overview	29
4.2	Progressive filters and scale-space generation	32
4.3	Reconstruction	32
4.4	Results	33
5	Conclusion	38
	Bibliography	40

Ninguém educa ninguém, ninguém educa a si mesmo, os homens se educam entre si, mediatisados pelo mundo.

Paulo Freire, *educador.*

1

Introduction

1.1

Motivation and contributions

Computer simulations of mechanical phenomena heavily rely on vector field generation, sampling, visualization and analysis. For their validation several techniques allow for measuring such vector fields, generating discrete data, which helps understanding physical behaviors. For example, Particle Image Velocimetry (PIV) is concerned with the quantitative investigation of fluids by imaging techniques (24) and has been used typically in mechanical engineering, in particularly modern aerodynamics and hydrodynamics research (29). The problem is that such real data or even simulation is typically corrupted by noise, which harms further simulation and puzzles the interpretation. Therefore, the processing of such vector fields typically involves a denoising step.

Classical denoising approaches rely on local coherences. They further consider that noise has a vanishing mean, and thus can be cancelled by averaging a piece of data with its neighbors. This has a smoothing effect which is widely used to generate scale-spaces of scalar fields such as images. However, as opposed to scalar physical quantities, for which one expects a globally smooth behavior in real experiments, vector fields can present rapidly changing directions. In fact those discontinuities are generally the most interesting part to analyze: they correspond to interfaces in fluid simulation, structural tissues when measuring brain water movement, faults and fractures in geophysical interpretation of soils. The problem is that when applying the classical filters these features are generally removed.

We address this problem with our first contribution, in Chapter 3, where we propose a feature-preserving vector field filter based on random walks (21). This filter preserves the field's discontinuities while denoising. Moreover, this formulation allows for processing unstructured or irregularly sampled fields. Moreover the filter is intuitive because it has a probabilistic interpretation from the random walks formulation.

There is still a persistent issue with such vector field denoising. Such random walk filters along with other classical convolution filters (37), rely on the assumption that the information is present in the measured data at a constant scale compared to the noise. In practice, an optimal filter scale can be automatically or manually chosen from a scale-space such as a hierarchical representation of the original data obtained by successive applications of such convolution filters (15, 4). However, for real vector fields with rapidly varying noise levels, using a *single* scale may keep either both noise and information or neither, leading to a delicate tradeoff.

As our second contribution, this problem is addressed in Chapter 4 where we propose a topology aware vector field denoising methodology (18) that lets the user control the topological changes caused by classical vector field filtering by the use of a suitable interface. The main part of this user interaction provides local tradeoffs between information and noise. The reconstructed field is then a smooth combination of different denoising scales.

Instead of preserving the discontinuities, as in our first contribution, we focus on controlling the topology of the vector field while denoising. In a variety of applications, in particular fluid dynamics, the field's singularities are the main features to be considered (13) and the interpretation is eased by detecting and identifying its singularities, like sinks, sources and saddles. Such topological features give a *global* information of the field which guides the user to adapt *locally* the scale filter.

1.2 Related work

In this section we briefly present some of the research literature related to our two contributions, the random walk and the topology-aware denoising techniques. We review denoising in general and the specific use of scale-spaces on vector fields. Then we describe the use of random walks and topology aware techniques in the literature.

Denoising. Among vector field filtering techniques on structured grids, several are specifically dedicated to colored image processing (23). In particular, color image filters focus on the reduction of impulse noise (34, 27, 17). In geometry processing several works have been proposed for noise reduction on surface normals (33, 19, 30, 31). Recently, Westenberg and Erlt (37) proposed a 2D vector field denoising algorithm that suppress additive noise by thresholding vector wavelet coefficients. Their method is restricted to work only on a structured grid of points. They compare their method to Gaussian filters, as

we will do throughout this dissertation. Close to our method, a class of filters has been introduced as generalized random walks for images (32) and meshes (30, 31).

Scale-spaces on vector fields. Scale-space techniques have become popular in computer vision for their capability to represent the multi-scale information inherently contained in real data. In particular, Bauer and Peikert (1) use scale-spaces to track vortices in 2D-time dependent computational fluid dynamics simulations. Klein and Ertl (13) proposes a strategy to track singularities over multiples scales in order to evaluate the importance of the critical points to the analysis and interpretation of the vector field. The methodology developed in Chapter 4 employs such scale-space representations to let the user choose *locally* which scale to utilize for reconstruction.

Random walks. Random walk has many applications nowadays not only in visual computing but also in genetics, physics, medicine, chemistry, computer science, just to cite a few. The first work using random walks in computer vision is in the application of texture discrimination (36), and recently has been applied to image segmentation (8). In the field of image processing, random walk has been used to image enhancement (28) and filtering (32). The use of random walks in geometry processing was recently proposed by Sun *et al* (30, 31) for mesh denoising and also for mesh segmentation (14). Chapter 3 is inspired in their work where instead we deal with unstructured 2D vector fields.

Topology-aware techniques. Turbulent vector fields usually have structures in different scales which complicates their analysis. A possible solution to this problem is to analyze the topology of the vector field in order to automatically simplifying while keeping the most persistent features (35). Another strategy, proposed in this work, relies on the user knowledge of the vector field, letting him decide interactively which topological singularities to keep or to smooth. Such approach has already been proposed in the problem of surface reconstruction (12, 25).

1.3 Organization

This dissertation is organized as follows. Chapter 2 briefly introduces vector field topology and its application to discrete vector fields. In Chapter 3 the notion of random walks is introduced along with its interpretation. We then

describe how to use these concepts to build the meshless feature-preserving filter. At the end of the chapter we present the results and implementation details. In Chapter 4 we describe the methodology for topology-aware denoising, detailing the techniques used in it and how they fit together. Again at the end, results are presented and analyzed. Finally, Chapter 5 concludes the dissertation by pointing out some limitations with both denoising techniques and suggesting improvements to be made.

2

Vector fields concepts

This chapter is a brief overview of vector field topology and how it carries over to the 2D discrete setting. The notation is defined for both an unstructured and a structured (grid) discrete vector field. For the structured vector field, we present two different ways to detect and classify singularities.

2.1

Continuous vector fields

A *vector field* \mathbf{v} on a planar domain $\mathcal{D} \subset \mathbb{R}^2$ is a function assigning to each point $(x, y) \in \mathcal{D}$ a 2D vector $\mathbf{v}(x, y) = (v^x(x, y), v^y(x, y))$. Assuming that v^x and v^y are differentiable bivariate functions, then the *Jacobian* matrix of \mathbf{v} at point (x_0, y_0) is:

$$J_{\mathbf{v}}(x_0, y_0) = \begin{bmatrix} \frac{\partial v^x}{\partial x}(x_0, y_0) & \frac{\partial v^x}{\partial y}(x_0, y_0) \\ \frac{\partial v^y}{\partial x}(x_0, y_0) & \frac{\partial v^y}{\partial y}(x_0, y_0) \end{bmatrix}.$$

A point $(x_0, y_0) \in \mathcal{D}$ is *singular* for \mathbf{v} if $\mathbf{v}(x_0, y_0) = (0, 0)$. According to the Hartman-Grobman theorem (9), a singular point can be partly classified by looking at the eigenvalues of the Jacobian matrix at that point:

- If the real parts of both eigenvalues are strictly negative, then the singular point is a *sink*.
- If the real parts of both eigenvalues are strictly positive, then the singular point is a *source*.
- If the real parts of both eigenvalues are non-zero real number with different signs (one positive and one negative), then the singular point is a *saddle*.
- If the real part of one of the eigenvalues is zero, the singularity is of higher order.

2.2

Discrete vector fields

In this dissertation we deal with sampled data of a vector field \mathbf{v} . Such data can be either unstructured, dealt with in Chapter 3, or structured (grid), the focus in Chapter 4.

2.2.1

Unstructured vector fields

Given a set of L unstructured points $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_L\}$, where each point $\mathbf{p}_i \in \Omega \subset \mathbb{R}^2$ and the associated vector $\mathbf{v}_i(\mathbf{p}_i) \in \mathbb{R}^2$, denote the set of vectors $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_L\}$ by \mathcal{V} (see Figure 2.1). A *vector field* is a map $\mathcal{F} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that associates to each point $\mathbf{p} \in \Omega$ a vector $\mathcal{F}(\mathbf{p})$. Since the vector field does not have any structure, the first preprocessing step, before we can process it, is to define a graph G .

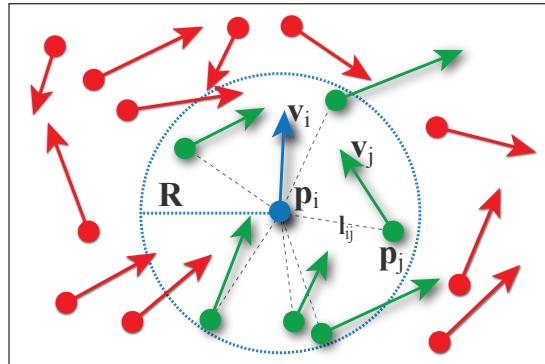


Figure 2.1: The support region and the induced graph

The nodes of G . There is a bijective mapping F from the nodes of graph G to the set of input points \mathcal{P} , which associates each node n_i on G the corresponding point \mathbf{p}_i on \mathcal{P} , i.e., $F(n_i) = \mathbf{p}_i$ and $F^{-1}(\mathbf{p}_i) = n_i$.

The links of G . The connectivity between the nodes of G depends on a model parameter R , which represents the radii of balls centered at each point of \mathcal{P} , the nodes that correspond to points that are inside the ball centered at \mathbf{p}_i are adjacent to n_i (see Figure 2.1). From now on, the graph defined for a given radius R will be denoted by G_R . More precisely, to define the links of G_R , the following rule is adopted:

“Node n_i is adjacent to node n_j through a link l_{ij} if and only if

$$\|\mathbf{p}_i - \mathbf{p}_j\| < R.$$

The *first neighborhood* of a node n_i , denoted by $N(n_i)$, is the set of all adjacent nodes of n_i on G_R .

Interpolation. When needed, we can interpolate the values in-between the sample points using SPH like interpolation (7) such as:

$$\sum_{\|\mathbf{p}_i - \mathbf{p}\| \leq R} W_R(\|\mathbf{p}_i - \mathbf{p}\|) \cdot \mathbf{v}_i$$

2.2.2

Vector fields on a grid

Now we will suppose that we have the values of vector field \mathbf{v} at the points (x_i, y_j) of a regular grid of size $M \times N$. We will denote $\mathbf{v}_{i,j} = (v_{i,j}^x, v_{i,j}^y) = \mathbf{v}(x_i, y_j)$, for $i = 1, \dots, M$ and $j = 1, \dots, N$.

Interpolation. In the grid we can also interpolate the values in-between the sample points. The simplest such interpolation is the bilinear interpolation, which can be written considering a grid cells as a unit square:

$$\begin{aligned} \mathbf{b}_{i,j} : [0, 1]^2 &\rightarrow \mathbb{R}^2, \\ \mathbf{b}_{i,j}(x, y) &= \mathbf{v}_{i,j} \cdot (1-x)(1-y) + \mathbf{v}_{i+1,j} \cdot x(1-y) \\ &\quad + \mathbf{v}_{i,j+1} \cdot (1-x)y + \mathbf{v}_{i+1,j+1} \cdot xy. \end{aligned} \tag{2-1}$$

When the vector field is given on a grid we can easily detect and classify its singularities, as we will see in the next section.

2.3

Detection and classification of singularities

Two classical approaches for the detection of critical points on a regular 2D grid are described and explicitly showed. The first one searches where the interpolation of the vector field vanishes. The second one computes the winding numbers.

2.3.1

Singularities of the interpolation

In the bilinear interpolation case, the detection boils down to solving the system of quadratic equations $\mathbf{b}_{i,j}(x, y) = (0, 0)$, where $\mathbf{b}_{i,j}$ is defined in Equation (2-1). This can be explicitly solved by computing the roots of a

polynomial in y :

$$\begin{aligned}
 & (-v_{01}^x v_{00}^y + v_{01}^x v_{10}^y + v_{11}^x v_{00}^y - v_{11}^x v_{10}^y + \\
 & + v_{00}^x v_{01}^y - v_{00}^x v_{11}^y - v_{10}^x v_{01}^y + v_{10}^x v_{11}^y) \cdot y^2 \\
 & + (-2v_{01}^x v_{00}^y - 2v_{00}^x v_{01}^y - v_{11}^x v_{00}^y - \\
 & - v_{01}^x v_{10}^y + v_{10}^x v_{01}^y + v_{00}^x v_{11}^y) \cdot y \\
 & + v_{00}^x v_{01}^y - v_{01}^x v_{00}^y
 \end{aligned}$$

To obtain the value of the x coordinate of the singular point we use the following expression:

$$x = \frac{(v_{00}^y - v_{01}^y) \cdot y - v_{00}^y}{(v_{00}^y - v_{10}^y - v_{01}^y + v_{11}^y) \cdot y - v_{00}^y + v_{10}^y}.$$

Sometimes this system degenerates to a lower degree polynomial. It can then have zero, one or two solutions. Each of them must be tested to lie in the quadrilateral. This method is the simplest way to detect first-order vector field singularities.

2.3.2 Winding numbers

The winding number counts the number of turns the vector fields achieves along a given closed curve Γ . It can be computed from the angular component of the vector field $\theta(\mathbf{v}(\mathbf{p}))$ at point $\mathbf{p} \in \Gamma$ by:

$$w_\Gamma(\mathbf{v}(\mathbf{p})) = \frac{1}{2\pi} \oint_\Gamma d\theta(\mathbf{v}(\mathbf{p}))$$

The winding number is zero if the region inside Γ does not contain critical points. If Γ contains a single saddle, then $w_\Gamma(\mathbf{v}) = -1$. If it contains a single sink or source, it will be $+1$.

The winding number is computed for each cell of the discrete grid using for Γ the square that bounds the cell. With the linear interpolation on the edges, we get the contribution of edge $(x_0, y_0) \rightarrow (x_1, y_0)$ to the above integral explicitly:

$$\begin{aligned}
 w_{00 \rightarrow 10} = & \arctan \left(\frac{v_{00}^{x^2} - v_{00}^x v_{10}^x - v_{00}^y v_{10}^y + v_{00}^{y^2}}{v_{10}^y v_{00}^x - v_{00}^y v_{10}^x} \right) \\
 & - \arctan \left(\frac{v_{00}^x v_{10}^x - v_{10}^{x^2} + v_{00}^y v_{10}^y - v_{10}^{y^2}}{v_{10}^y v_{00}^x - v_{00}^y v_{10}^x} \right)
 \end{aligned}$$

Summing over the four edges gives the desired winding number.

2.3.3

Singularity classification

To classify the singularities, as we mention in Section 2.1, one must look at the Jacobian matrix. Here we provide the explicit Jacobian matrix of the bilinear interpolation \mathbf{b}_{00} :

$$\begin{bmatrix} v_{11}^x y - v_{00}^x \bar{y} + v_{10}^x \bar{y} - v_{01}^x y ; v_{11}^x x - v_{00}^x \bar{x} - v_{10}^x x + v_{01}^x \bar{x} \\ v_{11}^y y - v_{00}^y \bar{y} + v_{10}^y \bar{y} - v_{01}^y y ; v_{11}^y x - v_{00}^y \bar{x} - v_{10}^y x + v_{01}^y \bar{x} \end{bmatrix},$$

where $\bar{x} = 1 - x$ and $\bar{y} = 1 - y$. The eigenvalues are directly computed using the trace and determinant of the matrix.

These tools are used in Chapter 4 to display to the user the topological changes such as the creation, destruction, or change of type of a singularity at a fixed grid point.

3

Denoising by random walks

In this chapter the feature preserving filter for vector fields based on random walks, our first contribution, is presented (21). In Section 3.1 the concepts of random walks in graphs are discussed in the context of Markov processes. Then in Section 3.2 these concepts are applied to build the filter. In Section 3.3 implementation details are discussed together with a brief suggestion on how the parameters can be chosen and the results are discussed at the end. This chapter follows the notation presented by Sun *et al.* (30).

3.1 Random walk

Random walk (RW) was one of the first chance-processes studied in the theory of probability and has gained a lot of attention in several areas in visual computing. The name random walk is used because one may think of it as being a model for an individual walking on a straight line who at each point of time either takes one step to the right with probability p or one step to the left with probability $1 - p$, for example.

Given a graph and a starting node, one selects one of its neighbor at random and moves to this neighbor then selects a neighbor of this node at random and moves to it and so on. This sequence of nodes selected randomly this way is a random walk on the graph. In Section 3.2 we see that the denoising method to be proposed applies random walks on a graph whose nodes are the base points of the vector field, and whose links represents the neighborhood relation between them. To do such random walk, a probability has to be assigned to each edge on the graph and this represents the chance to move from a vertex to its adjacent neighbor through an edge. In fact a Random walk on graph is a very special case of a Markov process (16).

A *Markov process* is a sequence of possibly dependent random variables (X_1, X_2, X_3, \dots) identified by increasing values of their index, commonly time. Its main property is that any prediction of the next value of the sequence (X_n) , knowing the preceding states $(X_1, X_2, X_3, \dots, X_{n-1})$, may be based only on the last state X_{n-1} . That is, the future value of such a variable is independent of

its past history: $P(X_{n+1} = x_{n+1}|X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_1 = x_1) = P(X_{n+1} = x_{n+1}|X_n = x_n)$.

When a Markov process is a sequence of discrete-valued variables it is called a *Markov chain* (20). The possible values of X_n are called the *state space* \mathcal{I} , which is a countable set and can be either finite or infinite. In this dissertation, the state space \mathcal{I} is finite and has L possible values. In the denoising method proposed here, L will represent the number of points in the input set.

A *transition probability* from state i to state j at the step n , where $i, j \in \mathcal{I}$, is equal to $P(X_{n+1} = j|X_n = i)$ and is denoted by $p_{i,j}(n)$. A Markov chain is *stationary* when the transition probability does not depend on n , that means: $P(X_{n+1} = j|X_n = i) = P(X_n = j|X_{n-1} = i)$.

The *transition probability matrix* $\Pi(n) \in \mathbb{R}^{L \times L}$ is the matrix whose the entry at the i^{th} row and j^{th} column is $p_{i,j}(n)$. Observe that each of its rows sums one. The probability that the Markov chain reaches the state i at the n^{th} time step is equal to $P(X_n = i)$ and is denoted by $p_i(n)$. The *probability distribution* of the Markov chain over all states at time n is represented by the vector $P(n) = [p_1(n), \dots, p_L(n)]$. Note that $\sum_{i=1}^L p_i(n) = 1$.

Given an initial probability distribution, denoted by $P(0)$, the distribution of the Markov chain in the first step is $P(1) = P(0)\Pi(1)$, and in the second step is $P(2) = P(1)\Pi(2) = P(0)\Pi(1)\Pi(2)$. So, after n steps, the distribution of the Markov chain is $P(n) = P(0)\Pi^n$ where $\Pi^n = \Pi(1) \cdots \Pi(n)$ is the n -step *transition probability matrix*. The entry at the i^{th} row and j^{th} column of Π^n is the probability of moving from state i to the state j after n steps, and is denoted by $p_{i,j}^n$. Observe that if the Markov chain is stationary, $\Pi(1) = \Pi(2) = \cdots = \Pi(n)$, so $\Pi^n = (\Pi(1))^n$.

3.2

Feature-preserving filtering

3.2.1

Problem description

Following the notation for unstructured vector fields from Chapter 2 we supposed that the vectors $\mathbf{v}_i \in \mathcal{V}$ are sampled from an unknown map \mathcal{F} and corrupted by an additive random noise. The problem is to develop a method that suppress the noise from the samples and maintains the relevant features of the vector field.

3.2.2

Random walk filter

The basic principle used in previous work (28, 30, 31), translated to a graph setting, is that the probability for moving from one node to its neighbor on the graph depends on how similar they are. Suppose that a single virtual particle i is located at every node $n_i \in G_R$, and that each particle i knows not only the position \mathbf{p}_i but also the vector \mathbf{v}_i . At each step of the random walk the particle moves from n_j to one of its neighbors or stays at its current position. After the application of n steps of these random walks, the L particles are redistributed on the graph according to the transition matrix Π^n . Such matrix induces a weighted average filter to be applied to each vector $\mathbf{v}_i \in \mathcal{V}$. The *random walk filter* computes, for each node n_i of the graph G_R , a new vector \mathbf{v}'_i , denoted by \mathbf{v}'_i , and is computed according to:

$$\mathbf{v}'_i = \sum_{j \in \mathcal{I}} p_{i,j}^n \mathbf{v}_j, \quad (3-1)$$

where $\mathcal{I} = \{1, 2, \dots, L\}$ and $p_{i,j}^n$ is the probability of moving from state i to the state j after n steps, which is the entry at the i^{th} row and j^{th} column of Π^n . The main question now is how to define the similarity functions for the transition matrices.

3.2.3

Similarity functions for vector fields

The idea to define the transition matrix $\Pi(n)$ is based on the fact that the larger the "difference" between two vectors is, the less similar they are. Sun *et al.* (30) suggest a set of similarity functions whose independent variable is the norm d of the difference between the normals of adjacent faces, like for example $s(d) = \frac{1}{C} e^{-\alpha d^2}$, where $\alpha \in (0, \infty)$ is a scale parameter and C is a normalization constant. When α is small, only faces with very close normals are considered similar. Thus, using such kind of similarity function, one has the property that the larger the difference between the normal vectors is, the smaller is the probability one should use to move a particle between the nodes. This function s is adopted in all examples of this dissertation.

A specific measure of similarity to cope with vector fields, inspired by Eibl and Brundle (6), is suggested here. For their vector field segmentation, they proposed three different measures for two given pairs of point/vector $\mathbf{f}_i = (\mathbf{p}_i, \mathbf{v}_i)$, $\mathbf{f}_j = (\mathbf{p}_j, \mathbf{v}_j)$:

- *Squared Euclidean distance* $\rightarrow d_1^2 = \|\mathbf{p}_i - \mathbf{p}_j\|^2 + \|\mathbf{v}_i - \mathbf{v}_j\|^2$

- *Mahalanobis distance* → $d_2^2 = (\mathbf{f}_i - \mathbf{f}_j)\Sigma^{-1}(\mathbf{f}_i - \mathbf{f}_j)^T$, where Σ is the covariance matrix of the coordinates of \mathbf{f}_k 's.
- *Weighted additive distances* → Given weights w_p , w_θ , w_r and w_β , $d_3^2 = w_p\|\mathbf{p}_i - \mathbf{p}_j\|^2 + w_\theta(\angle(\mathbf{v}_i, \mathbf{v}_j))^2 + w_r(\|\mathbf{v}_i\| - \|\mathbf{v}_j\|)^2 + w_\beta(\angle(\mathbf{p}_j - \mathbf{p}_i), \frac{1}{2}(\mathbf{v}_j + \mathbf{v}_i))^2$, where $\angle(\cdot, \cdot)$ is the angle between two vectors. Those weights balance the effects of each distances: the Euclidean distance from the base points, the vectors angle and norm difference and the difference of the points segment with the vector average direction.

After several experiments, we decided to adopt the *weighted additive distances*. The transition probability to move the particle from the node n_i to the node n_j at the n^{th} step is given by:

$$p_{i,j}(n) = \begin{cases} \frac{1}{C} e^{-\alpha d_{i,j}^2} & \text{if } n_j \in N(n_i), \\ 0 & \text{otherwise,} \end{cases} \quad (3-2)$$

where $d_{i,j}^2$ is the weighted additive distance between $(\mathbf{p}_i, \mathbf{v}_i)$ and $(\mathbf{p}_j, \mathbf{v}_j)$ and the value of the normalization constant is

$$C = \sum_{n_j \in N(n_i)} e^{-\alpha d^2}.$$

3.3 Implementation and results

3.3.1 Implementation

There are two ways to implement Equation (3-1). One is what Sun *et al.* (30) called the *batch* scheme, and the alternative one is what they called the *progressive* scheme. In the batch scheme the entries $p_{i,j}^n$ are computed by growing the neighborhood of the nodes, and computing for each step all transition probabilities, and use them at the end to compute the weighted average. In the progressive scheme, the algorithm runs step by step. It traverses only the first neighbors of the spot vertex and computes the probabilities for its neighbors. In the dissertation we use the progressive scheme, since it shows to be faster than the batch one in the majority of the experiments and the denoising requires only a few iterations. Moreover, the steps forms a scale-space, a notion we will intensively use in the next chapter.

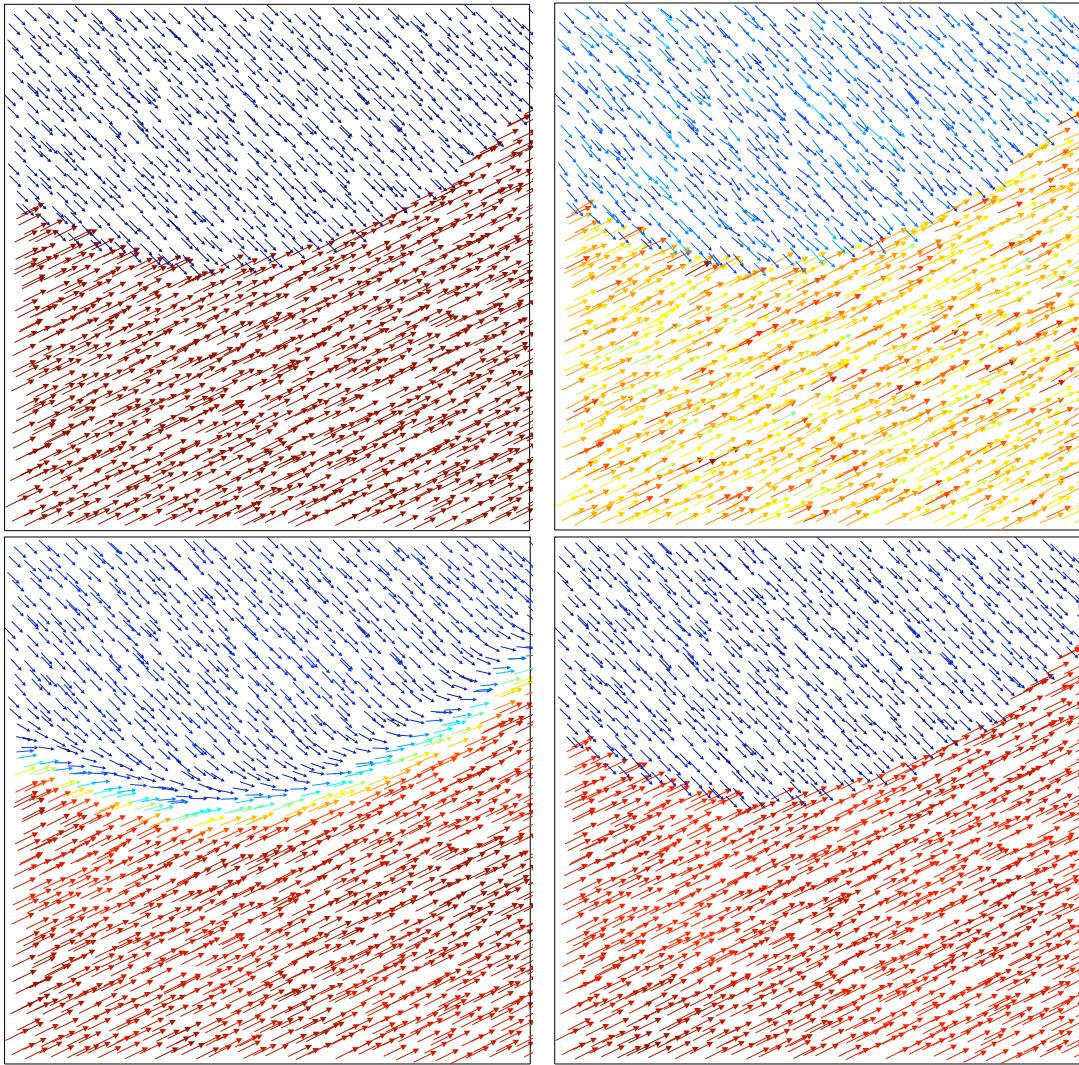


Figure 3.1: A simple discontinuous vector field (left) with gaussian noise added (center left). The gaussian filter (center right) blurs the interface, while the random walk (right) preserves it.

3.3.2

Parameters of the method

Besides the radius R used to construct the connections between the nodes of the graph, and the number n of steps for the random walk, there are more four parameters, the ones for the weighted additive distances: w_p, w_θ, w_r and w_β .

A suggestion for the weight w_p is $1/(2R^2)$, in order to give more weights to the points closer to each other in the ball of radius R . Notice that the term $w_p \|\mathbf{p}_i - \mathbf{p}_j\|_2^2$ naturally incorporates the distance between the base points, which is a nice advantage when the set of input points are unstructured. To fix parameter w_θ independently of the experiment, we optimize it for an average configuration: when the angles ($\angle(\mathbf{v}_i, \mathbf{v}_j)$) are uniformly distributed

in the interval $[0, \pi]$. Then one can set as default w_θ to be the variance of this distribution, i.e. $w_\theta = \pi/12$. Finally, if σ^2 is the variance of the lengths of the vector, then a suggestion for the value of w_r is $100/(2\sigma^2)$, since in this case it is considering the gaussian distribution with variance equal to the total variance over ten. Since the application is on denoising, as default the weight w_β is set to zero, because it usually destroys the interface of discontinuity of the vector field if it exists.

3.3.3 Results

We tested our denoising method on three kind of models: a simple noisy discontinuity test, where we expect the random walk to outperform the gaussian filter, measured vector field of physical systems and simulation models. For all examples of this section, we choose the parameters according to the suggestions presented in the previous section. We compare our method to a gaussian filter, which corresponds to a particular case by setting $w_p = (2R^2)^{-1}$, and $w_\theta = w_r = w_\beta = 0$. For all examples, we set $n = 2$ for both filtering methods.

Synthetic data This simple discontinuity test is constructed using a synthetic vector field:

$$\mathbf{v}(x, y) = \begin{cases} (2, 1) & \text{if } 10y < (x + 1)^2, \\ (1, -1) & \text{otherwise.} \end{cases}$$

The samples are created by evaluating this map on 900 base points that are randomly generated using a stratified distribution in the grid $[-3, 3] \times [-3, 3]$ (5). To each component of the sampled vectors we add an independent and identically distributed random gaussian noise with mean 0 and standard deviation equals to 0.05. Figure 3.1 shows that the gaussian filter blurs the interface, while the random walk nicely preserves it.

Simulation data We also checked our method on a simulation of shear bands in granular flows (3). The vectors on this example are placed on a 50×50 grid. The top picture of Figure 3.2 shows the equilibrium state of the mobility of grains in a dense granular system under shear, which almost half of the rows are moving one way, half moving the other way, with the shear band being formed at the very center. At this center area, the velocity is randomly distributed and its module is almost zero, resulting in a shear band. In this figure, for visualization purpose, the size of the vectors are the same, the colors are used

to represent their norm. In this example, the samples are originally with an unknown noise. The middle and the right pictures shows the filtered vector field by the gaussian and by the random walks method. The gaussian filter almost removes the shear band, while the random walk stresses it.

We finally checked our method on simulation models of a two-dimensional landslide and its impact into a water body (10), such data is available from SPHERIC (11). First our method is tested on an landslide measured by PIV methods (Figure 3.3) and then it is again tested on a SPH simulation data (Figure 3.4). We can see on both cases that the random walk matches the global behavior, sketched on Figure 3.4.

Measured data We perform a second test of our approach on a real data acquired from a PIV device. The top picture of Figure 3.5 shows the original data on $\Omega = [-1, 1] \times [-1, 1]$ with 15607 points. This sampled velocity field corresponds to a flow of water that is continuously injected vertically on the bottom right corner. The resulted vector fields after applying a gaussian and a random walk filter are illustrated, respectively, by the middle and the bottom pictures on Figure 3.5. On one hand, we see that the gaussian is more successful than the Random walk in removing the noise, however it destroys the singularities on the right near the wall. On the other hand, the random walk preserves the features but is unable to fully remove the noise on the left. There's a delicate tradeoff between noise and information in this data set which will be addressed in next chapter.

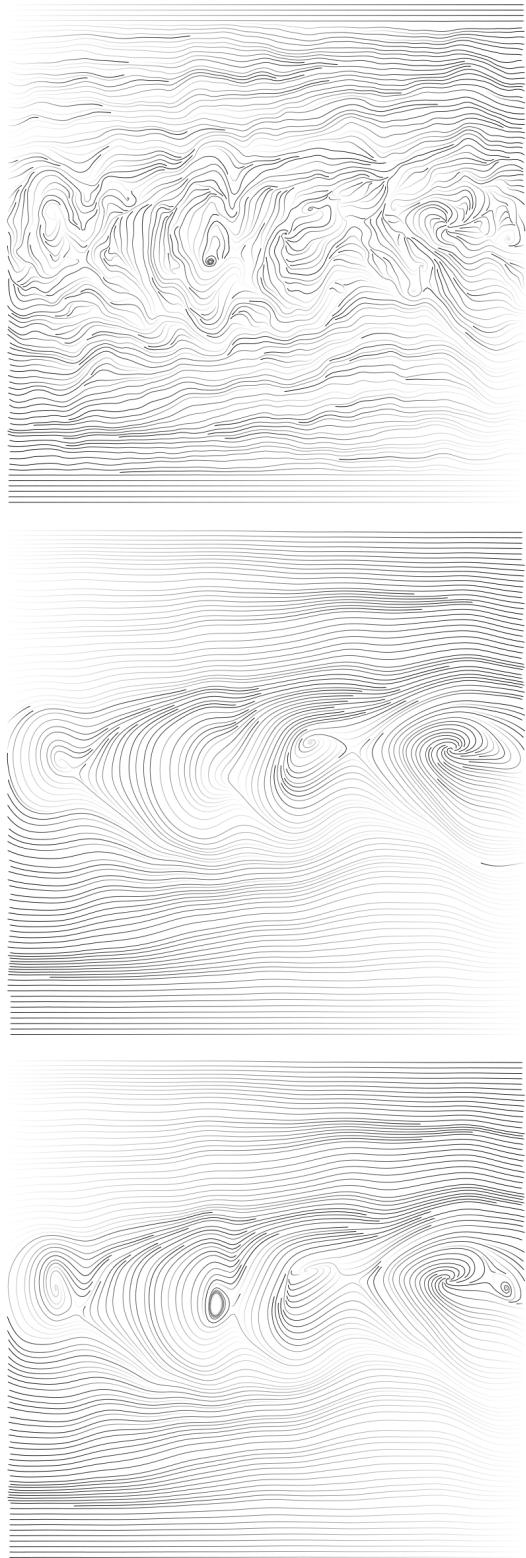


Figure 3.2: A shear band simulation of a granular flow (top): the gaussian filter (middle) removes the shear band, while the random walk (bottom) stresses it.

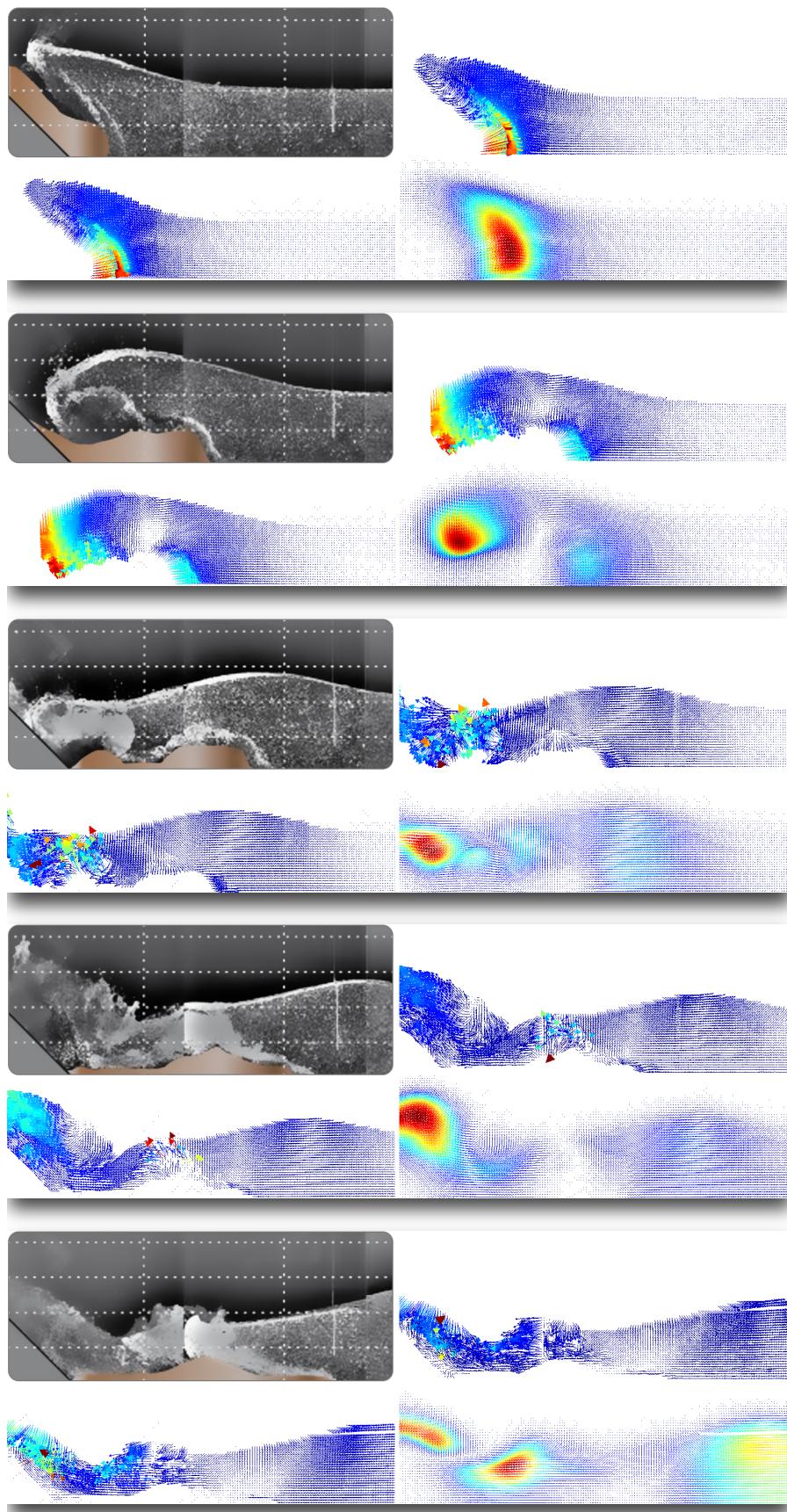


Figure 3.3: Landslide in 5 steps (each block): PIV measure (top left), simulated SPH vector field (top right). The bottom left and right pictures show the random walks and the gaussian filtered vector fields, respectively. The gaussian method oversimplifies the model.

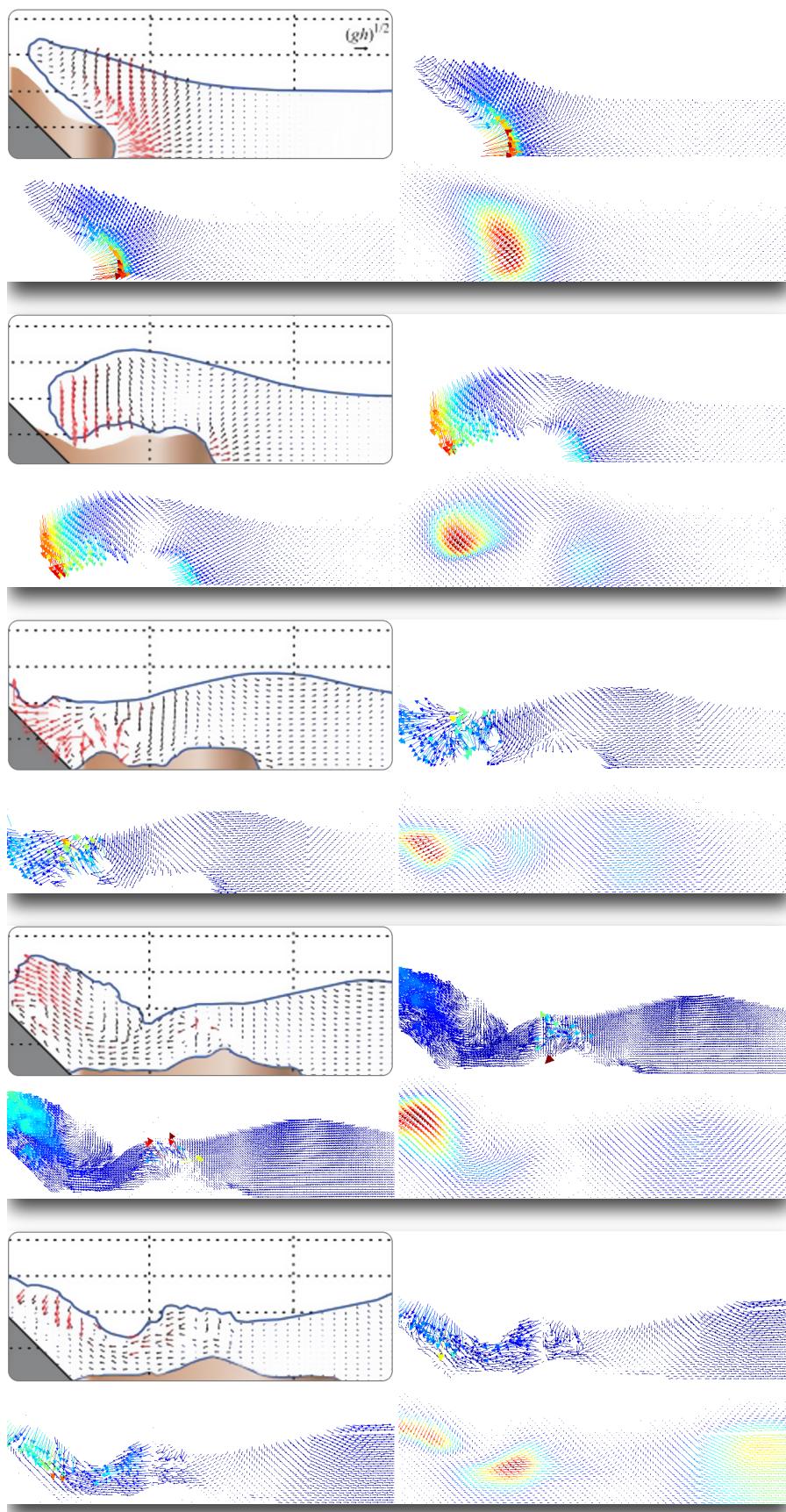


Figure 3.4: Sketched model of the landslide of Figure 3.3 (top left) that corresponds to the vector field on the top right rendered with a third of the samples. The bottom left and right pictures show the random walks and the gaussian filtered vector fields, respectively.

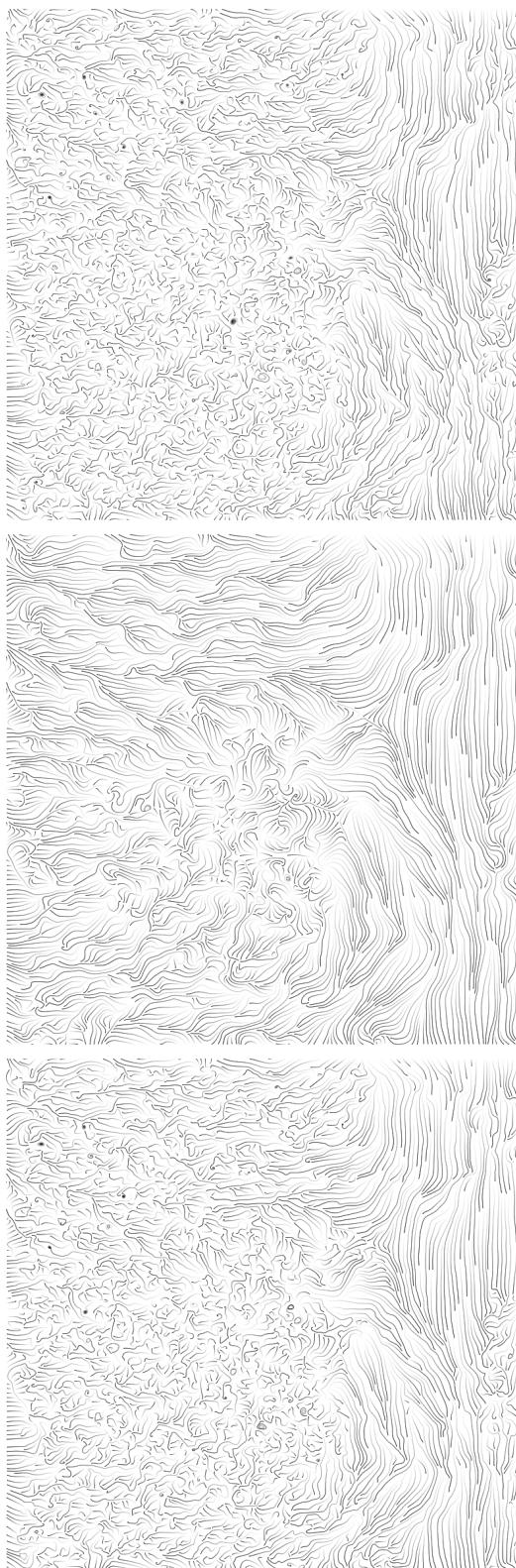


Figure 3.5: A PIV model of a fluid flow (left), filtered by a gaussian filter (middle) and by our random walk (right): while the gaussian removes the noise, it destroys the singularities on the right near the wall, the random walk preserves them but is unable to fully remove the noise on the left

4

Topology aware denoising

In this chapter our topology aware denoising technique will be detailed. This is a joint work with R. Nascimento (18). Instead of concerning ourselves with preserving the discontinuities of the vector fields as in the previous chapter, here we look at preserving singularities. There are four components in the methodology: generating the scale-space, detecting singularities in the scale-space, giving this information to the user, and finally reconstruct the desired vector field. A simple example to illustrate each component is presented in Section 4.1. Then the scale-space generation and reconstruction are presented in Section 4.2 and 4.3 respectively. At the end, in Section 4.4, the results are shown.

4.1

Methodology overview

The basic idea of our methodology is to let the user *locally* select the noise scale to remove, defining a scale parameter $s(x, y)$ at each point. We start by generating a scale-space from the original vector field and let the user choose a central scale s_0 . In order to avoid the arduous task of defining the scale parameter $s(x, y)$ sample by sample, we display to the user the singularities that appear or disappear at different scales nearby s_0 . When the user selects a topological change at a singular point (x_0, y_0) , we define $s(x_0, y_0)$ to be the closest scale to s_0 that reverts the change. Finally, we return the reconstructed vector field as a smooth mixture of different scales of the scale-space.

Before entering in detail for each step, let's illustrate our technique on the example of Figure 4.1. This field contains some relatively clean parts at the bottom, and noisy parts at the top. The singular points at the bottom should be retained, almost all the singularities at the top should be cleaned, except for a sink that many streamlines point to.

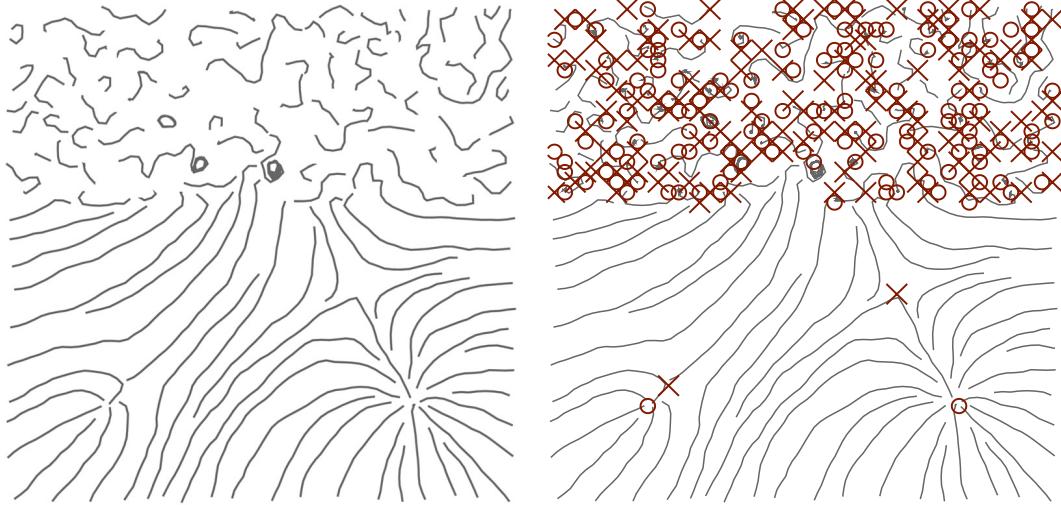


Figure 4.1: An artificial vector field represented by its streamlines (left) with its singularities marked (right).

Scale-space. In this example, we use a simple gaussian filter to generate a scale-space (see Figure 4.2). Our method can build on any denoising scale-space, as exemplified in Section 4.2 using isotropic or anisotropic filters.

Singularity detection. All the vector fields of the scale-space are available to the user at any time. We display the singularities of the field in each scale. There are different methods to detect singularities and our technique is independent of a specific choice of detection. As it can been seen in Figure 4.2, even though the field is still noisy at scale $s_0 = 10$, the meaningful singularity shown in the bottom left of Figure 4.1 was lost in the denoising process. The top part of the field is still noisy, needing more filtering.

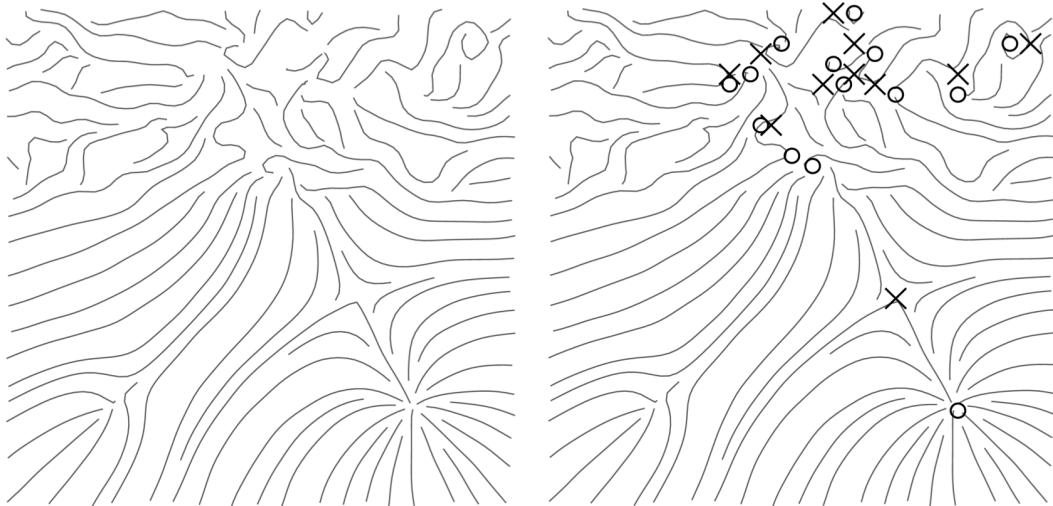


Figure 4.2: The vector fields at scale $s_0 = 10$ of its gaussian scale-space (left) with its singularities marked (right).

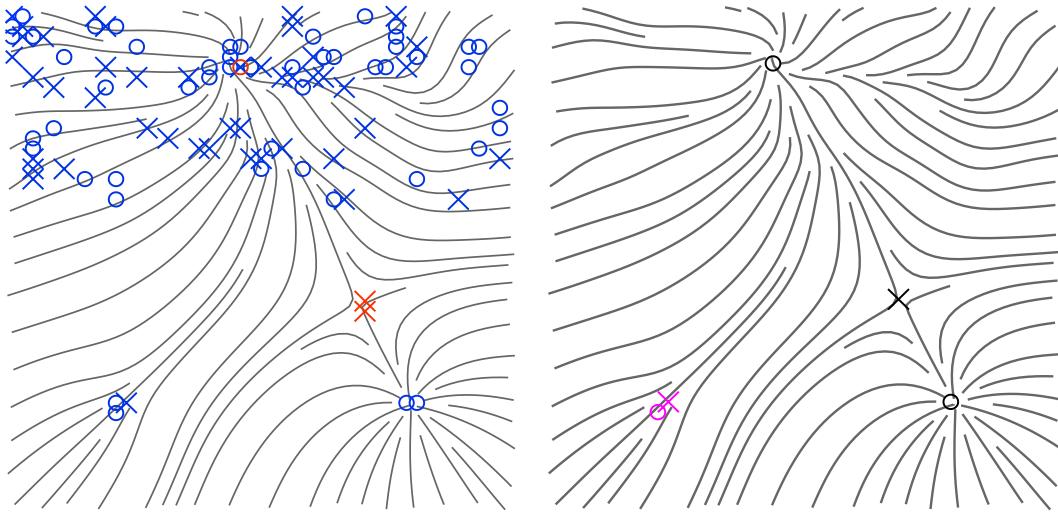


Figure 4.3: Our interface shows to the user the topological changes in nearby scales, here from 5 to 15 (left). The user then selects which topological changes he wants to revert (in purple on the right image).

Interface. In order to allow the user to denoise more of the top part while denoise less of the bottom to keep the meaningful singularity, we display to the user the topological changes at $s_0 = 10$ (see Figure 4.3). The user then selects which topological changes he wants to revert by a simple click.

Reconstruction. Each user selection defines a scale at the chosen point as the closest scale to $s_0 = 10$ that reverts the topological change. This gives a sparse sampling of the per-point scale parameter, which is smoothly interpolated to the whole domain. Our scheme supports different interpolations, and we provide two examples in Section 4.3. From this interpolation we can reconstruct an adaptively denoised vector field (see Figure 4.4).

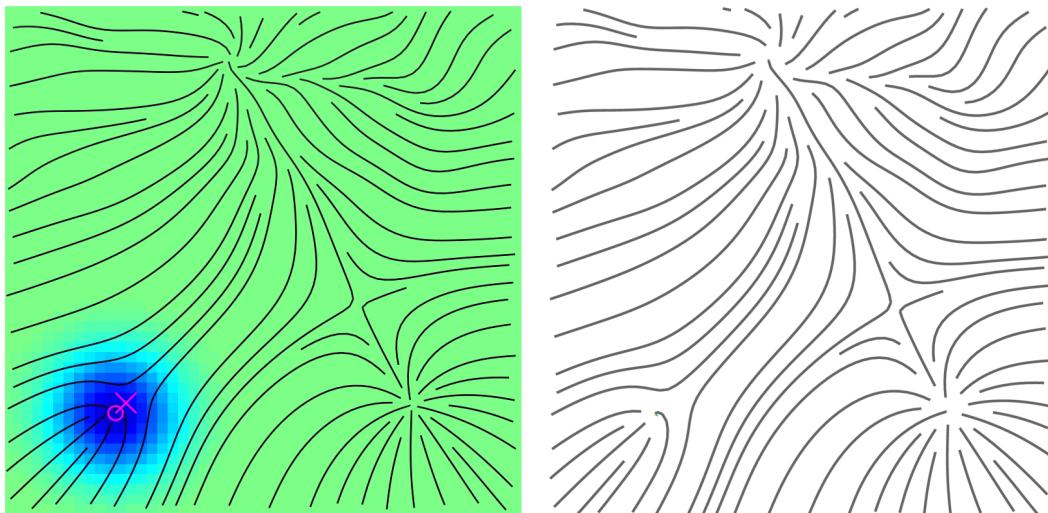


Figure 4.4: We finally interpolate the scales indicated by the user into a smooth function (left) which defines the reconstructed vector field (right).

4.2

Progressive filters and scale-space generation

The scale-space representation of the vector field is a collection of progressively denoised versions of the vector fields such as the one generated by the progressive method of Section 3.3. Each version is associated to an increasing scale parameter s . We denote $\bar{\mathbf{v}}(s, x, y)$ the vector value of the field at scale s and point (x, y) . The fundamental example of a scale-space on continuous vector fields is the gaussian scale-space, obtained by convolving with a gaussian kernel of increasing variance: $G_\sigma(x, y) = \exp(-\frac{x^2+y^2}{2\sigma^2})$: $\bar{\mathbf{v}}(s, x, y) = \mathbf{v}(x, y) * G_s(x, y)$ (13).

In the discrete setting, this convolving approach fits into the more general framework of random walks from Chapter 3, which ensures nice scale-space properties from local convolution masks. The scale parameter is then the number of convolutions applied or the number of steps in the random walk. We exemplify our editing interface using two types of similarity functions to generate the space-scale: the gaussian G_σ and the feature preserving similarity function from the previous chapter:

$$A_{\sigma, \tau}(x, y, \mathbf{v}) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \exp\left(-\frac{\|\mathbf{v}\|^2}{2\tau^2}\right),$$

which takes into account the direction of the vector field and better preserves discontinuities. The scale-space is then directly generated by the repeated application of a 3×3 mask with the above kernels.

4.3

Reconstruction

The singularities selected by the user provides a sampling of the scale function $s(x, y)$ on the domain. To reconstruct the whole vector field, we interpolate this sampling. Then denoting $\bar{\mathbf{v}}_{i,j}(s)$ the vector field sample at scale s , we define the reconstructed vector field $\tilde{\mathbf{v}}$ at grid point (x_i, y_j) by:

$$\tilde{\mathbf{v}}_{i,j} = \bar{\mathbf{v}}_{i,j}(s(x_i, y_j)).$$

Virtually any interpolation scheme works, although with different resulting qualities. If the interpolation is not smooth enough, the rapid changes in the scale parameter may create artifacts in the reconstructed field. Moreover, the interpolation must maintain the scale in a neighborhood of the singularity to preserve it. We implemented two methods for the interpolation of s that gave satisfactory results: radial basis functions (RBF), with gaussian basis,

and kernel Shepard interpolation (2) with gaussian kernel.

The RBF interpolation of $s(x, y)$ from the scales of the used selected singularities s_k at (x_k, y_k) is obtained by a least-squares minimization on the coefficients α_k of

$$\min_{\{\alpha_k\}} \sum_k \|s_{rbf}(x_k, y_k) - s_k\|^2, \quad \text{where} \quad (4-1)$$

$$s_{rbf}(x, y) = \sum_k \alpha_k G_\sigma(x - x_k, y - y_k). \quad (4-2)$$

The kernel Shepard method modifies the original Shepard interpolation (26) by using kernels instead of the Euclidean distance:

$$s_{ks}(x, y) = \frac{1}{\sum_k G_\sigma(x - x_k, y - y_k)} \cdot \sum_k G_\sigma(x - x_k, y - y_k) \cdot s_k.$$

A important property of this method is that the image is limited to $[\min_k s_k, \max_k s_k]$.

4.4 Results

In this section we present our experimental results on synthetic, simulated and measured vector fields. Since we work with relatively small 2D vector fields stored in regular lattices compared to the computing power of actual hardware, the interface responds in real-time to user interactions, except for the initial scale-space generation (see Table 4.1). In all the experiments presented here, the singularities detected by the winding number method and the bilinear one were the same, although they may differ in very particular cases.

Table 4.1: Timings, in milliseconds, for each step of the edition.

Data	Fig	Size	Filter type	ms)	Singularity type	ms)	Scale select	Reconstruction type	solve	eval
Analytic	7	2500	G_σ	18.9	w_Γ	98.0	7.3	KS		0.1
Granular	8	2500	$A_{\sigma,\tau}$	587.5	w_Γ	110.8	8.3	RBF	0.8	0.9
PIV 1	1	15624	G_σ	135.0	$\mathbf{b}=0$	947.6	65.6	RBF	0.1	7.6

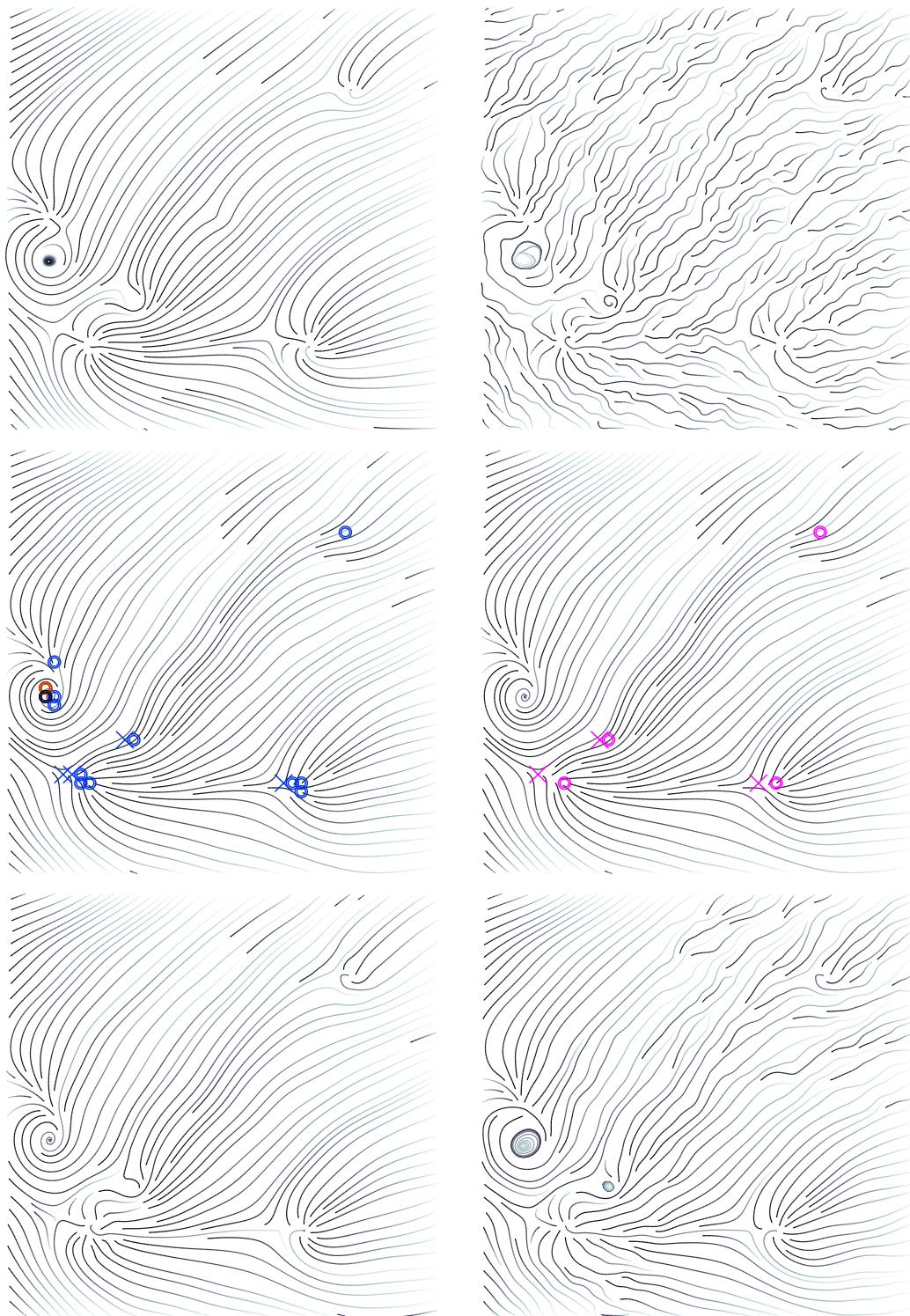


Figure 4.5: Experiments on an analytic vector field (top left) artificially corrupted by non-gaussian noise (top right). The user can choose between singularities that disappeared before scale s_0 (in blue) or singularities that could be smoothed out at scale $s > s_0$ (in red) (middle left). From the user selection (middle right), we reconstruct the vector field maintaining the selected scale in a small (bottom left) or larger radius (bottom right).

Synthetic data. We first validate our approach on a synthetic vector field, corrupted by an artificial, non-gaussian noise (see Figure 4.5). We can denoise adaptively the vector field, recovering the original singularities. We use gaussian scale-space with a kernel Shepard interpolation. Observe that, varying the σ of the kernel used in the reconstruction, we can carry larger portion of the fields at the selected scale.

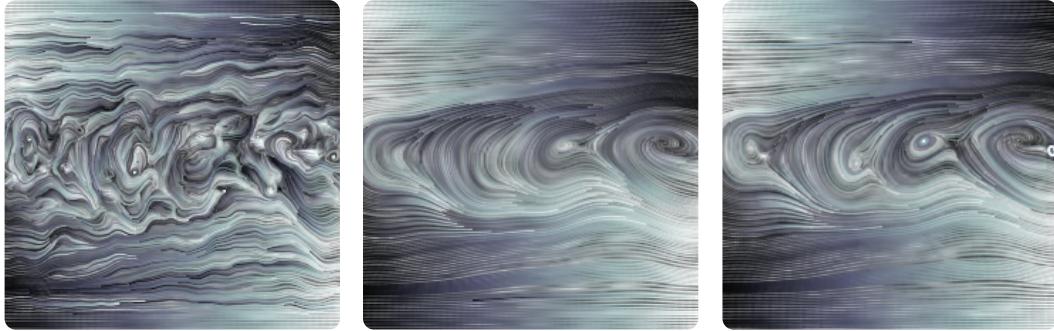


Figure 4.6: On a vector field from a simulated shear band granular system (left) 40 steps of denoising recovers the granular bands but loses one of the main vortices (middle). Selecting that vortex in our interface allow for a denoised reconstruction with the main singularities (right).

Simulation data. We then experimented on a vector field of 2500 samples issued by a granular simulation (3). The shearing of the granular system generates five main vortices between the shear bands, which are clearly visible in Figure 4.6 besides the noise. We use an anisotropic filter to generate the scale-space, requiring around $s = 40$ steps to denoise the granular bands at the top and bottom. However, this smoothens out one of the main vortices. Selecting it in our interface allows to reconstruct a clean vector field with the main singularities, using here the RBF interpolation.

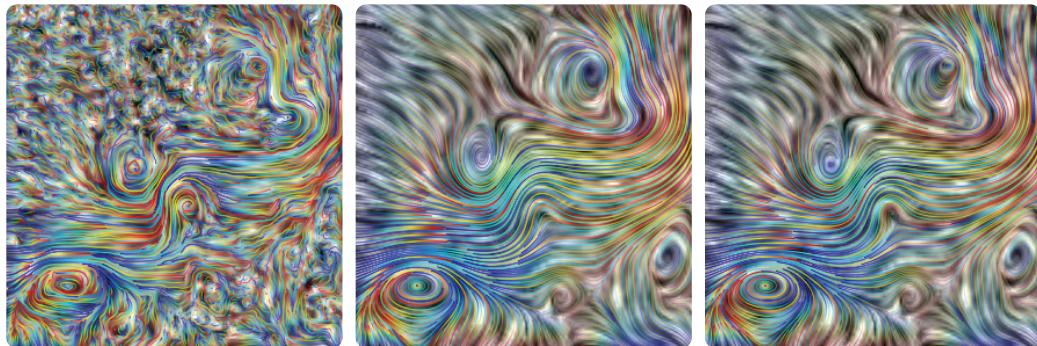


Figure 4.7: Topology-aware denoising of a measured fluid velocity field: (left) original field, (middle) gaussian denoising, (right) gaussian denoising preserving topological singularities selected through our interface.

Measured data. We finally experimented our method on real measured vector field of 15624 samples, acquired through PIV imaging. The experiments of Figure 4.7 and Figure 4.8 are measured from a wall-jet setup, where water is injected from the left of the image and kicks on the wall on the right. The images correspond to the top half of the jet. The water injection is stronger in the experiment of Figure 4.7 as compared to the one of Figure 4.8. In both cases, the top left part of the image is very noisy since there is less water, while the right part is turbulent. This leads to several important singularities on the right part of the field to disappear before the singularities caused by the noise. In the reconstructed vector field, those singularities are recovered. We used a gaussian scale-space for this experiment. While denoising this PIV data set with the Random Walk filter in Chapter 3, we saw how tricky it was since the information on the right side was considered noise and removed before the true noise on the left. Here, with our proposed methodology, we were able to get around the problem by selecting the right scales and reconstructing the desired field.

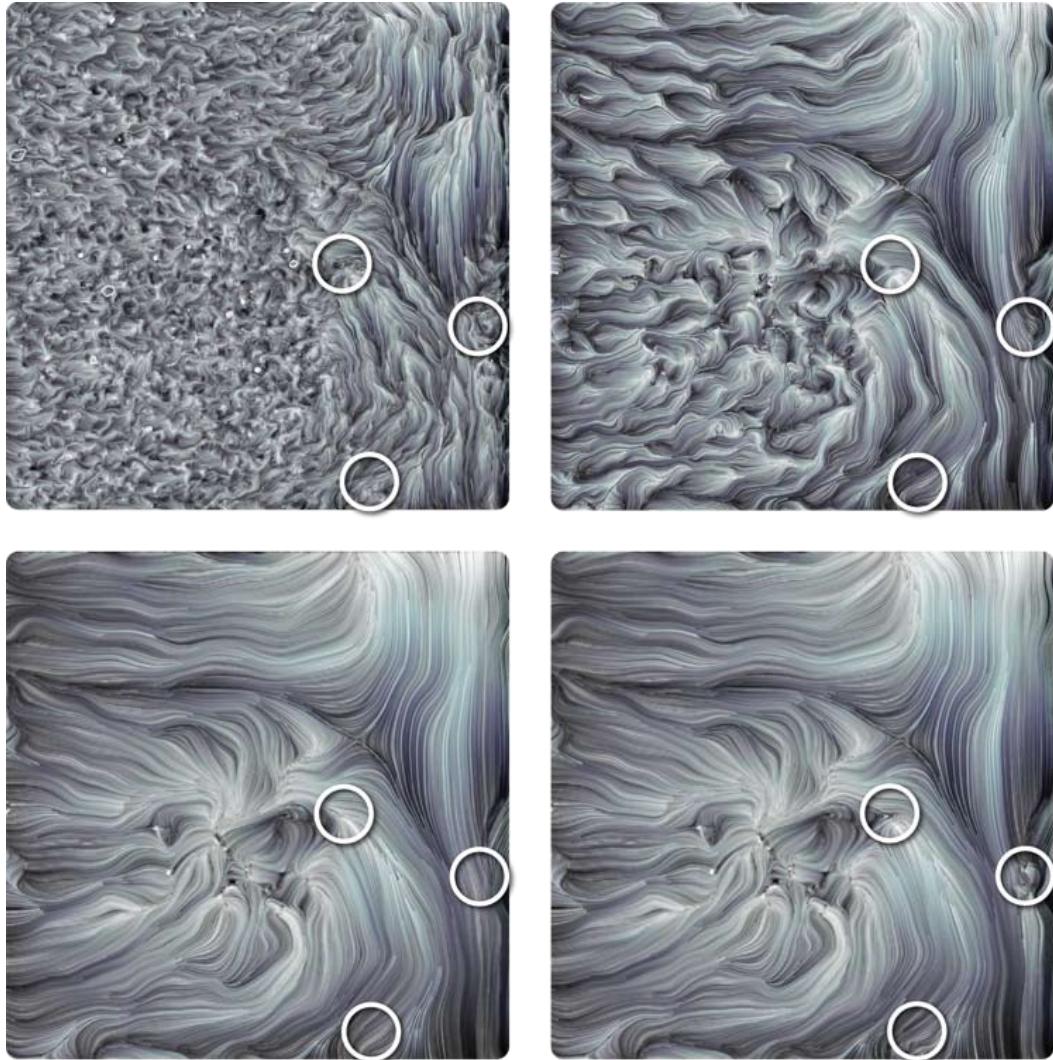


Figure 4.8: Denoising a vector field (top left) measured by a PIV device on a wall-jet experiment: the scale-space at steps 30 and 100 (middle images) smoothens out the important singularities, at the right part of the image, while keeping some singularities related to the noise at the left part of the image. Selecting singularities at the right of the image better recovers the behavior of the fluid (bottom right).

5 Conclusion

Random walk filter. This dissertation first proposes a vector field denoising technique based on random walks, whose main characteristic is the preservation of coherent discontinuities while removing noise under the vanishing-mean *per continuous* region model. Initially, the vectors are updated through weighted averaging of the neighboring vectors, with the weights being determined by probabilities of random walk steps between each face and its neighbors. These probabilities in turn depend on the differences in a feature such as vector magnitude and direction or a combination of the two. To do so, a suitable similarity function with weighted additive distance for the point/vector pair was proposed.

We then show several applications of the method to PIV images, SPH and granular flows simulation data. These experiments presented a visual comparison between our approach and gaussian filtering. In the majority of the results, the random walk filter method outperforms the gaussian filtering by preserving the field's main features. In the PIV data set, the random walk filter was unable to deal with the fact that the information present was not at a constant scale compared to the noise. To address this problem we developed the topology-aware denoising methodology in Chapter 4.

Topology-aware denoising. To circumvent the problem mentioned above, we proposed a methodology to denoise vector fields taking advantage of the user knowledge of the data. With the singularity detection detailed in Chapter 2, our interface displays topological changes throughout the scale space generated by successive filters to give the user global information of the field. As a consequence, the user can easily adapt the local filtering scale in order to preserve important information while aggressively removing noise. The result is generated by smoothly interpolating the field. The method supports different techniques for singularity detection, scale function interpolation, and scale-space generation.

Discussion and future work

Random walk filter. While our algorithm is effective for feature-preserving vector field denoising, it is still has certain problems that other filters also possess. The tuning of parameters is a weak point of the random walk filter, where it can have from 2 to 4 parameters to adjust when filtering. Even though suggestions were made in this dissertation for each parameter, this is clearly data-dependent. One example is that we have to interactively determine the number iterations. Using a small number of iterations fails to fully remove the noise the vector field, while too many causes too much smoothing. Future work is needed to find an automatic method of determining the optimal number of iterations.

Also when trying to preserve a certain feature such as a discontinuity another problem arises; the random walk filter might destroy or displace another feature such as a singularity. We address a part of this problem with the topology-aware methodology by giving the user direct control.

Topology-aware denoising. The methodology proposed has a few shortcomings with the topology detections. First of all the current detection of singularities is only done locally and the reconstruction is done on a local base. Therefore, it does not handle non-local singularities such as a closed orbit. Also the methodology does not include the tracking of the singularities and only displays topological changes, which would have improve the robustness of the method and the visualization for the user.

Unlike the random walk filter, the methodology only works on a structured grid, while many recent vector field datasets are unstructured to better take into account errors in the measure-point localization. In the future we plan on implementing the techniques for singularity detection and classification to unstructured vector fields with a interpolation such as the SPH interpolation (22).

Finally, as mentioned above, large-scale denoising may displace the location of the singularity and it is very problematic since in this case our interface displays two very close topological changes, which are not relevant. In the future, a study of the behavior of critical points in vector field, at first, under gaussian filtering would be imperative. This will allow us to design filters to minimize the dislocation of the singularities.

Bibliography

- [1] BAUER, D.; PEIKERT, R. Vortex tracking in scale-space. In: PROCEEDINGS OF THE SYMPOSIUM ON DATA VISUALISATION 2002, p. 233. Eurographics Association, 2002.
- [2] BECKER, J. Método de interpolação de shepard baseado em núcleo. 2002. Master's thesis - Department of Mathematics, PUC-Rio. Advised by Hélio Lopes.
- [3] BORDIGNON, A.; SIGAUD, L.; TAVARES, G.; LOPES, H.; LEWINER, T. ; MORGADO, W. Arch generated shear bands in granular systems. **Physica A: Statistical Mechanics and its Applications**, v.388, n.11, p. 2099–2108, 2009.
- [4] BORDIGNON, A.; VATH, B.; VIEIRA, T.; FERREIRA, C.; CRAIZER, M. ; LEWINER, T. Scale-space for union of 3d balls. In: SIBGRAPI, p. 9–16. IEEE, 2009.
- [5] COOK, R. L. Stochastic sampling in computer graphics. **Transactions on Graphics**, v.5, n.1, p. 51–72, 1986.
- [6] EIBL, G.; BRANDLE, N. Evaluation of clustering methods for finding dominant optical flow fields in crowded scenes. **International Conference on Pattern Recognition**, p. 1–4, 2008.
- [7] GINGOLD, R.; MONAGHAN, J. Smoothed particle hydrodynamics-theory and application to non-spherical stars. **Monthly Notices of the Royal Astronomical Society**, v.181, p. 375–389, 1977.
- [8] GRADY, L. Random walks for image segmentation. **Transactions on Pattern Analysis and Machine Intelligence**, v.28, n.11, p. 1768–1783, 2006.
- [9] GROBMAN, D. **Homeomorphisms of systems of differential equations**. In: DOKL. AKAD. NAUK SSSR, volume 128, p. 880–881, 1959.

- [10] HELLER, V.; HAGER, W. ; MINOR, H. Scale effects in subaerial landslide generated impulse waves. **Experiments in Fluids**, v.44, n.5, p. 691–703, 2008.
- [11] HELLER, V.; HAGER, W. ; MINOR, H. Scale effects in subaerial landslide generated impulse waves. **Experiments in Fluids**, v.44, n.5, p. 691–703, 2008.
- [12] JU, T.; ZHOU, Q.-Y. ; HU, S.-M. **Editing the topology of 3d models by sketching**. In: SIGGRAPH, volume 26, p. 42, 2007.
- [13] KLEIN, T.; ERTL, T. Scale-space tracking of critical points in 3D vector fields. **Topology-based Methods in Visualization**, p. 35–49, 2007.
- [14] LAI, Y.-K.; HU, S.-M.; MARTIN, R. R. ; ROSIN, P. L. **Fast mesh segmentation using random walks**. In: SOLID AND PHYSICAL MODELING, p. 183–191. ACM, 2008.
- [15] LEWINER, T.; FERREIRA, C.; CRAIZER, M. ; TEIXEIRA, R. **Curvature motion for union of balls**. In: SIBGRAPI, p. 47–54. IEEE, 2005.
- [16] LOVASZ, L. **Random walks on graphs: A survey**. **Bolyai Society Mathematical Studies**, v.2, n.1, p. 1–46, 1993.
- [17] LUKAC, R.; PLATANIOTIS, K. N.; SMOLKA, B. ; VENETSANOPOULOS, A. N. Generalized selection weighted vector filters. **Applied Signal Processing**, v.2004, p. 1870–1885, 2004.
- [18] NASCIMENTO, R.; PAIXÃO, J.; LOPES, H. ; LEWINER, T. **Topology aware vector field denoising**. In: SIBGRAPI. IEEE, 2010.
- [19] NEHAB, D.; RUSINKIEWICZ, S.; DAVIS, J. ; RAMAMOORTHI, R. Efficiently combining positions and normals for precise 3d geometry. **Transactions on Graphics**, v.24, n.3, p. 536–543, 2005.
- [20] NORRIS, J. **Markov chains**. Cambridge University Press, 1998.
- [21] PAIXÃO, J.; LAGE, M.; PETRONETTO, F.; BORDIGNON, A.; PESCO, S.; TAVARES, G.; LEWINER, T. ; LOPES, H. **Random walks for vector field denoising**. In: SIBGRAPI, p. 112–119. IEEE, 2009.
- [22] PETRONETTO, F.; PAIVA, A.; LAGE, M.; TAVARES, G.; LOPES, H. ; LEWINER, T. **Meshless Helmholtz-Hodge Decomposition**. **Visualization**

- and Computer Graphics, IEEE Transactions on, v.16, n.2, p. 338–349, 2010.
- [23] PLATANIOTIS, K. N.; VENETSANOPoulos, A. N. **Color image processing and applications**. Springer, 2000.
- [24] RAFFEL, M.; WILLERT, C. ; KOMPENHANS, J. **Particle Image Velocimetry: A Practical Guide**. Springer, 2002.
- [25] SHARF, A.; LEWINER, T.; SHKLARSKI, G.; TOLEDO, S. ; COHEN-OR, D. **Interactive topology-aware surface reconstruction**. In: SIGGRAPH, volume 26, p. 43.1–43.9. ACM, 2007.
- [26] SHEPARD, D. **A two-dimensional interpolation function for irregularly-spaced data**. In: ACM NATIONAL CONFERENCE, p. 517–524, 1968.
- [27] SMOLKA, B.; LUKAC, R.; CHYDZINSKI, A.; PLATANIOTIS, K. N. ; WOJCIECHOWSKI, W. **Fast adaptive similarity based impulsive noise reduction filter**. **Real-Time Imaging**, v.9, n.4, p. 261–276, 2003.
- [28] SMOLKA, B.; WOJCIECHOWSKI, K. W. **Random walk approach to image enhancement**. **Signal Processing**, v.81, n.3, p. 465–482, 2001.
- [29] STANISLAS, M.; KOMPENHANS, J. ; WESTERWEEL, J. **Particle Image Velocimetry - Progress Towards Industrial Application**. Springer, 2000.
- [30] SUN, X.; ROSIN, P. L.; MARTIN, R. R. ; LANGBEIN, F. C. **Random walks for mesh denoising**. In: SOLID AND PHYSICAL MODELING, p. 11–22. ACM, 2007.
- [31] SUN, X.; ROSIN, P. L.; MARTIN, R. R. ; LANGBEIN, F. C. **Random walks for feature-preserving mesh denoising**. **Computer Aided Geometric Design**, v.25, n.7, p. 437–456, 2008.
- [32] SZCZEPANSKI, M.; SMOLKA, B.; PLATANIOTIS, K. N. ; VENETSANOPoulos, A. N. **On the geodesic paths approach to color image filtering**. **Signal Processing**, v.83, n.6, p. 1309–1342, 2003.
- [33] TAUBIN, G. **A signal processing approach to fair surface design**. In: SIGGRAPH, p. 351–358. ACM, 1995.
- [34] TOMASI, C.; MANDUCHI, R. **Bilateral filtering for gray and color images**. In: ICCV, p. 839–846, 1998.

- [35] TRICOCHE, X. **Vector and Tensor Field Topology Simplification, Tracking, and Visualization.** 2002. 107–116p. PhD thesis - University of Kaiserslautern. Advised by G. Scheuermann, H. Hagen and S. Clauss.
- [36] WECHSLER, H.; KIDODE, M. A random walk procedure for texture discrimination. **Transactions on Pattern Analysis and Machine Intelligence**, v.1, n.3, p. 272–280, 1979.
- [37] WESTENBERG, M. A.; ERTL, T. Denoising 2-d vector fields by vector wavelet thresholding. **Journal of WSCG**, v.13, n.1-3, p. 33–40, 2005.