

HABILITATION A DIRIGER DES RECHERCHES

Présentée par

Christophe Hurter

Discipline : Informatique

Spécialité : Interaction Homme-Machine

Toward image based algorithm to support interactive data exploration

2014

Rapporteurs:

Jack van Wijk, Professor in visualization at the Department of Mathematics and Computer Science of Eindhoven University of Technology (TU/e)

Niklas Elmqvist, Associate Professor, College of Information Studies Affiliate Associate Professor, Department of Computer Science Faculty, University of Maryland Institute for Advanced Computer Studies University of Maryland, College Park, MD, USA

Guy Melançon, Professor at Université de Bordeaux France, affiliated with CNRS UMR 5800 LaBRI

Jury:

Sheelagh Carpendale, Professor in the Department of Computer Science at the University of Calgary., Canada

Michel Beaudouin-Lafon, Professor in computer science at Paris-Sud Université France

Stéphane Conversy, Associate Professor at ENAC, french civil aviation Université, France

Jean-Pierre Jessel, Professor at Paul Sabatier Université - Toulouse III, Institut de Recherche en Informatique de Toulouse, France

Habilitation à Diriger des Recherches, préparé au sein de l'équipe ENAC-LII

Abstract

Our society has entered a data-driven era, in which not only enormous amounts of data are being generated every day, but there are also growing expectations placed on the analysis of these data. OpenData programs, in which data are available for free, are growing in number. Analyzing these massive and complex datasets is essential to making new discoveries and creating benefits for people, but it remains a very difficult task. In many cases, the ability to make timely decisions based on available data is crucial to business success, clinical treatments, cyber and national security, and disaster management. As such, most data have become simply too large and often have a too short lifespan, i.e. it changes too rapidly for classical visualization or analysis methods to be able to handle it properly. One potential solution is not only to visualize data, but also to allow users to be able to interact with them.

Therefore my research activities have essentially focused on two main topics: large dataset visualization and interaction design. During my investigations, I tried to take advantage of graphic card power with techniques called GPGPU. Since data storage and memory limitation is less and less of an issue, I tried to reduce computation time by using memory as a new tool to solve computationally challenging problems. I have tested this approach to improve brushing techniques, animation between different data representations, to compute static and dynamic bundling and to produce fast enough visualization to be interactive.

During my research I also investigated innovative data processing: while classical algorithms are expressed in the data space (e.g. computation on geographic locations), I developed algorithms expressed in the graphic space (e.g. raster map like a screen composed of pixels). This consists of two steps: first, a data representation is built using straightforward InfoVis techniques; second, the resulting image undergoes purely graphical transformations using image processing techniques. This type a technique is called *image based algorithm*.

My goal was to explore new computing techniques with image based algorithm to provide efficient visualizations and user interfaces for the exploration of large datasets. My project themes belong to the areas of Information Visualization, Visual Analytics, Computer Graphics and Human Computer Interaction. This opens a whole field of study, including the scientific validation of the method, its limitations, and its generalization to different types of datasets, other algorithms, and other time-dependent representation patterns.

Keywords: Human-Computer Interaction, Interaction Techniques, Visualization Techniques, Information Visualization

Table of Content

1 Introduction 7

1.1 Image-based assets 8
1.2 Image-based algorithm opportunities 9
1.3 Structure of the presented document 9
1.4 Timeline of projects and student advisory 10

2 From visualization characterization to data exploration 13

2.1 Evaluation of visualizations 13
2.2 Application domain 14
2.2.1 Instance of design evaluation: the radar comet 14
2.3 The Card and Mackinlay model improvements 16
2.4 Characterization or data exploration tool 17
2.5 FromDaDy: from data to Display 18
2.6 Conclusion 20

3 Data Exploration with data density maps 22

3.1 Kernel Density Estimation: an image based technique 22
3.2 Density map visualization and interaction techniques 24
3.2.1 Brushing Technique with density maps 24
1.1.1 Interactive lighting direction 25
3.2.2 Density maps as data sources 26
3.3 Application domains 27
3.3.1 Exploration of aircraft proximity 27
3.3.2 Patterns detection in dense datasets 28
3.3.3 Density flaw detection in a dense dataset 28
3.3.4 Exploration of gaze recording 29
3.4 Conclusion 31

4 Edge bundling 32

4.1 MoleView 33
4.2 SBEB: Skeleton-based edge bundling 33
4.3 KDEEB: Kernel Density Edge Bundling 33
4.4 Dynamic KDEEB 35
4.5 3D DKEEB 36
4.6 Directional KDEEB 36
4.7 Conclusions 38

5 Animation as an efficient data exploration tool 39

5.1 From the Mole View to Color Tunneling: the animation as an data exploration tool 40
5.2 GPGPU usages to address scalability issues 40
5.2.1 GP/GPU technique and history 40
5.2.2 Instances of GPU usages 41
5.3 Color Tunneling: a scalable solution to large dataset manipulation with image based interaction 43
5.4 Conclusions 44

6 Strip'TIC: Striping for tangible interface for controllers 45
6.1 Strip'TIC and image based techniques 47
6.2 Conclusion 48
7 Research program 49
7.1 Computed graphic and raster data 49
7.2 Raster data inaccuracy 50
7.3 Technical challenges 51
7.4 Personal image based road map 52
7.4.1 Tasks 53
7.4.2 Dynamic graphs 54
7.4.3 Algorithm setting 54
7.4.4 Bundling faithfulness and accuracy 54
7.5 Proposal to improve bundling techniques 55
7.5.1 Particle system 58
7.6 Image based algorithm in application domains 58
7.6.1 Cognitive maps 59
7.6.2 Eye tracking 60
7.6.3 Image processing: skin cancer investigation 61
7.6.4 Point cloud display 62
7.6.5 Movement data analysis 63
7.1 Conclusion 65
8 Bibliography 67
9 Selected research papers 74

Table of figures

County-to-county migration flow files (http://www.census.gov/population/www/cen2000/ctytoctyflow/, the Census 2000): people who moved between counties within 5 years. Original data only shows the outline of the USA (left), bundled and shaded path (right) shows multiple information like East-West and north-South paths, [Hurter et al 2012]. 8
Timeline of my academic activities 12
The design of the radar comet used by Air Traffic Controllers. 14
the radar design with old radar technology (left), a modern radar screen (right). 15
Halley drawing (1686) of the trade wind. 15
The visual caracterization of the radar screen for air traffic controlers. 17
DataScreenBinder and the visualization of aircraft speeds in a circular layout. 18
One day of recraded aicraft trajectory over France. 19
Union Boolean operation 19
FromDaDy with two layouts and its animation 20
GPU implementation of the brushing technique 20
day aircraft trajectory over France (left), 3D density map (right). 22
Kernel profiles 23
Kernel Density Estimation convolving principle with a raster map. 23
Brushing over a density map. 24
Points or pick and drop trajectories on an density map. 25
Interactive light manipulation to emphasise ridges and furrows on a shaded density map. 26
Matrix view with standard color blending (left) and customized visual mapping with the size (right). 26
KDE maps value displayed with the size visual variable to display a matrix 27
Density map of the safety-distance alarms triggered over France over a one year period. Red colored areas correspond to dense alarm areas where aircraft triggered proximity alerts 27
Design configuration and accumulation maps without shading. 28
Time series of the incremental number of aircraft over time (left) and the corresponding density map (right). 29
Density map computation with gaze recording. 30
Accumulation view and its configuration to produce a per trajectory distance accumulation (*a*,*b*). Comparison between trail visualization with or without the accumulation (*c,d*) 30
First mole view prototype with a semantic lens 32
Exploration of the original version and the bundle version of a graph (Hurter et al., 2011b) 33
Skeleton computation. A set of edges (a) is inflated (b), then the distance tansform is computed (c) and finally the skeleton is extracted (d). 33
First bundling prototype based on density computation 34
US migration graph. Original (top), bundled (bottom), with shaded density (bottom right). 35
Bundling of dependency graph with obstacle ovoidance (right). 35
Eye gaze recording of a pilot when landing with a flight simulator. Bundled trails (right) with KDEEB. 36
3D DKEEB, top view. 36
Investigation of aircraft trails with a directional bundling algorithm. 37

Paris area with the KDEB algorithm. 38
Point based rendering of a 3D ball with pseudo color and a semantic 3D lense..... 39
Visualization of a 3D scan with point based rendering and color transfer function. 39
First prototype of an animation between an image and its corresponding histroram..... 40
Color Tunneling (Hurter et al., 2014b), finding intensity outliers with isolated ranges in an astronomical data cube (Taylor et al., 2003)..... 43
A) Cycling data: a map with a route and area graphs with average altitude (purple) and heart-rate (orange) at each kilometer. B) The rectified route map aligned under the linear graphs enables comparison of the measured variables to the map features. C) The heart-rate graph wrapped around the route in the map shows effort in spatial context. D) Same as C, but with multiple variables. Map used contains Ordnance Survey data c Crown..... 44
very first paper based prototype. Radar screen with and without the Anoto dot layer (right). 46
Strip tracking system with bottom and top projection. 46
Radar screen with and without the Anoto dot layer..... 47
Raster map size effects. Original graph (left), bundled version with a small raster map (middle) and with a large raster map (right). 51
small size density map (left), large size density map (right). 51
bundling, original US migration grap (top), KDEEB bundled version (middle), with bump mapping (bottom). 53
interactive tools to explore the dual layout of a graph(bundled and unbundled) 53
Graph pseudorandom and bundle version KDEEB (C. Hurter et al., 2012)..... 55
: examples of bundling results with very different visual renderings on the same datasets..... 57
directional graph visualization with a particle system. Particles can overlap but one can still visualize their direction..... 58
Visualization of cited cities over time. We notice a significant simplification of the network for demented subjects. 60
Simulation environment with head mounted eye tracking system (left). Visualization of fixation points inside the cockpit when gathering flight parameters in red, and elsewhere in blue. 60
gaze analysis with bundling technique [hurter et al. 2012] [hurter et al. TVCG 2014]. Color corresponds to the gaze direction. Trail width corresponds to the density of the path. 61
Skin tumor investigation with Color tunneling (Hurter et al., 2014b). One can navigate between data configuration and brush pixel to select a skin-mole frontier. 62
3D point scan (left), with edge filtering (right, MoleView (Hurter et al., 2011b)). 63
small multiples of US air lines over one week..... 63

1 Chapter
Introduction

Our society has entered a data-driven era, in which not only enormous amounts of data are being generated every day, but also growing expectations are placed on their analysis (Thomas and Cook, 2005). With the support of companies like Google[1], big data has become a fast emerging technology. OpenData programs, in which data are available for free, are growing in number. A number of popular web sites, instead of protecting their data against "scripting", have opened access to their data through web services in exchange for pecuniary retribution (eg. SNCF[2], France's national railway company, IMDb movie data base[3]). Taking advantage of this, new activities are emerging such as data journalism that consists in extracting interesting information from available data and presenting it to the public in a striking fashion. Analyzing these massive and complex datasets is essential to make new discoveries and create benefits for people, but it remains a very difficult task. Most data have become simply too large to be displayed and even the number of available pixels on a screen are not sufficient to carry every piece information (Fekete and Plaisant, 2002). These data can also have too short a lifespan, i.e. they change too rapidly, for classical visualization or analysis methods to handle them properly.

These statements are especially true with time dependent data: these data are by their intrinsic nature larger than static data, and their analysis must be performed in a constrained time frame: the data validity time. Movement data, which are multidimensional time-dependent data, describe changes in the spatial positions of discrete mobile objects. Automatically collected movement data (e.g. GPS, RFID, radars, and others) are semantically poor as they basically consist of object identifiers, coordinates in space, and time stamps. Despite this, valuable information about the objects and their movement behavior as well as about the space and time in which they move can be gained from movement data by means of analysis. Analyzing and understanding time-dependent data poses additional non-trivial challenges to information visualization. First, such datasets are by their very nature several orders of magnitude larger than static datasets, which underlines the importance of relying on efficient interactions with multiple objects and fast algorithms. Secondly, while patterns of interest in static data can be naturally depicted by specific representations in still visualizations, we do not yet know how to best visualize dynamic patterns, which are inherent to time-dependent data. While there are many solutions for displaying patterns of interest in static data with still visualizations, little work has addressed the issue of dynamic patterns (von Landesberger et al., 2011).

During my research work, I have tried to address the two following scientific challenges. The first concerns large data representation: how can these datasets be represented and how can this be done in an efficient manner? Second challenge addresses data manipulation: how can we interact effectively with them and how can this be done in a way which fosters discovery.

[1] http://www.google.fr
[2] http://data.sncf.com/
[3] http://www.imdb.com/

When dealing with large data, both interaction and representation heavily rely on algorithms: algorithms to compute and display the representation, and algorithms to transform the manipulation by the user into updates of the view and the data. Not only does the performance of these algorithms determine what representations can be used in practice, but their nature also has a strong influence on what the visualizations look like.

The classical usage of algorithm in InfoVis is expressed in the data space (e.g. computation on geographic locations). In my research projects, I have investigated an alternative approach: algorithms expressed in the graphic space (*image-based algorithms*). This consists of two steps:

- First, a data representation is constructed using straightforward InfoVis techniques.
- Secondly, the resulting image undergoes purely graphical transformations using image processing techniques. Furthermore, rather than only modifying the data-to-image mapping, user manipulations also modify the image processing. For instance, users manipulate the lighting of the scene to reveal interesting data.

This approach, called *image-based InfoVis*, differs from most other Infovis works in that it not only uses pixel-based visualization techniques, but it also performs data exploration using image-based algorithms. I aim to explore a domain that is not just classical InfoVis because it relies on Computer Graphics, and not just Computer Graphics either because it still focuses on interaction rather than just the creation of graphics. My goal has been to explore new computing techniques to provide efficient visualizations and user interfaces for the exploration of large datasets. My research project is at the crossroads of Information Visualization, Visual Analytics, Computer Graphics and Human Computer Interaction.

As a first example, I investigated the mean shift algorithm (Comaniciu and Meer, 2002), a clustering computer graphic algorithm, and developed Kernel Density Estimation Edge Bundling, KDEEB (C. Hurter et al., 2012) a new image-based bundling algorithm (Figure 1).

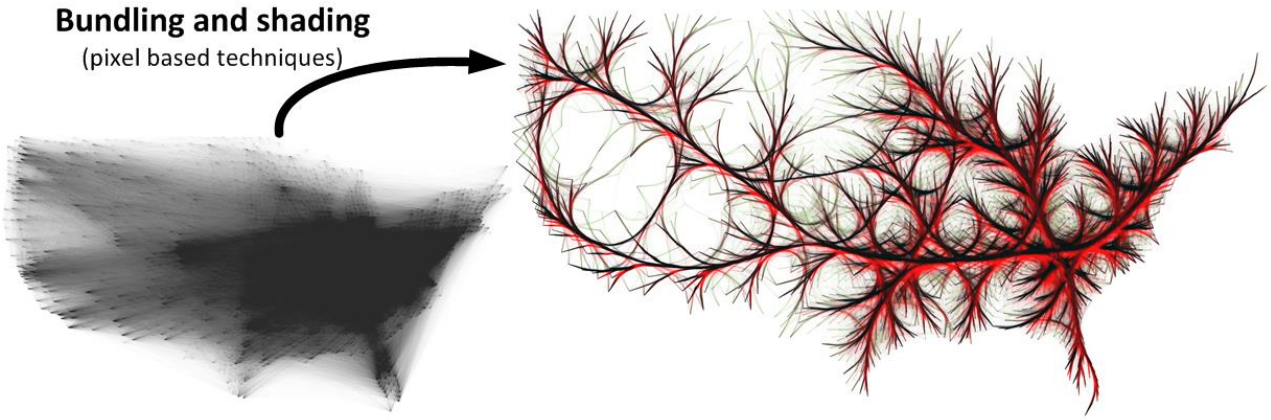


Figure 1: County-to-county migration flow files (http://www.census.gov/population/www/cen2000/ctytoctyflow/, the Census 2000): people who moved between counties within 5 years. Original data only shows the outline of the USA (left), bundled and shaded path (right) shows multiple information like East-West and north-South paths, [Hurter et al 2012].

1.1 Image-based assets

The Image-based approach takes advantage of changes in the bottlenecks of computer graphics: since data storage and memory limitation is becoming less and less of an issue (Sutherland, 2012), we can plan to reduce computation time by using memory as a new tool to solve computationally challenging

problems. Furthermore, even if graphic cards were initially developed to produce 2D/3D views close to photo-realistic images, their power has also been used to perform parallel computations (so called GPGPU techniques) (Owens et al., 2007).

I have recently tested this approach to compute representations based on static and dynamic bundling of transport flows (Figure 1) and it proved to be a most efficient way of producing interactive representations (Hurter et al., 2013b). This opens a whole field of study, including the scientific validation of the method, its limitations, and its generalization to different types of datasets, other algorithms, and other time-dependent representation patterns.

Conceptually, this approach takes its roots in the following pioneering works:

- texturing as a fundamental primitive (Cohen et al., 1993),
- cushion Treemaps (Van Wijk and van de Wetering, 1999),
- dense-pixel visualizations (Fekete and Plaisant, 2002) which use every available pixel of an image to carry information.

These works are consistent with how most laboratories, including the Interactive Computing Laboratory at ENAC, have approached InfoVis so far: as a branch of HCI that aims to exploit all human ability to absorb or find information, including through interactive representations.

1.2 Image-based algorithm opportunities

Based on my research results, I discovered three potential benefits of the pixel-based algorithm:

First, pixel-based algorithms can greatly benefit from the use of graphic cards and their massive memory and parallel computation power. They are highly scalable (each pixel can be used to display information) and graphic cards can easily handle a large quantity of them. In addition, classical image processing techniques such as sampling and filtering can be used to construct continuous multiscale representations, which further helps scalability.

Secondly, image-processing field offers many efficient algorithms that are worth applying to image-based information visualization. By synthesizing color, shading, and texture at a pixel level, we achieve a much higher freedom in constructing a wide variety of representation that is able to depict the rich data patterns we aim to analyze.

Thirdly, I am strongly convinced that the use of memory instead of computation can reduce algorithm complexity. Under a given set of restrictions, reduced complexity should reduce computation time and thus improve the ability of users to interact with complex representations. Furthermore, reduced complexity should facilitate comprehension by programmers, and thus foster maintainability, dissemination and reuse by third parties.

1.3 Structure of the presented document

In this document, I have summarized and structured my work during my PhD and my position as an assistant professor. In addition to the outlines of key papers, I will also give details that have not been published. This document is an ideal occasion to provide the reasoning behind published work, extensions and ideas which did not find a concrete form.

This document is chronologically ordered with a list of works that started from the characterization of visualization and that led me to investigate image based algorithms to develop visualization and interaction techniques.

1.4 Timeline of projects and student advisory

As an assistant professor I had the opportunity to advice, supervise and work with many PhD students. Figure 2 summarizes my academic activities with PhD students, my academic positions and my projects. The following list gives details regarding my involvement with these PhD students and the projects conducted:

Maxime Cordeil (PhD advisor: Stéphane Conversy): After my PhD, many research questions regarding multidimensional dataset exploration warranted further investigation. Upon the advises of Stéphane Conversy, my former PhD advisor, we decided to investigate in more detail how animations and smooth transitions could improve the data exploration process. As a co-advisor, I defined the research project and I advised Maxime Cordeil to investigate this topic. We investigated the design of visual and animated transitions and proposed a taxonomy of animated transitions. With this taxonomy, we studied the features of 3D animated transitions and proposed a set of new interactions to control animated transitions in data visualizations. With regards to visual transitions, we analyzed the visual path of air traffic controllers and designed animated transitions which improve the search and retrieval of information amongst different visualizations. Maxime defended his PhD in 2013 and we published several research papers (Cordeil et al., 2013, 2011a, 2011b; Savery et al., 2013).

Jean-Paul Imbert (PhD advisor: Frédéric Dehais): As a co advisor, I supervised Jean-Paul Imbert during his PhD. He investigated how situation awareness can be improved and monitored to support supervision activities like air traffic control. During his PhD, my role was to guide him to fulfill academic and scientific requirements.

Gwenael Bothorel (PhD advisor: Jean-Marc Alliot): During his PhD, Gwenael Bothorel investigated the visualization of frequent itemsets and association rules. I helped him with my knowledge regarding large dataset exploration. I advised him to develop a visual analytics version of FromDaDy (Hurter et al., 2009b), a multi-dimensional exploration tool developed during my PhD. I also provided him with edge bundling algorithms (C. Hurter et al., 2012) and data density computation (Hurter et al., 2010b). We published several research papers (Bothorel et al., 2013a, 2013b, 2013c, 2011).

Ozan Ersoy (PhD advisor: Alexandru Telea): During his PhD, Ozan Ersoy investigated image-based graph visualization and we worked together in a fruitful collaboration with his PhD advisor Alexandru Telea. Together, we developed new edge Bundling algorithms and interactive systems to support data exploration. We published several research papers (Ersoy et al., 2011; Hurter et al., 2011b; C. Hurter et al., 2012; Hurter et al., 2013a, 2013b).

Cheryl Savery (PhD advisor: Nick Graham): During the LEIF exchange program between Canada and France (http://www.leif-exchange.org), I had the chance to supervise Cheryl Savery. She worked on the extension of Strip'Tic (Christophe Hurter et al., 2012), an augmented paper based system to support air traffic controller activity. During her internship, we worked together to improve the system using multi touch capability and we conducted one design study. We published one research paper on this topic (Savery et al., 2013).

Sarah Maggi (PhD advisor: Sara Fabrikant): As a member of her PhD program committee, I had the opportunity to work with Sarah Maggi. We investigated with Maxime Cordeil specific designs used by air traffic controllers (the radar comet). We defined and conducted experimentations to assess how animations of these designs can carry perceivable information.

Aude Marzuoli (PhD advisor: Eric Feron): Aude Marzuoli started her Phd in 2010 at the Georgia Institute of Technology, Atlanta, USA. While visiting ENAC, she investigated multi-source dataset exploration to support en-route air traffic flow management optimization. I introduced her to new data visualization toosl and advised her on existing multidimensional data exploration methods and specifically on how to use my current data exploration tools and graph simplification methods. We published a research paper modeling aircraft trajectories into flows to support the analysis of airspace complexity (Marzuoli et al., 2012). She is currently looking into using the same tools to estimate airspace efficiency.

Vsevolod Peysakhovich (PhD advisor: Frédéric Dehais): During his PhD which started in October 2013, Vsevolod Peysakhovich has been investigating new metrics to assess users' behavior thanks to pupil and gaze recorded data. I co-advise him and we are working together to improve and to apply edge bundling algorithms to eye tracker recorded data.

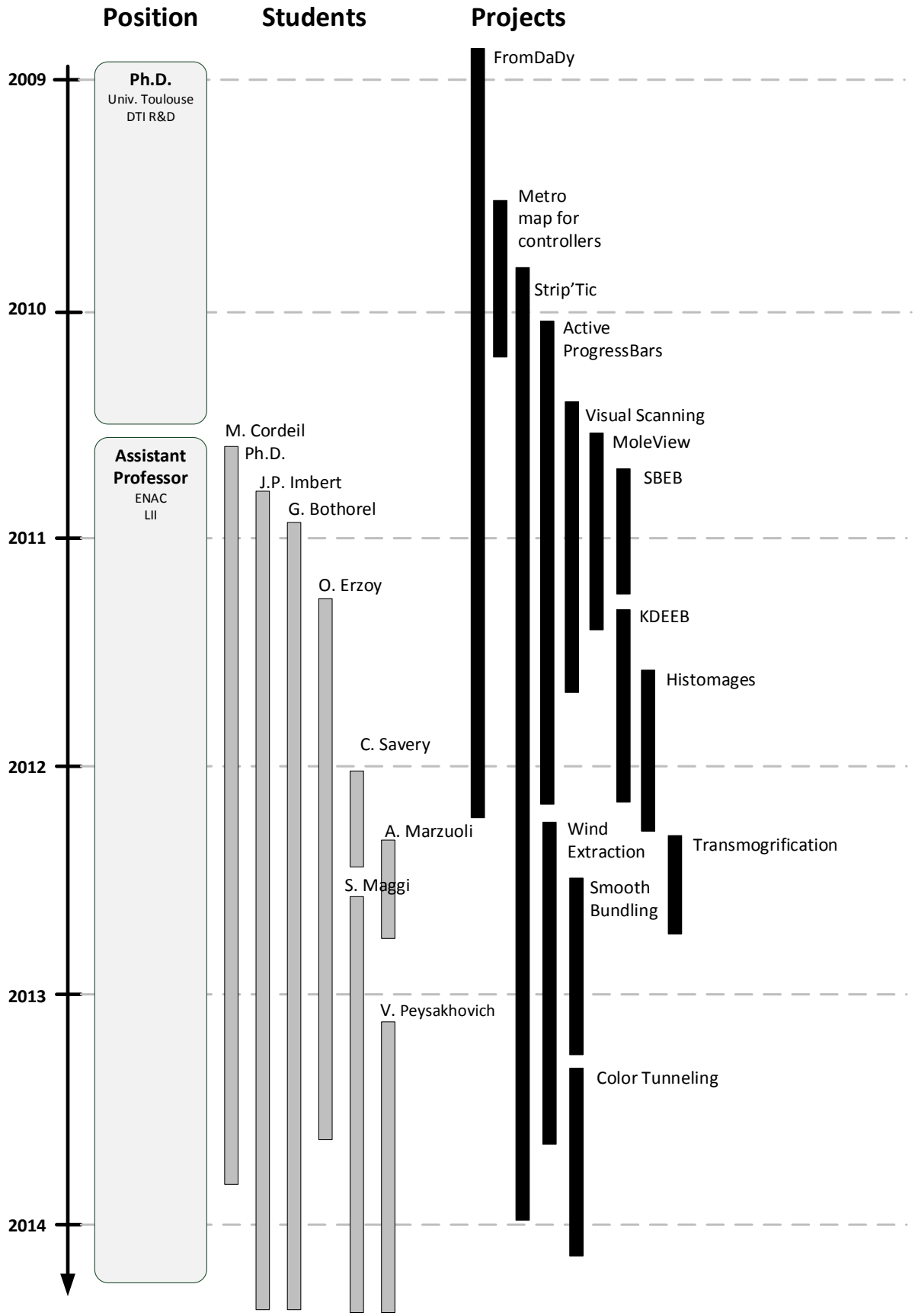


Figure 2: Timeline of my academic activities

2 Chapter
From visualization characterization to data exploration

In this section, I will give additional details concerning my PhD (Hurter, 2010) and I will outline my first 2 PhD years where I focused on visual design issues. From this topic, I evolved toward large data exploration. Even if this change of research direction seems drastic, it is a natural evolution based on the data transformation pipeline (Card et al., 1999) which both visualization characterization and data exploration tool share. This initial work helped me to structure my reasoning, and help me discover GPU usages.

2.1 Evaluation of visualizations

The evaluation of visualizations is a long and difficult process which is often based on the completion time and error measurement to perform a task. Since users are involved in the evaluation process, this method is time costly and requires numerous users to yield reliable results. Some methods exist to assess visualizations before user tests but they only concern the effectiveness of interaction. These methods rely on models of the system and they have proved to be accurate and efficient when designing new interfaces. For example, KeyStroke (Card et al., 1983) and CIS (Appert et al., 2005) are *predictive* models that help compute a measurement of expected effectiveness, and enable quantitative comparison between interaction techniques. If methods to assess interactive systems do exist, very few can assess visualization before user tests.

During the first part of my PhD, I tried to go beyond *time* and *error* evaluation and propose an assessment of the *bandwidth* of available information in a visualization. Therefore, I focused on analyzing visualizations to extract relevant characterization dimensions. My goal was to perform an accurate visualization evaluation and to answer these questions: “What is the visible information?”, “What are the phenomena/mechanisms that make them visible?”. To characterize visualization, I then faced the following issues:

- How could the relevant characterization dimensions for the description be found (the content of the description)?
- How could an accurate and exhaustive description of a visualization be formatted?
- How could they be represented to enable comparisons (the representation of the description)?

Previous works use the data transformation pipeline to find relevant characterization dimensions of a visualization (Card and Mackinlay, 1996). This pipeline model uses raw data as an input and transforms them with a transformation function to produce visual entities as an output. Thus, the characterization of visualization consists of describing the Transformation Function. However this method is not sufficient to fully describe visualization, especially with a specific class of design that uses emerging information (Hurter and Conversy, 2008). Basically, emerging information is perceived by users without being transformed by the pipeline model functions.

The first step of this work was to gather enough examples of ATC visualization to cover the largest design space. Then I proposed to apply available characterization models, to assess if they were suitable for the activity to be supported and if the need arose, to improve them. This characterization

had to be done with objective and formal assessments. The designer should have been able to use this characterization to list the available information, to compare the differences between views, to understand them, and to communicate with accurate statements (Hurter, 2010).

2.2 Application domain

In order to benefit from concrete cases, we used the Air Traffic Control (ATC) application domain. ATC activities employ two kinds of visualization systems: real-time traffic views, which are used by Air Traffic Controllers (ATCos) to monitor aircraft positions, and data analysis systems, used by experts to analyze past traffic recording (e.g. conflict analysis or traffic workload). Both types of system employ complex and dynamic visualizations, displaying hundreds of data items that must be understandable with the minimum of cognitive workload.

As traffic increases together with safety concerns, ATC systems need to display more data items with at least the same efficiency as existing visualizations. However the lack of efficient methods to analyze and understand why a particular visualization is effective spoils the design process. Since designers have difficulty analyzing previous systems, they are not able to improve them successfully, or to communicate accurately about design concerns. Visualization analysis can be performed by *characterizing* them. In the InfoVis field, existing characterizing tools are based on the dataflow model (Card et al., 1999) that takes as input raw data and produces visualizations with transformation functions. Even if this model is able to build most of the existing classes of visualization, we show in the following that it is not able to characterize them fully, especially ecological designs that allow emerging information.

2.2.1 Instance of design evaluation: the radar comet

The main task for an ATCo is to maintain a safe distance between aircraft. To be compatible with this task, the process of retrieving and analyzing information must not be cognitively costly. Especially in this field, the precise analysis of visualization is useful to list the visually available information, and to forecast the resulting cognitive workload.

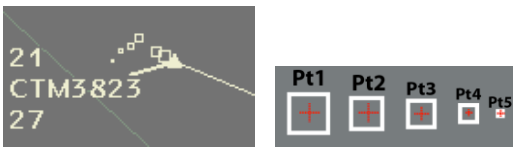


Figure 3: The design of the radar comet used by Air Traffic Controllers.

ODS coded information	Visual code
Aircraft position	Position
ageing of each position	Size
Aircraft speed	Size (comet length)
Aircraft tendency (left, right)	Comet curvature
Aircraft acceleration	Regular/irregular point spacing
Aircraft entity	Gestalt (proximity and size)

Table 1 : information coded with a Radar comet.

As an example, in the radar view, comets display the position of aircraft. The design of the comet is constructed with squares (Figure 3), whose size vary with the proximity in time of the aircraft's position: the biggest square displays the latest position of the aircraft, whereas the smallest square displays the least recent aircraft position. The positions of the aircraft merge through the effect of Gestalt continuity, in which a line emerges with its particular characteristics (curve, regularity of the texture formed by the points, …); thus this design codes a large amount of information (Table 1).

Before describing the comet design fully, it is interesting to understand where the design comes from. In fact, the visual features of the comet were first used in the early 17th century by Edmond Halley (Thrower, 1969) (Figure 5). In this drawing, the comet helps to understand the trade wind direction with a thicker stroke representing the head of the comet.

The radar comet, used by Air Traffic Controllers, has the same properties as the one introduced by Halley, but this design was created with technological considerations in mind. Early radar screens used the phosphorescent screen effect to display the position of aircraft. Between two radar updates, the previous position of an aircraft was still visible, with a lower intensity. Thus, the Radar plot has a longer lifetime than the Radar period (Figure 4). The resulting shape codes the direction of the aircraft, its speed, its acceleration, and its tendency (the aircraft is tending to turn right or left). For instance, Figure 3 displays an aircraft that is turning to the right and it has accelerated (the non-constant spacing indicates the increase in aircraft speed). With technological improvements, remanence disappears, together with the additional information it provides. Designers and users felt the need to keep the remanence effect, and emulated it.



Figure 4: the radar design with old radar technology (left), a modern radar screen (right).

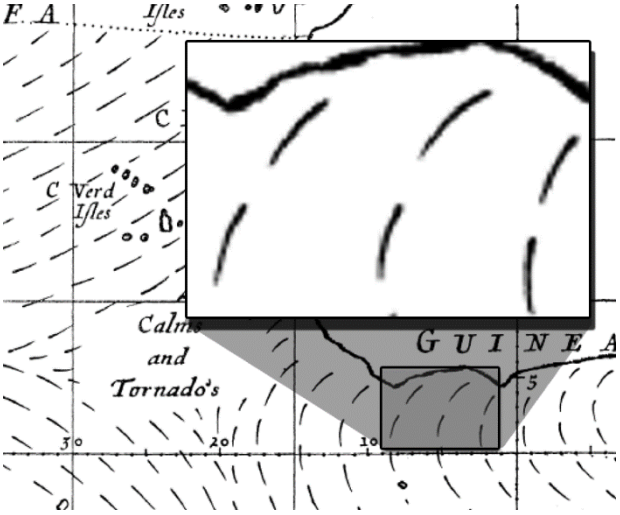


Figure 5: Halley drawing (1686) of the trade wind.

A deeper analysis of the comet design allows us to understand that the user perceives an emerging shape: the regular layout of squares, and the regular decrease in size, configure in a line with the Gestalt effect (Koffa, 1963). Not only does a new visual entity emerge, but its own graphical properties (length and curvature) emerge as well. The graphical properties encode additional information (speed and tendency respectively). Furthermore, this line is able to “resist” comet overlapping; the user can still understand which comet is which, despite tangling.

The design of the comet and its associated information are summarized in Table 2. All emerging information is due to the comet design that uses remanence: several instances of the same object at different times.

Transformation Function	Source of emerging information	Emerging data
Point (latitude, longitude)	comet length	speed
	curvature	tendency
Size(time)	Regular size progression	direction
	square spacing	acceleration

Table 2 : Radar comet characterisation.

It can be noted that this design exhibits a large amount of emerging information. We can thus say that the radar comet is efficient in this respect. This is the reason why this kind of design is widely used in other visualizations (Mnemonic Rendering (Bezerianos et al., 2006), Phosphor (Baudisch et al., 2006)).

2.3 The Card and Mackinlay model improvements

I applied the Card and Mackinlay model (Card and Mackinlay, 1996) to different kinds of ATC visualization (Hurter and Conversy, 2008) (Hurter et al., 2008). I assessed its effectiveness in exhibiting visual properties. When studying the radar comet, the concept of current time was introduced (Tcur: the time when the image is displayed). The size of the square is linearly proportional to current time with respect to its aging. The grey row and column are two additional items from the original C&M model (Table 3).

Name	D	F	D'	X	Y	Z	T	R	-	[]	CP
X	QLon	f	Q Lon	P				Emerging ol			
Y	QLat	f	Q Lat		P						
T	Q	f(Tcur)	Q					S			

Table 3 : C&M Radar Comet characterization

However, the characterization cannot integrate controllers' perception of the aircraft's recent evolution (speed, evolution of speed and direction). For instance, in Figure 3, the shape of the comet indicates that the plane has turned 90° to the right and that it has accelerated (the variation of the dot spacing). These data are important for the ATCo. The comet curvature and the aircraft acceleration cannot be characterized with this model because they constitute emerging information (there is no raw data called 'curvature' to produce a curving comet with the dataflow model). In Table 1, italic script represents emerging information.

Whereas Card and Mackinlay depicted some InfoVis visualizations (Card and Mackinlay, 1996) without explicitly demonstrating how to use their model, we have shown the practical effectiveness of the C&M model when characterizing the radar comet (Hurter and Conversy, 2008). Although the C&M tables make visualization amenable to analysis as well as to comparison, this model does not allow essential information to be highlighted for designers, and does not allow any exhaustive comparison of different designs. We extended this model with the characterization of emerging data. The emerging process stems from the embedded time in the radar plot positions. The time can be easily derived into speed and acceleration. We communicated about this work in a workshop (Hurter et al., 2009a) and

we have extended it with the analysis of the visual scan path the user has to perform to retrieve a given information (Conversy et al., 2011).

2.4 Characterization or data exploration tool

To support the characterization of visualizations, I developed a simple software based on the dataflow model. I started with the following statement: if I managed to produce one visualization thanks to its description, then this description is one valid characterization. I found my inspiration in the previous work of J. Bertin (Bertin, 1983) with a graphical description, T. Baudel (Baudel, 2004) and (Wilkinson et al., 2005) with a description close to a programmable language and finally with the C&M characterization table (Card and Mackinlay, 1996).

I called this prototype DataScreenBinder since it takes as an input a data table and with connected lines then binds fields of the dataset to visual variables (Bertin, 1983). Thanks to this prototype I managed to replicate and thus to provide a potential characterization of the radar screen used by Air Traffic Controllers (Figure 6).

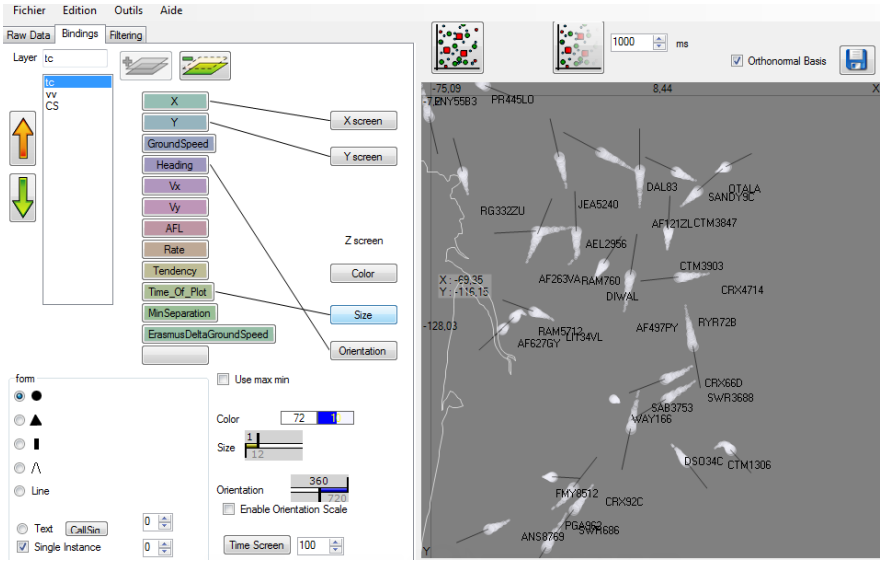


Figure 6: The visual caracterization of the radar screen for air traffic controlers.

Even if this prototype is able to duplicate existing visualization, the produced characterization was not fully suitable to support their detailed comparison. Only a visual comparison between connected lines and the data field names can be performed which is too limited.

Nevertheless this prototype better fits some other purposes such as data exploration with different visual mapping. For instance the same dataset (Figure 6) can be visualized with a circular layout of aircraft speed (Figure 7). Such visualization shows that aircraft flying at high altitudes (large, blue dots) also have fast speeds (close to the border of the circular shape).

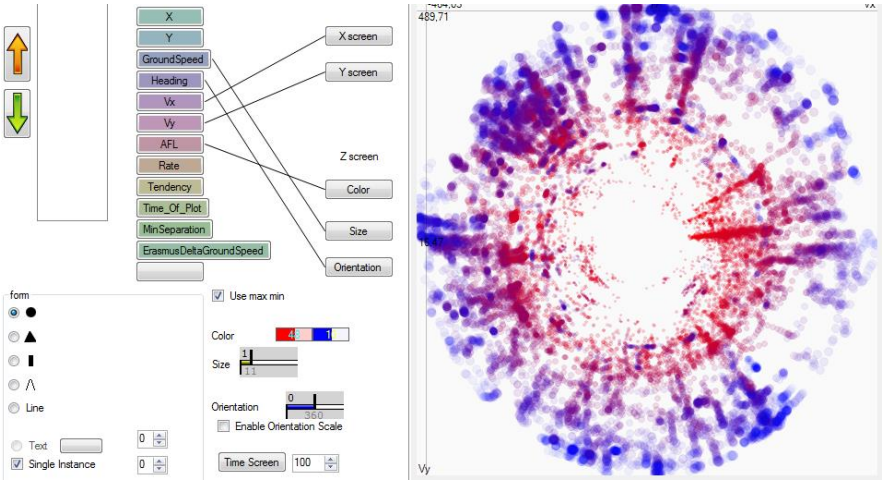


Figure 7: DataScreenBinder and the visualization of aircraft speeds in a circular layout.

This prototype has been intensively extended with interactive techniques like pan, zoom, dynamic filtering, selection layers… This prototype was written in C# and used the GDI graphic library, which hindered the visualization of large datasets (up to 10 000 displayed dots). Therefore I started to use OpenGL/DirectX to support large dataset visualization. This worked very well until I decided to implement animation and brushing techniques. To support such tasks with an interactive frame rate, I had to investigate GPGPU technique (Owens et al., 2008) and thus I developed, thanks to Benjamin Tissoires, the software FromDaDy (Hurter et al., 2009b). At that time, Benjamin was a PhD student working on graphical compilers (Tissoires, 2011) and he was a great help to me by explaining existing GPGPU techniques.

2.5 FromDaDy: from data to Display

Thanks to the first investigation with DataScreenBinder, we developed FromDaDy (From Data to Display (Hurter et al., 2009b)). This multidimensional data exploration is based on scatterplots, brushing, pick and drop, juxtaposed views, rapid visual design (Figure 8) and smooth transition between different visualization layouts (Figure 10). Users can organize the workspace composed of multiple juxtaposed views. They can define the visual configuration of the views by connecting data dimensions from the dataset to Bertin's visual variables. One can brush trajectories, and with a pick and drop operation spread them across views. One can repeat these interactions until a set of relevant data has been extracted, thereby formulating complex queries.

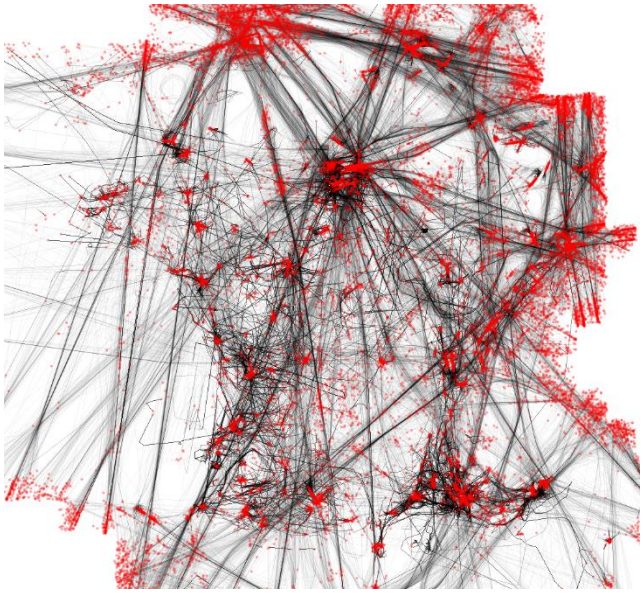


Figure 8: One day of recraded aicraft trajectory over France.

FromDady eases Boolean operations, since any operation of the interaction paradigm (brushing, picking and dropping) implicitly performs such operations. Boolean operations are usually cumbersome to produce, even with an astute interface, as results are difficult to foresee (Young and Shneiderman, 1993). The following example illustrates the union, intersection and negation Boolean operations. With these three basic operations the user can perform any kind of Boolean operation: AND, OR, NOT, XOR… In Figure 9, the user wants to select trajectories that pass through region A *or* through region B. He or she just has to brush the two desired regions and Pick/Drop the selected tracks into a new view. The resulting view contains his or her query, and the previous one contains the negation of the query.

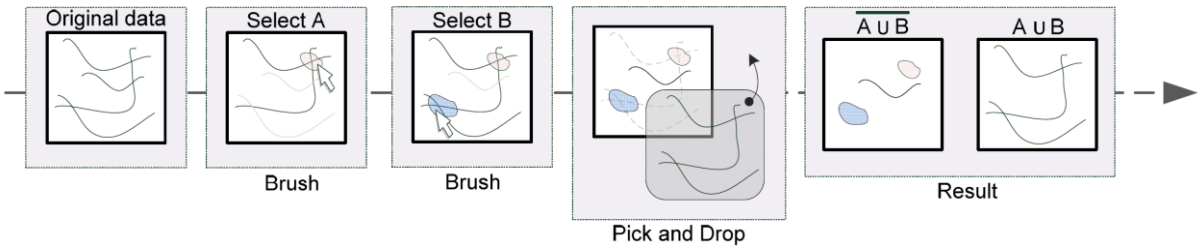


Figure 9: Union Boolean operation

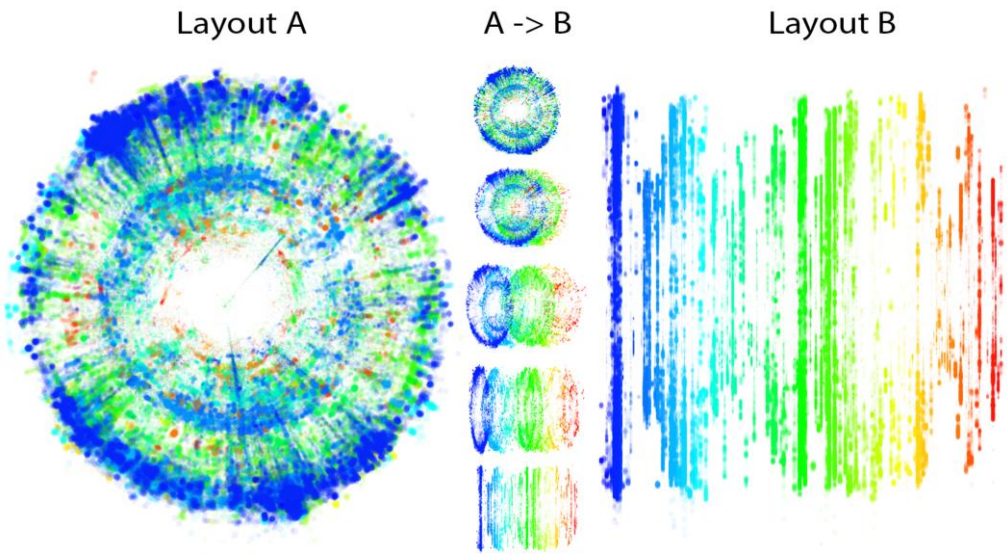


Figure 10: FromDaDy with two layouts and its animation

The brushing technique with numerous points is technologically challenging. Therefore we had to take full advantage of modern graphic card features. FromDaDy uses a fragment shader and the render-to-texture technique (Harris, 2005). Each trajectory has a unique identifier. A texture (stored in the graphic card) contains the Boolean selection value of each trajectory, false by default. When the trajectory is brushed its value is set to true. The graphic card uses parallel rendering which prevents reading and writing in the same texture in a single pass. Therefore we used a two-step rendering process (Figure 11): firstly we tested the intersection of the brushing shape and the point to be rendered to update the selected identifier texture, and, secondly, we drew all the points with their corresponding selected attribute (gray color if selected, visual configuration color otherwise). This technique illustrates my very first usage of an image-based algorithm to outperform brushing technique.

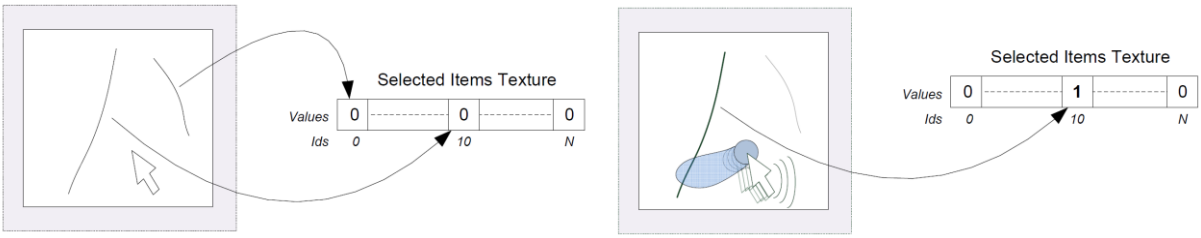


Figure 11: GPU implementation of the brushing technique

2.6 Conclusion

In this chapter, I summarized the work I conducted during my PhD to support the characterization of visualizations with ad hoc methods to depict the *bandwidth* of available information in designs. With a table as a representation for the description, I managed to describe designs that use emerging information. This work is the very first step towards more formal methods to improve the design and re-use of visualization. The characterization of visualizations remains an open area of investigation with the following items left for future work:

- Refine the presented ad hoc method to retrieve all the information of a design and then define completely and automatically the *bandwidth* of each design.

- Define other relevant characterizing dimensions like the *directness* of perception, the amount of emerging information, and its value with regard to specific user tasks.
- Use other representations to describe designs. For instance Wilkinson used a textual "program-like" description (Wilkinson et al., 2005), and Bertin preferred a graphical representation (Bertin, 1983). All of these formalisms allow comparisons of visualization at different levels.
- Propose a generic method to compare designs. Tables can be compared row by row; textual information has to be integrated by the user to make comparisons; training is required to be able to compare graphic information.

Starting from the characterization of visualization, I developed FromDaDy, a data exploration tool. An increasing number of researchers and ATC practitioners were using it and numerous improvements and open questions also had to be investigated. Graphic card usages were also fascinating and promising with emerging technologies like OpenCL/Cuda. For these reasons, I focused my next researches on large dataset exploration with interactive techniques. With serendipity, the characterization of visualization led me toward the usage of image based techniques with the brushing within FromDaDy. The characterization of visualizations needs more longitudinal study and will be part of my long term research topics.

In the following chapters, I will present the research that I performed after my PhD:

- Chapter 3: Density map investigation,
- Chapter 4: Edge bundling techniques,
- Chapter 5: Animation,
- Chapter 6: Strip'Tic,
- Chapter 7: Future research program.

3 Chapter
Data Exploration with data density maps

In this chapter I will detail research mainly performed with FromDaDy regarding data exploration with density map computation. This work tried to address the exploration of dense datasets. Since data density hinders data exploration with numerous overlapping visual marks, I focused part of my research on this topic.

The use of the popular scatterplot method (Cleveland, 1993) is not sufficient to display all information because a lot of overlapping occurs. When transforming data to graphical marks, a regular visualization system draws each graphical mark independently from the others: if a mark to be drawn happens to be at the same position as previously drawn marks, the system replaces (or merges using color blending) the pixels in the resulting image. The standard visualization of this pixel accumulation process is not sufficient to accurately assess their density. For instance Figure 12 left shows one day of recorded aircraft trajectories over France with the standard color blending method. Figure 12 right shows the same dataset with a 3D and shaded density map and one can easily perceive that the data density is drastically higher over the Paris area which is not that obvious with the standard view.

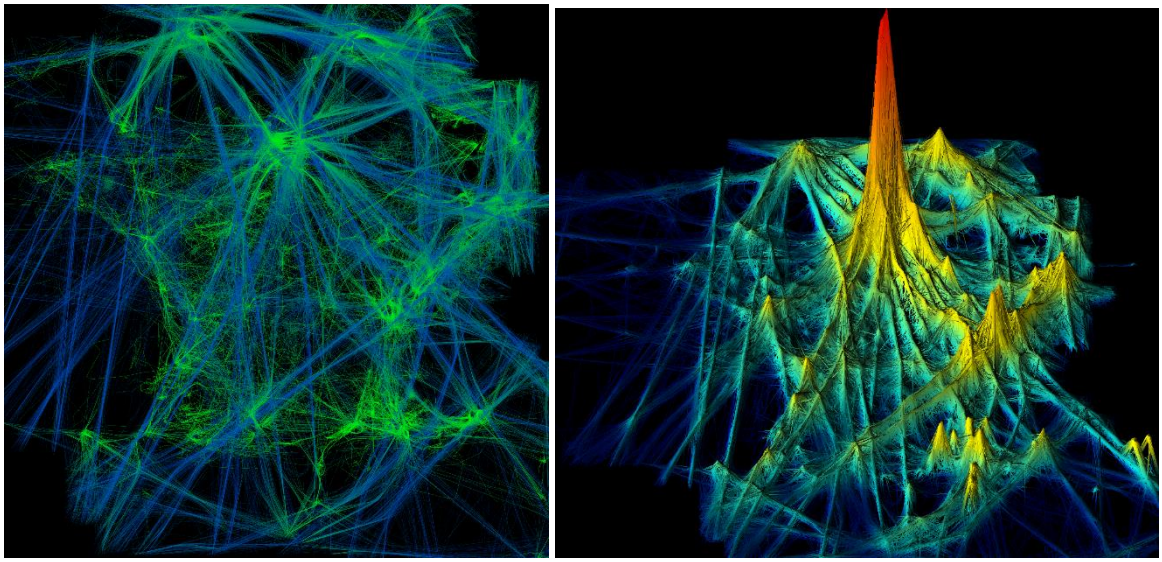


Figure 12: day aircraft trajectory over France (left), 3D density map (right).

I investigated this density computation algorithm with a hardware-accelerated extension of FromDaDy (Hurter et al., 2009b) to support the exploration of aircraft trajectories (Hurter et al., 2010b) with the Kernel Density Estimation (Silverman, 1986).

3.1 Kernel Density Estimation: an image based technique

Kernel Density Estimation (KDE) (Silverman, 1986) is a mathematical method that computes density by a convolution of a kernel K (Figure 13: Kernel profiles) with data points. This method produces a smooth data aggregation which also reduces data sampling artefacts and is suitable for showing an overview of amounts of data.

Given a graph $G = \{e_i\}_{1<i<N}$ consisting of edges $e_i \in \mathbb{R}^2$ and a point $x \in G$, we can estimate the local spatial density ρ of points x using the kernel density estimation:

$$\rho(x) = \sum_{i=1}^{N} \int_{y \in e_i} K\left(\frac{x-y}{h}\right)$$

Where K: $\mathbb{R}^2 \to \mathbb{R}^+$ is the so-called density of bandwidth $h > 0$. Typical kernel choices are Gaussian and Epanechnikov (quadratic) functions (Figure 13). ρ can be computed by convolving G with K, or building an accumulation map of K over G.

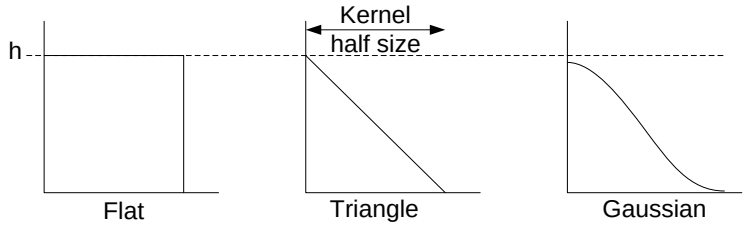


Figure 13: Kernel profiles

The density ρ can be visualized as a 2D height field by a straightforward color map, contour plot, or terrain map (Figure 12). Landscape visualization with hills and valleys have been shown to be easy to interpret (Wise et al., 1995). For quantitative analysis, a contour plot is preferred over a colormap, since value estimation by colors is perceptually difficult. Since contour plots only use isolines, color can be used for other purposes. In 2D, the density plot becomes visually more detailed by using shading and can be enriched to a contour map (van Wijk and Telea, 2001). KDE maps can be interactively explored and modified (Van Liere and Leeuw, 2003). The KDE algorithm has also been used to investigate objects movements (Willems et al., 2009), (Scheepens et al., 2011).

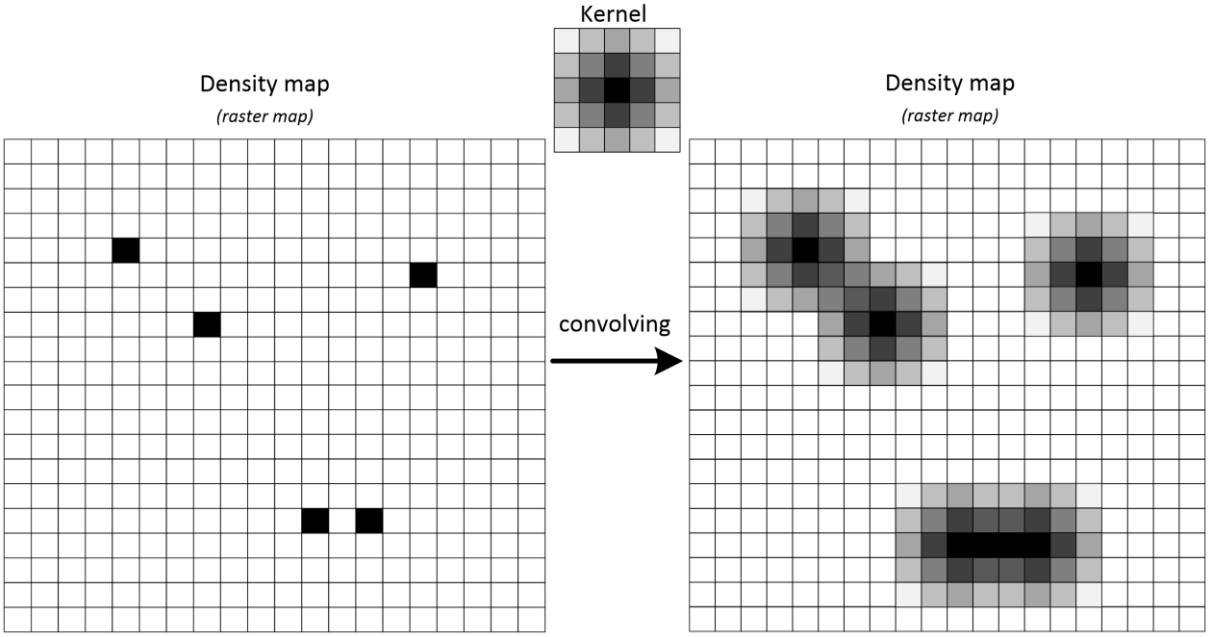


Figure 14: Kernel Density Estimation convolving principle with a raster map.

Figure 14 shows a simple principle to compute the local spatial density of points with a kernel and a density raster map. The kernel map is applied (convolved) at every point location. This technique can

be considered as an image-based algorithm and produces a grid with a smooth transition between the cells.

The initial version of FromDaDy computed the density map and stored it in an off-screen buffer with additive blending enabled. To achieve this, fromDaDy used a 32-bit floating color texture. More recent versions use OpenCL with the full support of the convolution process with tabular data. To ensure data integrity with the multi-threaded data processing, FromDaDy uses atomic functions. This GPU implementation allows interactive manipulation and visualization of the density map. The computation time varies with the number of points in the data set and the Kernel size. As an example, the frame rate is around 10 frames per second with 400 000 points and a Kernel point size of 20 pixels and a Nvidia GTX 275 graphic card.

3.2 Density map visualization and interaction techniques

In the following, I describe a set of interaction and visualization techniques with density maps to perform interactive data exploration. Thanks to a GPU implementation the users can interact in real time with the density map and the process is divided into three steps:

- Users can choose which data dimensions to accumulate, and can adjust the kernel size,
- Users can “brush, pick and drop” data to remove them from, or add them to the density map,
- Users can explicitly choose to use the computed density values with one of the available design customizations (color, size or position).

3.2.1 Brushing Technique with density maps

Originally, FromDaDy supported the brushing of trajectories with their spreading across views. This interaction helps to select an entire trajectory with the brushing of only few points, but in certain cases, the data exploration requires only parts of trajectories. We added the brushing of points, which allows the selection and manipulation of points. The user can brush in the standard view or in the accumulator view. The user uses a size configurable round shape to brush the view to selected trajectories or points (Figure 15).

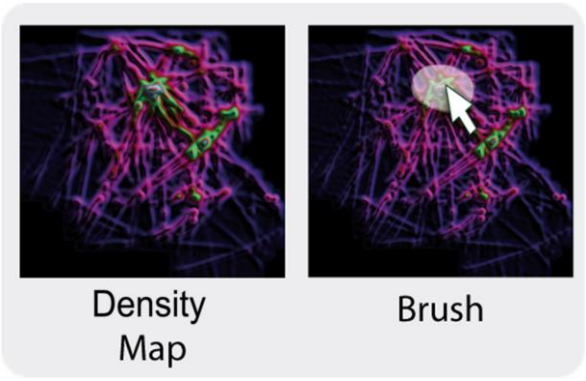


Figure 15: Brushing over a density map.

Thanks to the brushing technique, the user can select and highlight parts of the displayed data. By pressing the space bar, the user can extract previously selected data and attach them to the mouse cursor. By default, the selected data are *picked*: they are removed from the view, and appear in a “fly-over” view. When the user presses the space bar for the second time, a *drop* occurs in another view under the cursor. Although it resembles a regular drag’n’drop operation, we prefer to use the term

"pick'n'drop" (Rekimoto, 1997) in the sense that data is removed from the previous view and is attached to the mouse even if the space bar is released.

We also added the pick and drop paradigm to the accumulator map. Figure 16 shows the difference between the point and the trajectory mode. With the point mode only the brushed points are selected and isolated. With the trajectory mode, brushing points also selects their entire trajectory.

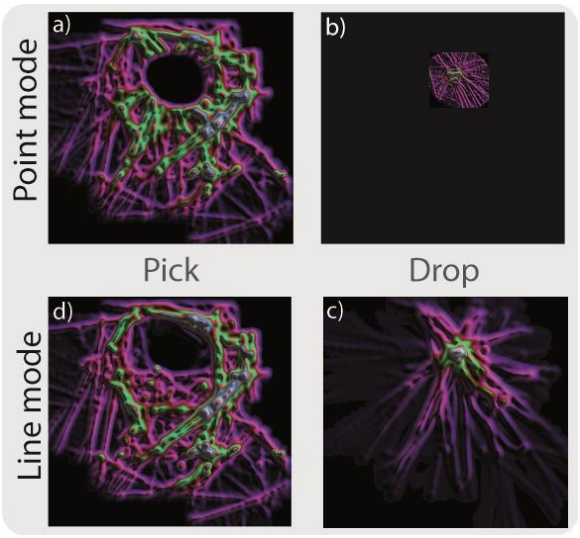


Figure 16: Points or pick and drop trajectories on an density map.

The pick and drop of brushed data, from the accumulation map to another view, is useful during the exploration process for three reasons:

- It helps to isolate data to perform separate analyses,
- In the trajectory mode, the brush selects entire trajectories. When picking these trajectories, a new accumulation map is computed and unveils new accumulation initially hidden by the picked trajectories (image *d* in Figure 16),
- With the point and the trajectory mode, FromDaDy uses the full gradient scale in such a way that the minimum accumulation value has the first gradient color and the maximum accumulation value has the last gradient color. When brushing/picking and dropping points with minimum or maximum accumulation value, FromDaDy computes a new accumulation map that unveils a new maximum value with the maximum gradient color and then unveils new patterns (comparison of Figure 15 and Figure 16).

1.1.1 Interactive lighting direction

In order to compute the shaded density map, one can consider it as a height map and use the standard Phong light computation(Phong, 1975). Since this technique needs a normal vector, a normal map can be generated thanks to the computation of the gradient of the density map. The user can choose to display the accumulator map with, or without, this shading and interactively set the lighting direction with the mouse pointer. High accumulation values are considered as mountains that create shade, and low accumulation values are considered as valleys. By pointing with the mouse pointer to a specific area, the lighting direction can be interactively manipulated. This manipulation allows furrows or ridges to be emphasized. When defining the lighting direction from the left or from the right, vertical furrows are accentuated whereas a lighting direction from the top or bottom emphasizes horizontal furrows (Figure 17).

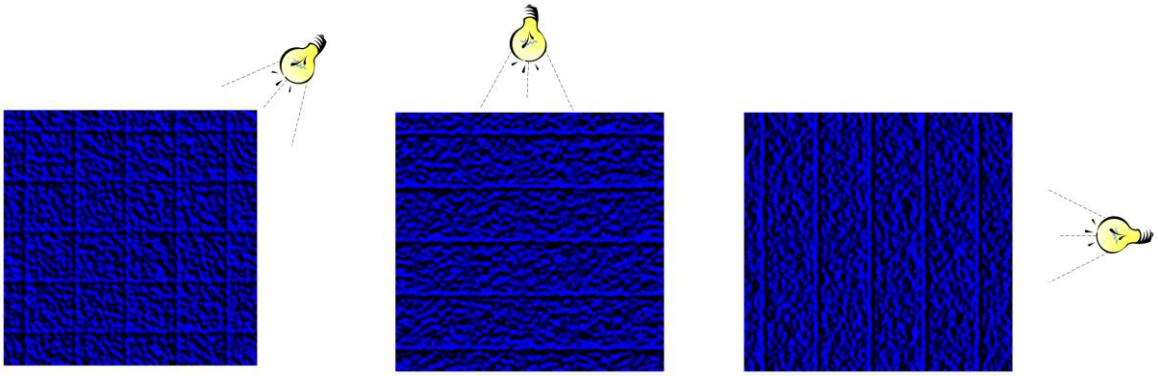


Figure 17: Interactive light manipulation to emphasise ridges and furrows on a shaded density map.

3.2.2 Density maps as data sources

The color blending process computes an implicit density map since it combines pixels with the following blending formula:

OutputPixel = SourcePixel × SourceBlendFactor + DestinationPixel × DestinationBlendFactor; with 4D color vector (r,g,b,a) and the × symbol denotes component-wise multiplication.

Even if this formula can be customized, the pixels produced do not always provide an efficient quantitative comparison of the accumulation value. In Figure 18, the visualized data base is a one day record of aircraft flight plans (the routes that aircraft are supposed to follow). The view shows a matrix of points with the aircraft departure airport on the X axis and the aircraft type (Boing 747, Airbus A380…) on the Y axis. Since many aircraft have the same departure airport and aircraft type couple, many points in the matrix have the same location on the screen. In the standard blended view, the brighter points show the most frequent couple in one day's traffic (Figure 18 left).

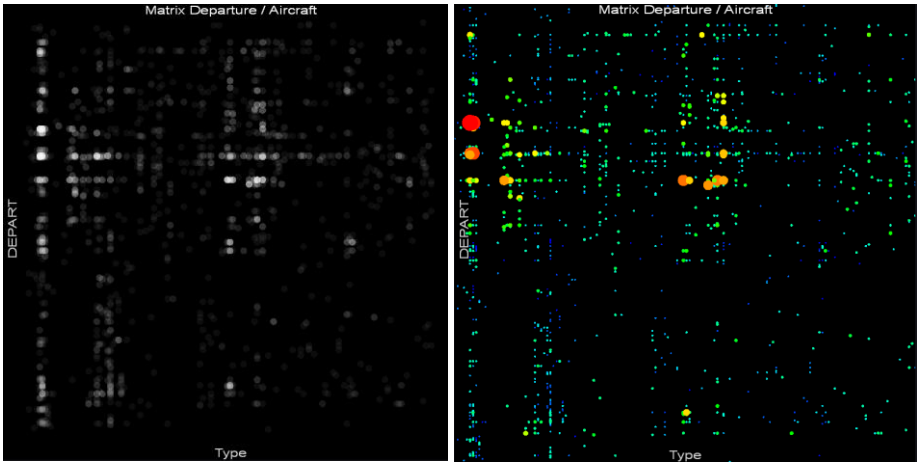


Figure 18: Matrix view with standard color blending (left) and customized visual mapping with the size (right).

FromDaDy offers another solution with a specific visual mapping. First, the user defines the data fields he or she wants to investigate (departure, aircraft type). Secondly, the system computes the corresponding density map. Finally the user defines the visual mapping of the output view. In Figure 18, the density is mapped to the size and the color. Figure 19 summarizes this configuration. This process operates as if a new field was provided into the dataset. The computed density map acts as a new data source.

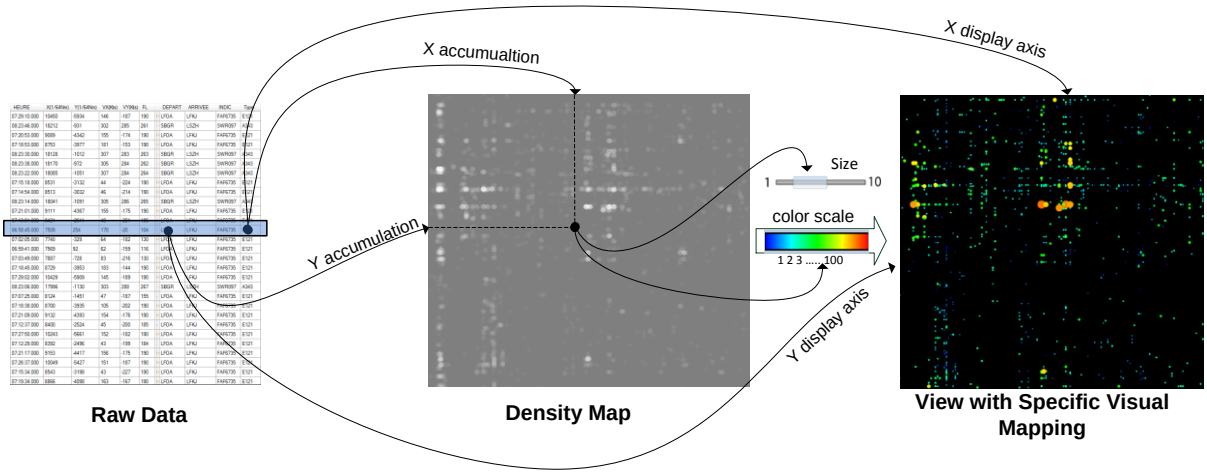


Figure 19: KDE maps value displayed with the size visual variable to display a matrix

3.3 Application domains

In this section I will give some examples of usage in specific application domains. Additional examples are provided in a survey dedicated to air traffic control (Hurter et al., 2014a).

3.3.1 Exploration of aircraft proximity

The main activity of air traffic controllers consists of maintaining safe distances between aircraft by giving clearances to pilots (heading, speed, or altitude orders). However, when aircraft fly below the safety distance, an alarm is triggered. These alarms are common since Air Traffic Controllers supervise aircraft in dense areas. Nevertheless they are all monitored to avoid aircraft collision. In this example, the dataset contains only safety distance alarms with the recorded aircraft positions. The user connects the X and Y position of each aircraft to the X and Y density entries. The computed accumulation is visualized with a blue (low accumulation values) to red (high accumulation values) color scale.

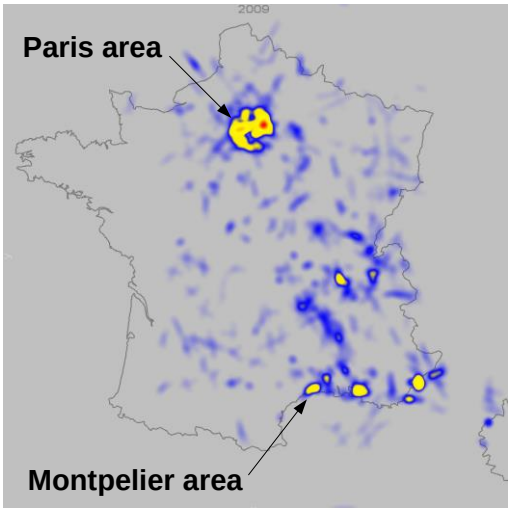


Figure 20. Density map of the safety-distance alarms triggered over France over a one year period. Red colored areas correspond to dense alarm areas where aircraft triggered proximity alerts

Paris is of course the main dense area with the largest proportion of alarms. However, when visualizing all these alarms over a one year period, users discovered that some unexpected dense areas emerge

(Figure 20): for example Montpellier, which is a far smaller airport, shows a lot of alarms. The user can use the selective brushing to retrieve the exact number of alarms.

3.3.2 Patterns detection in dense datasets

This first example shows how density maps and their height map visualization can be used to isolate relevant aircraft trajectories. With the recording of one day's flights over France, each datum represents the position of an aircraft at a certain time. The corresponding density map (Figure 21) is the result of the accumulation of plots with a triangular kernel. Hence, the X and Y position of each plot is mapped on the X and Y dimensions of the density map and on the X and Y dimensions of the resulting image. The image produced shows very dense areas over the main airports in France (Roissy, Orly, Lyon...), which was expected.

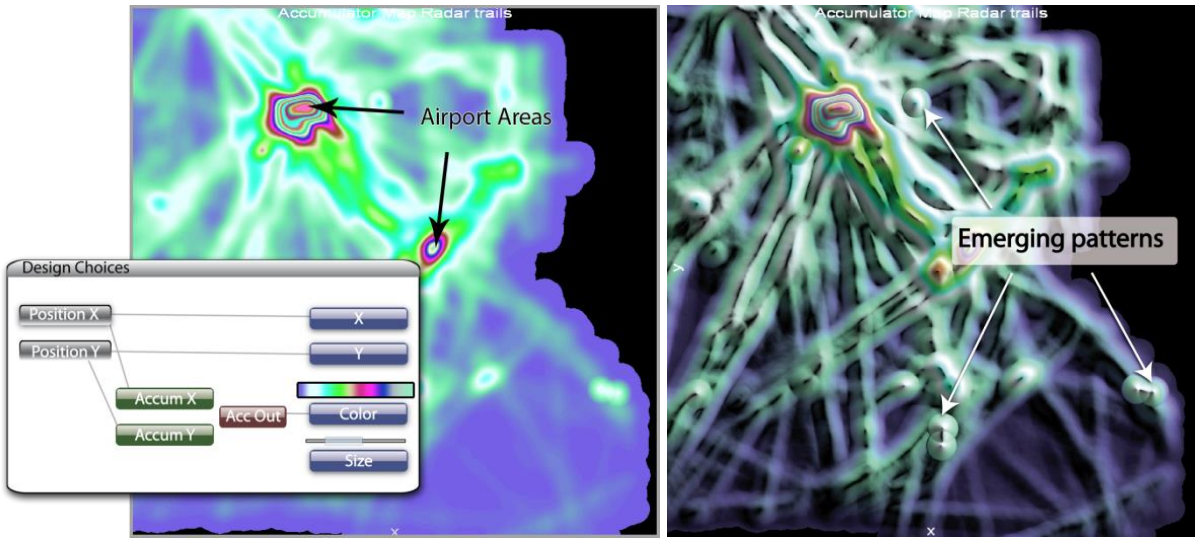


Figure 21: Design configuration and accumulation maps without shading.

When visualizing the density map with illumination, circular shapes emerge (Figure 21, right) that were not initially noticeable (Figure 21, left). Thanks to the shading process, density gradients are emphasized and this is the reason why these circular hills stand out. The user can then brush these shapes to extract the aircraft that cause this accumulation of data recording. Thus, the user brushes the hills and drops these data onto a second view. The user discovers that the picked trajectories correspond to stationary radar test plots recorded throughout the whole day. These radar test plots are mandatory to assess the correctness of the whole Radar data processing (merging of multiple radar sources)(Renso et al., 2013).

3.3.3 Density flaw detection in a dense dataset

In this example, we use the data density as a tool to highlight flaws in the dataset. The dataset is a one-day record of aircraft positions. Radars send data over networks with a constant stream rate (in our dataset, one radar position of each aircraft every 4 to 8 minutes). Figure 22 shows the content of our dataset. The X screen axis shows the time of each radar plot and the Y screen axis shows the aircraft's identifier. Since the identifier of each aircraft is a number incremented over the day, the resulting shape shows a remarkable continuous pattern in which each horizontal line represents the lifetime of one flight (each flight has a unique identifier). The longest lines at the bottom of the visualization are the stationary Radar test points recorded all day long. The width of this shape gives the average flight duration in the dataset: it is about 2 and a half hours which represents the average time to cross France by airplane (most aircraft cross France at a high altitude).

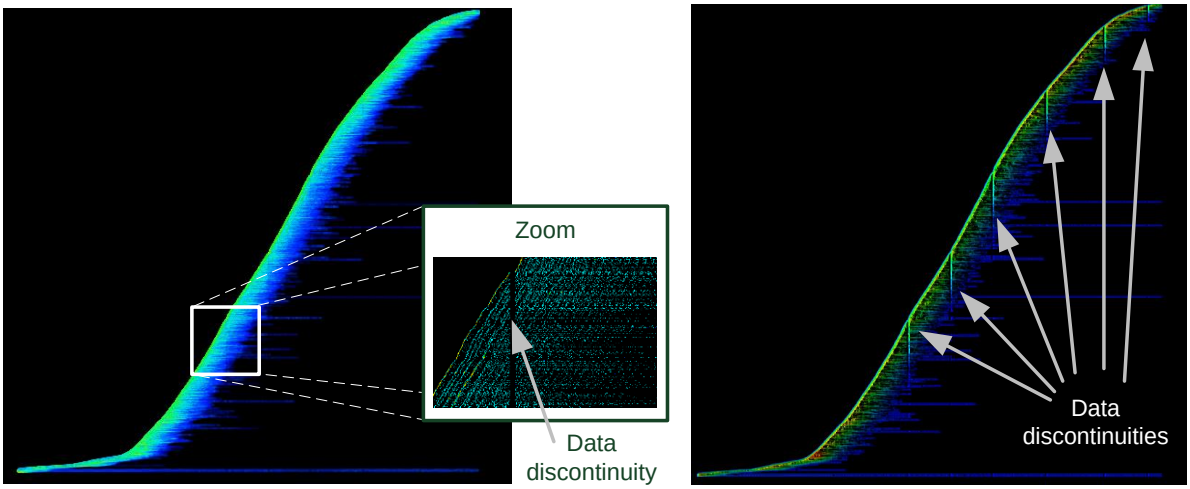


Figure 22: Time series of the incremental number of aircraft over time (left) and the corresponding density map (right).

Due to the large number of records, a lot of cluttering occurs when displaying the time series of the aircraft's identifier (Figure 22). This clutter hides the areas where, during a small time period, no data are recorded. Hence the user is unable to discover this flaw in the database, unless by chance to zoom over the specific areas to reduce the cluttering of points, which is unlikely (Figure 22, zoom part). To notice this flaw without serendipity, the user can display the corresponding data density with shading (Figure 22 right). Thus, the density view unveils continuous and discontinuous data streams. A continuous data stream over time produces flat accumulations (the same amount of data are accumulated over time), whereas a discontinuous data stream produces ridges (increase of the data stream rate) or furrows (decrease of the data stream rate).If no data are recorded during a specific time-span, the produced accumulation view displays many furrows. Each of these furrows indicate that during the time corresponding to the thickness of the furrows, no data were recorded which reflects a failure in the recording system.

3.3.4 Exploration of gaze recording

In the last example, I conducted an experiment that used an eye tracker. During the experiment, users were required to look at the center of the screen, then at a target located elsewhere on the screen, and finally to look back at the center of the screen. The database contains 200 eye trails with a total of 100,000 points. We used the accumulator map to reveal the speed of the users' gaze and eye fixation locations. To do this, we used the following configuration to compute the density map: trajectory Id is mapped to the Y axis, and curvilinear distance to the X axis (Figure 24 *a*). The curvilinear distance corresponds to 0 at the beginning of the trails and increases until the trajectory ends. This distance is correlated with time, since trails are regularly sampled: hence, accumulation occurs between sampled points when the gaze speed slows down. The last step of the design is to retrieve for each gaze point its corresponding density. To do so, we map the density result to the size and the color of the displayed lines (Figure 20).

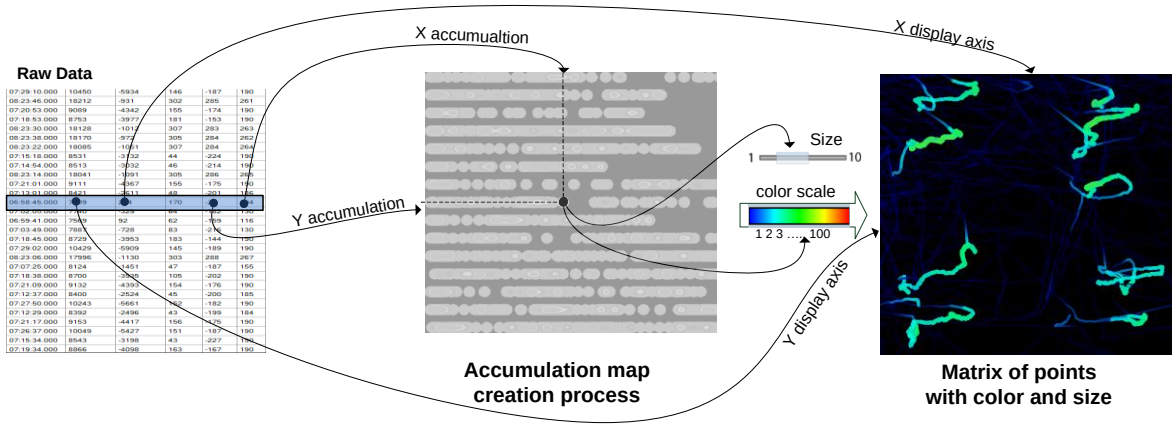


Figure 23: Density map computation with gaze recording.

Figure 24 *d* shows the eye trails with standard color blending (recorded gazes are displayed with transparent dots): main stops are visible. Figure 24 *c* shows the same trails with the use of the computed density map. The stops are visible with finer details thanks to the color scale and the variation of size. Thanks to the interactive brushing technique, the user can investigate specific trails in more detail.

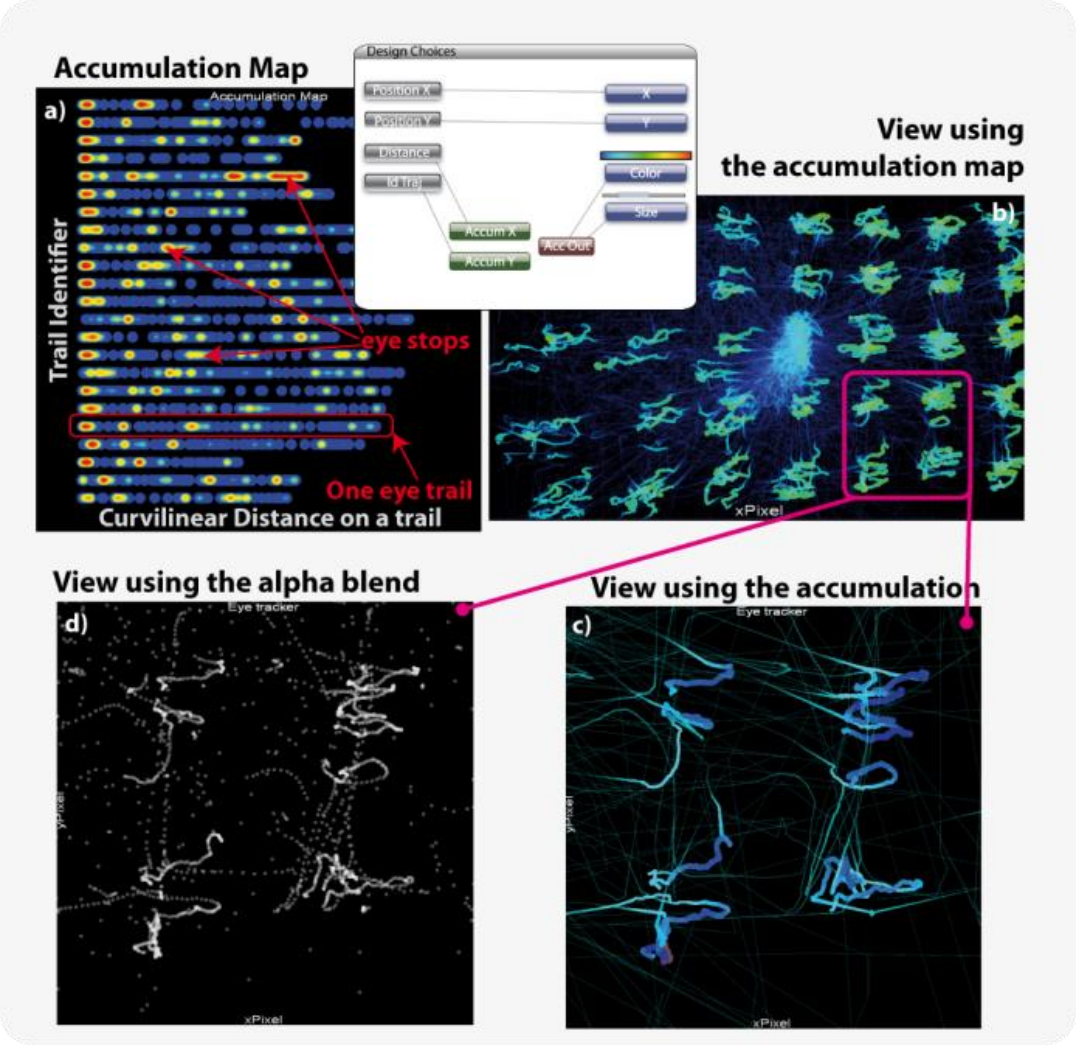


Figure 24: Accumulation view and its configuration to produce a per trajectory distance accumulation (*a,b*). Comparison between trail visualization with or without the accumulation (*c,d*).

3.4 Conclusion

In this chapter, I described a way to interact dynamically with density maps. Users can customize them (data fields, kernel size), manipulate them with the “pick and drop” technique and set up the shaded view to support data exploration.

Thanks to the graphic card computation power, this image based density computation can support interactive data exploration. Even if graphic card power never stops improving, density computation remains challenging with large dataset or with big kernel size. Improvements in this area are possible by, for instance, using a better algorithm such as Divide and Conquer to implement an efficient parallel density map computation.

I started the investigation of the density map with graph splatting to address the data overlapping issue, but as I will explain in the following chapter, the KDE algorithm was the first stop toward a far more efficient technique: edge bundling.

4 Chapter
Edge bundling

This chapter relates how my adventure with edge bundling started and how it has become one of my major topics. This story started in 2010 with Jason Dykes who invited me to a Dagstuhl seminar "Schematization in Cartography, Visualization, and Computational Geometry". I presented my work regarding the schematization of Flight Routes for air traffic controllers (Hurter et al., 2010a). This was my very first experience where we presented our work but also our ongoing research projects. I had many fascinating discussions and one of them with Alex C. Telea where I presented my semantic lens prototype to him, Figure 25.



Figure 25: First mole view prototype with a semantic lens

This prototype was inspired by the paper "Powers of ten thousand" (Lieberman, 1994) when navigating in large information space. My rationale was to use data deformation rather than data filtering and thus display all information in a distorted context. I started the development of this prototype during InfoVis 2010, and I knew it was far from being ready for publication. Its contribution remained too small. In this prototype, a lens pushes at its border points whose semantic do not correspond to the requested one (focus plus context technique). I also had labels with a basic avoidance process (gradient computation) (Tissoires, 2011). This prototype was CPU based and thus pretty slow (maximum 40000 points could be handled with a reasonable frame rate). During this Dagstuhl seminar I joined Alex's group on schematization and edge bundling. During these fascinating discussions, Alex taught us the current state of the art in this area, and the available techniques. From these discussions emerged two possible directions: one was to use the skeletisation technique to produce edge bundling (Alex and one of his PhD students, Ozan Ersoy, were currently investigating this area), and the other was the development of an interactive tool to investigate bundling results. After this seminar we continued to work on these topics trying to find a suitable tool to help users in the investigation process.

4.1 MoleView

My initial prototype computation is a vector force base interactive system which resembles to the dust and magnet system (Yi et al., 2005). With a small change in the algorithm, I could handle dual layout visualization, and use the lens as an interactive tool to navigate between these two layouts. Therefore, I sent Alex one of my graphs (airline movements), and he sent me back its bundled version with its new Skeleton Based Edge Bundling. The animations produced (Figure 26) were smooth and significantly helped to link edges from their original location to their bundled one. This investigation helped to detect flaws and parameter issues in the bundling process: too tight bends, incorrect bundles…

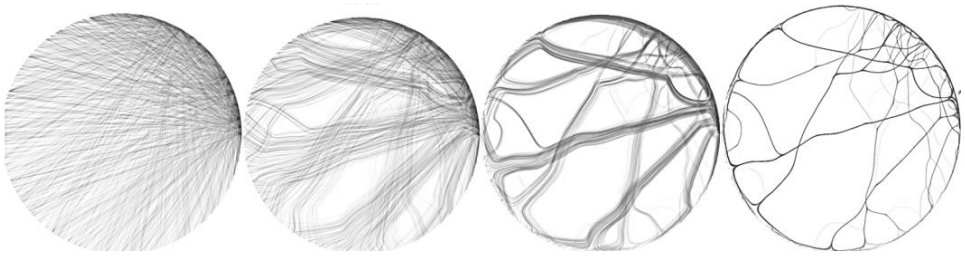
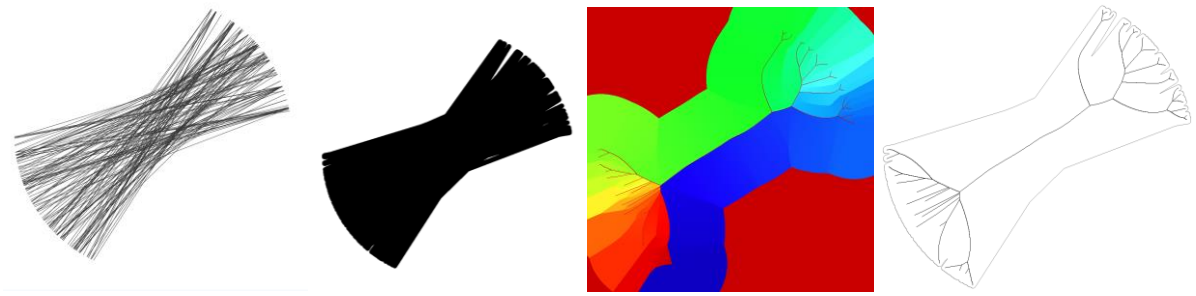


Figure 26: Exploration of the original version and the bundle version of a graph (Hurter et al., 2011b).

The following investigations with the mole view were fast, since the technical and contribution part were already performed: we developed the first tool to interactively investigate a dual layout with a bundled and an unbundled graph.

4.2 SBEB: Skeleton-based edge bundling

When I joined this project, it was already advanced. The mole view (Hurter et al., 2011b) provided a suitable tool to investigate the bundled edges results. I also provided aircraft datasets and my aeronautical expertise. This bundling technique is a pixel based technique which computes the skeleton of a splatted graph and attracts edges toward it (Figure 27). A clustering stage is mandatory to produce detailed bundles. Ozan Ersoy, one of A. Telea PhD students implemented the directional clustering method and edge bundling processing (Ersoy et al., 2011). This algorithm only needs two parameters to make it work: its attraction factor and its smoothing factor. Some post processing can be performed to improve the visualization such as edge borders (Telea and Ersoy, 2010) and relaxation (to slightly unbundle the graph).



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

Figure 27: Skeleton computation. A set of edges (a) is inflated (b), then the distance tansform is computed (c) and finally the skeleton is extracted (d).

4.3 KDEEB: Kernel Density Edge Bundling

In October 2011, Alex invited me to the University of Groningen in the Netherlands. During the visit Alex and I had numerous discussions while trying to improve the mole view, and discussions around time dependent datasets (e.g. trajectories). In the middle of my stay, I had to present two papers at

InfoVis 2011, in Providence USA: SBEB with Ozan Ersoy and The mole View. InfoVis is the best place to gain inspiration and I found at that time the rational of a new bundling algorithm. My main concern with edge bundling algorithms was their complexity and their data requirements. Hierachical Edge Bundling (Holten, 2006) is the fastest algorithm but needs a data hierarchy which drastically restricts its usage. Force Directed Edge Bundling (Holten and van Wijk, 2009) is flexible enough but its computation time is prohibitive when using large a dataset such as one day of aircraft trajectories. Skeleton Based Edge Bundling (Ersoy et al., 2011) is scalable, but needs data clustering and the skeleton computation takes time (one can use CUDA, but its implementation remains complex). SBEB was the first algorithm to use pixel based algorithm: for the skeleton computation and the gradient map computation. These gradient maps will attract edges toward the medium skeleton axis. My idea was simple: since the skeleton and the gradient were the complex and time consuming parts of this algorithm, I wondered if there were not another solution to compute them. I had the intuition that using a density map rather than a skeleton map to compute the gradient could be a potential solution. I postulated that when attracting an edge toward a dense area, I would make them overlap and thus create a denser area but also empty spaces. This iterative algorithm may produce a clearer view, and a new bundling algorithm. I remember when I came back from Groningen and I explained this algorithm to Alex who answered that this could not work since we could not understand why this algorithm might converge. I showed him my prototype with a very bad bundling result, but in practice this algorithm converged.

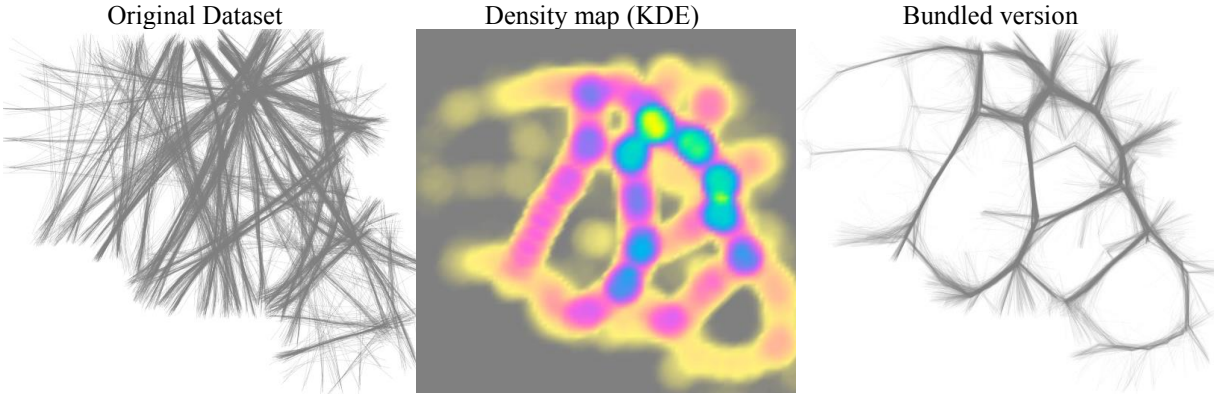
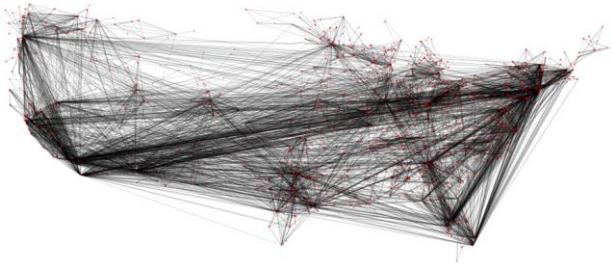


Figure 28: First bundling prototype based on density computation

Thanks to Alex and Ozan we drastically improved this algorithm by adding a resampling process, smooth density map computation, and many investigations with different datasets. This algorithm uses a density map and thus we can directly take this information to improve the visualization of other techniques, for instance, bump mapping (Figure 29). We added at that time a new bundling algorithm, but no formal explanation was provided to explain how it converged. It took us almost one year to find the explanation and the Mean shift algorithm had already proved it in a different field: data clustering (Comaniciu and Meer, 2002).



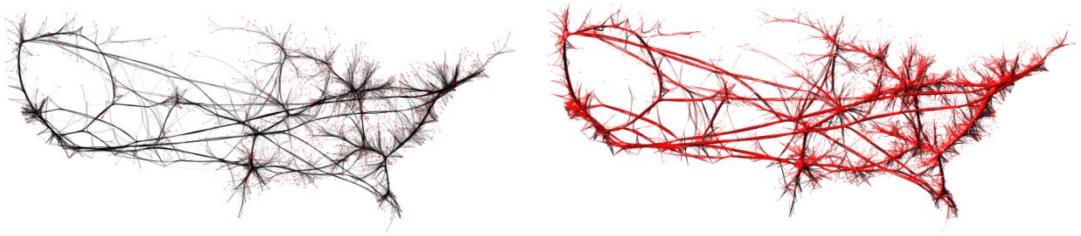


Figure 29: US migration graph. Original (top), bundled (bottom), with shaded density (bottom right).

As an extension we also added Ozan's work with obstacle avoidance which uses the distance transformto compute a way that an area will not be overlapped by bundled edges (Figure 30).

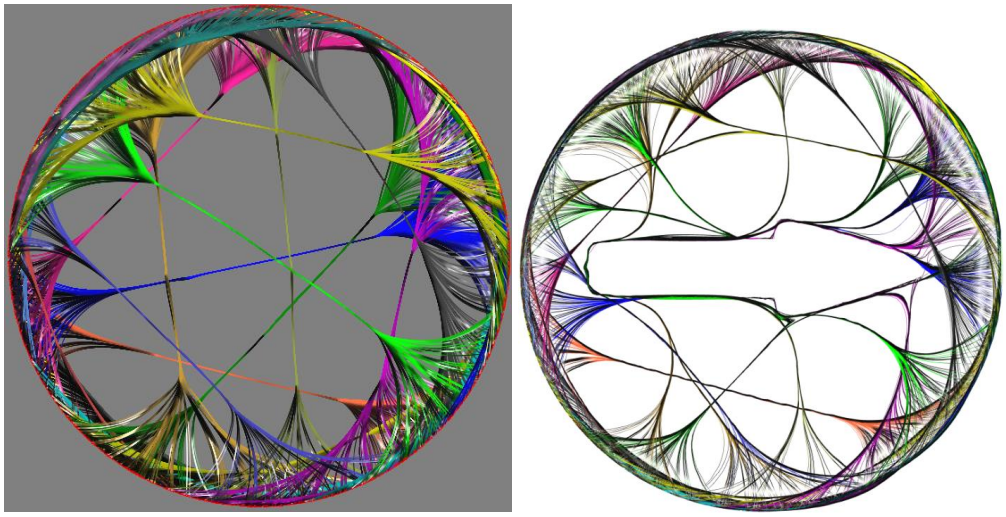


Figure 30: Bundling of dependency graph with obstacle ovoidance (right).

4.4 Dynamic KDEEB

We did many extra investigations after our publication of KDEEB (C. Hurter et al., 2012). One of them was to discover why this algorithm converges (Comaniciu and Meer, 2002), and why it is noise resistant: with the same parameters (kernel size, attraction, interaction, smoothing, resampling) a bundled graph remains almost the same even if a small number of edges are added or removed. This property is a great asset, especially to produce a bundled version of a dynamic graph. A dynamic graph has a given amount of edges that change over time (removed, added or displaced). Previous attempts only managed to produce dynamic graph visualization with large jumps between key frames, and with reduced datasets (Nguyen et al., 2013a). KDEEB uses an accumulation map which will not drastically change over time and thus ensures continuity of the bundled dynamic graph. Furthermore, KDEEB uses a pixel based technique which is highly scalable, and is probably the fastest bundling algorithm (except for HEB (Holten, 2006) which has only a display time and no data processing). With the algorithm assets, nothing prevented us anymore from producing the first bundling algorithm of a dynamic graph. This initial publication depicts the dynamic bundling principle with airlines and software visualization dynamic graph exploration (Hurter et al., 2013b). And thanks to my new interest in eye tracking systems, we also discovered a very promising usage of bundling techniques with eye gaze (Hurter et al., 2013a). As an example, one can bundle the pilots' gaze during a landing sequence and discover the standard monitoring pattern (Figure 31).

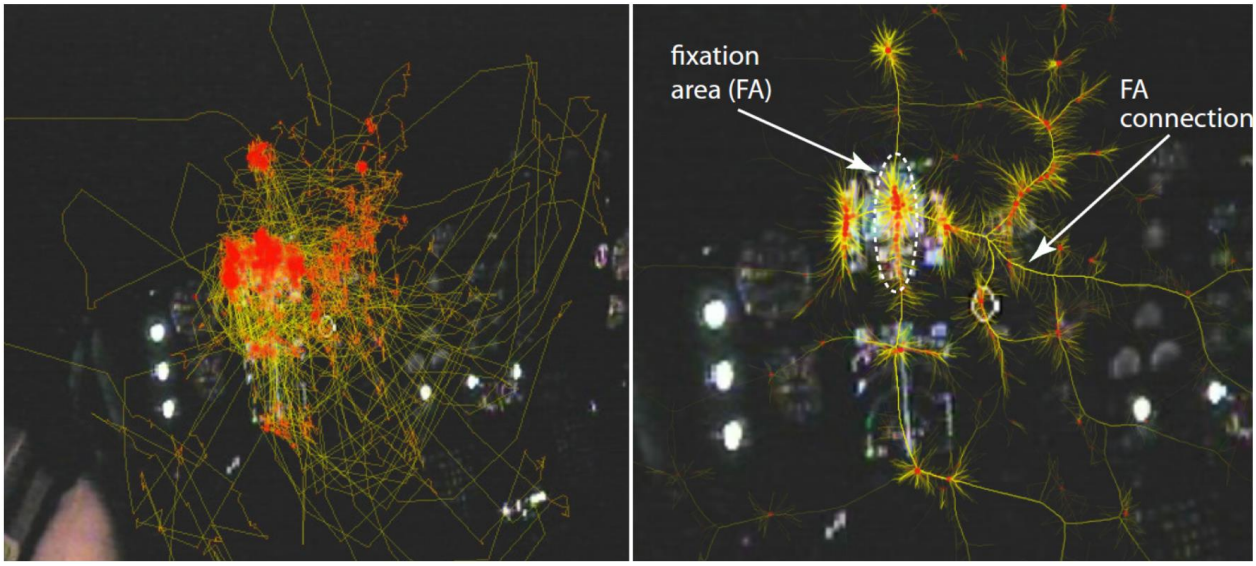


Figure 31: Eye gaze recording of a pilot when landing with a flight simulator. Bundled trails (right) with KDEEB.

4.5 3D DKEEB

I also investigated an extension of the Kernel Density Edge Bundling to take into account 3D graphs (x,y,z). This work is a simple extension of the original algorithm (C. Hurter et al., 2012) in which a 3D density map is computed with a 3D gradient. The computation time drastically increases due to the additional volume of data. The density map becomes a 3D texture (e.g. 400^3 voxels) and the gradient becomes a 3D vector.

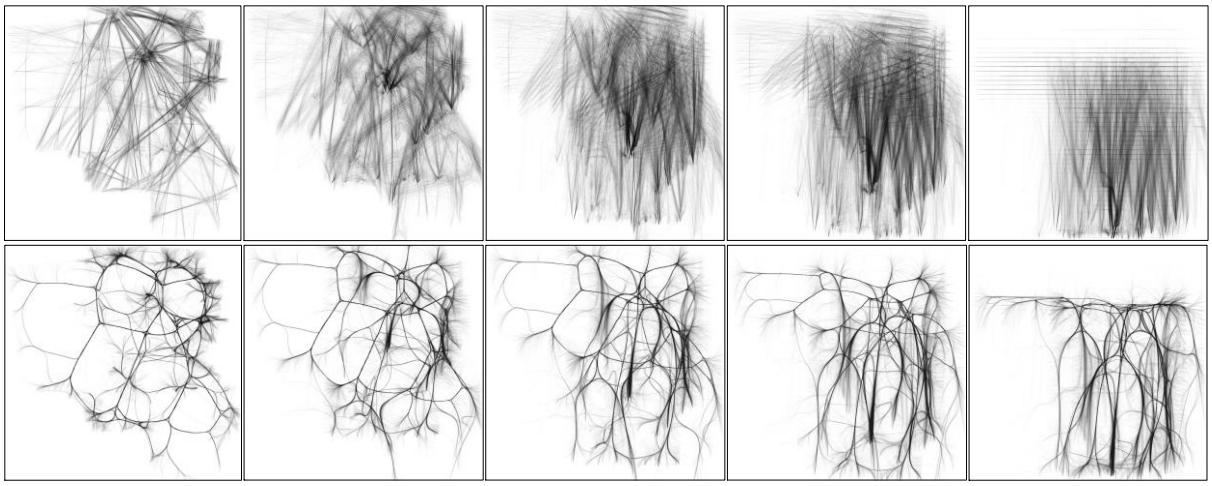


Figure 32: 3D DKEEB, top view.

The results are visually compelling by reducing the clutter of the 3D visualization of aircraft trajectories (Figure 32) but do not provide sufficient increment to be published. I also investigated space time cube bundling, with is not subject to any technical limitation: the z axis can directly be mapped to the time dimension.

4.6 Directional KDEEB

KDEEB has proved to be a very promising algorithm as it is scalable and convergent, but one limitation remains in the usage of direction graph/trajectories. When visualizing an aggregated view of aircraft trajectories, it is nonsense to bundle opposite direction trajectories. To solve this issue, one can use preprocessing clustering. During a discussion with Vsevolod Peysakhovich, one of my PhD students, he

suggested a simple extension of KDEEB to take into account the direction. Instead of using a gradient map, we also computed an accumulation direction map. For each pixel in a raster map we computed the average direction of the edge which would be located over this pixel. To apply the gradient, we first tested the compatibility of the edge direction and the average direction. If they were close, the gradient was applied, if not, the edge did not move. This extension did not hinder the algorithm complexity which remained linear O(E). In addition, we managed to take into account the edge direction. We also implemented a fast GPU software to help the scalability.

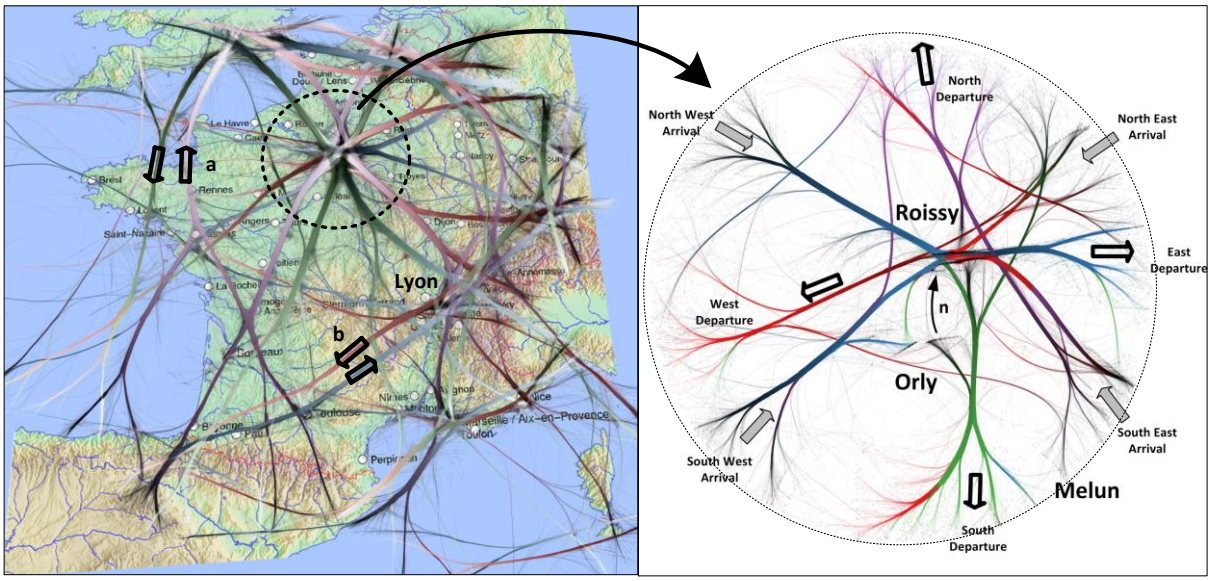


Figure 33: Investigation of aircraft trails with a directional bundling algorithm.

Figure 33 shows the global data visualization with the Direction-KDEEB technique. Trails with the same direction have been bundled. The color corresponds to the global trail direction; we computed the direction for each trail with its start and end point. This is valid, since during a flight aircraft can change directions (follow a specific flight route, avoid dense area of traffic, apply air traffic controller orders), but each aircraft has a main direction given by the start and end points. We chose color coding with a bright color at the start and a dark one at the end point; in this way we emphasized the visualization of outcoming and incoming bundles. As such, the London and Paris areas contain numerous incoming and outgoing flows, which makes these areas dense and complex.

The visualization also highlights the complex configuration over the Lyon Area (see Figure 33 left). The Lyon area is a central crossing for European flight routes which does not have the same flow configuration as Paris. Whereas the Paris area does not have transit aircraft (aircraft that fly over a given area without landing), the Lyon area does have major transit flows (e.g. Switzerland/France/Spain, arrows b). In terms of traffic management, one can observe that many airways are grouped in pairs with opposite directions. This is especially the case with arrows a to reach and to exit the London area, and the airways between Switzerland/France/Spain (arrows b in Figure 33). The width of the bundles also highlights traffic density between aircraft. Arrows b, trails between Switzerland/France/Spain, are balanced in their two directions. In the Paris area, flows are mainly incoming, or outgoing from the North/South rather than from the East-West. This can be explained by the traffic from/to United Kingdom and the location of Paris which is in the north of France (a lot of aircraft connect southern cities).

In Figure 33 right, we applied the Direction-KDEEB algorithm on a subset of the trail dataset. Due to the density and the dataset size of aircraft trajectory, no previous investigation had been previously able to bundle and extract a meaningful result, especially over the Paris Area (Figure 34).

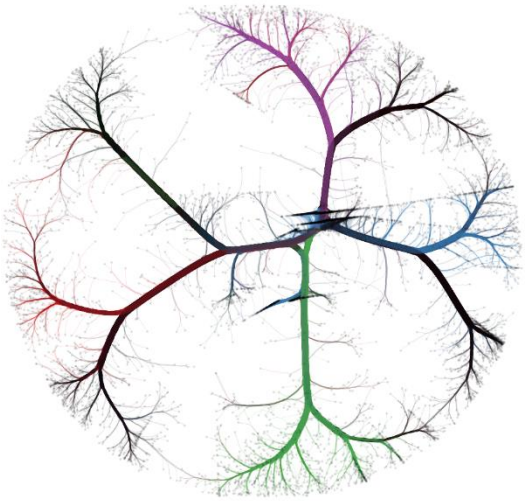


Figure 34: Paris area with the KDEB algorithm.

Figure 33 right shows the specific flow configuration with 4 incoming flows and 4 outgoing ones. The Paris Area has two main airports: Roissy Charles de Gaulle (CDG) and Orly (ORY). This visualization shows by the traffic density that Westerly departures are the least dense, and Southerly and Easterly the highest. Since Paris is in the north of France, local traffic is mainly to the south. Easterly traffic corresponds to European destinations. Specific analysis of this visualization shows that Orly, the second biggest airport in France has a strong departure flow to the south but a reduced one to the north (barely visible on the map with the arrow n). Since Orly is a domestic airport (flights remaining in France) located in in the north of France, only few aircraft head to the north.

4.7 Conclusions

In this chapter I explained how my initial investigations with density maps became the cornerstone of the developed edge bundling algorithm. This work contributes to the dense graph visualization and provided a definition for the edge bundling algorithm:

Edge bundling techniques trade clutter for overdraw by routing related edges along similar paths.

Graph visualization supports various comprehension tasks such as understanding connectivity patterns, finding frequently-taken communication paths, and assessing the overall interaction structure in relational datasets (Lee et al., 2006). Much further work is required to fully understand how edge bundling algorithms support such tasks. These extensions will be discussed in the perspectives chapter of the present document.

These works also highlight how the mole view (Hurter et al., 2011b) and dual layout animation help improve and explore bundled graphs. Interactive animation techniques are also one of my investigation topics and I will provide additional details in the following chapter.

5 Chapter
Animation as an efficient data exploration tool

In 2011, during my visit to the University of Groningen, I had the opportunity to work on a possible extension of the mole View (Hurter et al., 2011b). Alexandru Telea and I decided to extend this work by using a 3D layout rather than 2D dual structures. Alexandru had already been working quite a while on 3D modeling and 3D Direct Volume Rendering (DVR). Furthermore, I had fascinating discussions with Moritz Gerl, a PhD student at Groningen University, who was investigating tools to improve 3D DVR Scan visualization. Such a scan contains 3D voxels with a density value which corresponds to the density of the tissues: skin has a low density, bones a high density. I first managed to gather such 3D datasets, and tried to use the mole view with then. My first attemptd used pseudo color without a color transfer function, but I developed a simple prototype with a sphere on which voxels can be pushed to the edge of a semantic 3D lens (Figure 35).

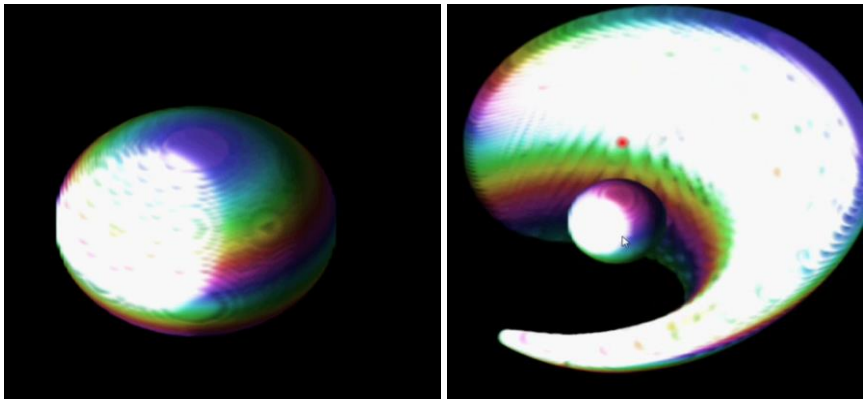


Figure 35: Point based rendering of a 3D ball with pseudo color and a semantic 3D lense

I also improved the visualization by using a point sprite technique (Pajarola et al., 2004; Sainz and Pajarola, 2004). As such, each voxel is displayed with a 2D point sprite whose border is transparent. This transparency follows a Gaussian transfer function. With a suitable transfer function one can visualize a more realistic picture (Figure 35).

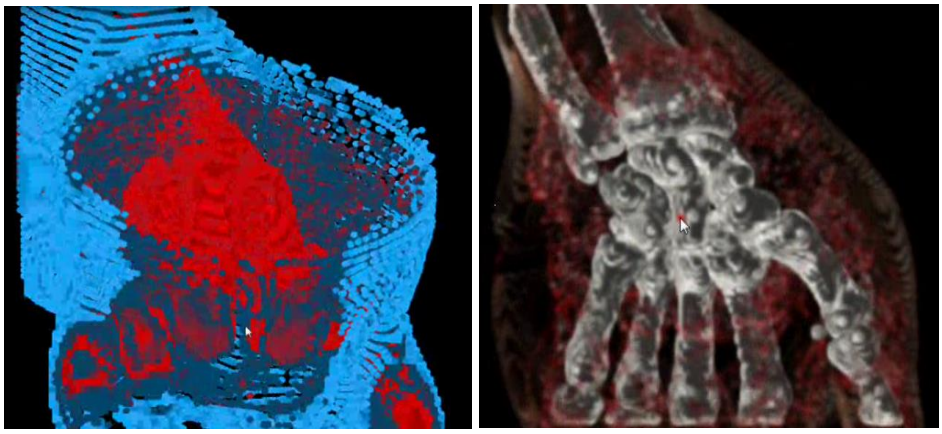


Figure 36: Visualization of a 3D scan with point based rendering and color transfer function.

This first version of the 3D mole had many technical limitations (number of voxels, limited interactions...) and thus a reduced contribution. Even if my implementation produced more aesthetic

visualizations thanks to recent improvements with a powerful graphic card, previous works have already investigated such interactive techniques to explore such 3D visualization (McGuffin et al., 2003) and (Elmqvist, 2005). My prototype could not handle more than 400000 voxels with a reasonable frame rate (10 fps). The bottleneck lies in the rendering process: the blending of point sprite is computationally challenging and the physic (movements around the 3D semantic lens) of the point sprite is also difficult. This prototype needs to update these point sprite positions every frame to ensure smooth animation. At that time, this bottleneck could not be resolved. My knowledge was too limited to find a suitable solution and to propose an actual contribution. For these reasons, this project had to stop and I focused on the other possible extension of the mole view: the dual layout.

5.1 From the Mole View to Color Tunneling: the animation as an data exploration tool

At the end of presentation at InfoVis 2011 of the Mole View (Hurter et al., 2011b), I showed a demo of the possible mole view extensions with the dual layout animation between an image and its corresponding histogram of luminosity (Figure 37).



Figure 37: First prototype of an animation between an image and its corresponding histroram.

After this presentation, Pierre Dragicevic and I started to discuss and to exchange basic ideas regarding possible usages of such visual transition. During the train trip back from Providence to Boston, Fanny Chevalier joined us and we decided to investigate together image manipulations using histograms. The development of this new prototype was relatively quick (3 months) since no GPU accelerating was provided. Fanny handled the core programming of this prototype and did a fabulous work implementing it. With Histomage, histograms are a new tool to link, select and change the color of pixels in an image (Chevalier et al., 2012). During that time I tried very hard to find a solution to the scalability issues. Histomage can handle images up to a certain size, but images more than 1 M Pixel in size cannot be developed interactively. This limitation is linked to the intrinsic histomage visualization process and suffers from the same limitation as the Mole View (Hurter et al., 2011b). Since the animation is applied to every pixel of the image, the computation time is directly linked to their number.

5.2 GPGPU usages to address scalability issues

Thanks to recent progress with the graphic card programming pipe line, I have found a solution to the scalability issue. Before presenting the solution, I will first give some GPU/GPU background. The following is mainly extracted from a book chapter I co-wrote where we discussed the scalability issue with multivariate graphs during a Dagstuhl seminar (Jankun-Kelly et al., 2014).

5.2.1 GP/GPU technique and history

In this section, I focus on how graphics processors can enable scalable visualization and I give some limitations.

GPU Pipeline, fixed vs programmable: the rise of special-purpose graphics hardware for the accelerated monochrome and color display of 2D/3D raster and vector graphics began during the mid to late 1970s and widespread consumer adoption especially of hardware 3D-acceleration solutions was obtained during the late 1990s. Such hardware was originally built around *fixed functionality pipelines* (FFPs), i.e., special purpose hardware that supports a limited and fixed set of instructions (drawing commands) to display various types of graphics primitives. Typical FFPs support operations such as geometric transformation, lighting, and rasterization, all of which are necessary for displaying and projecting 3D graphics on 2D raster displays. From the mid-2000s onwards, graphics hardware manufacturers as well as graphics API developers gradually shifted their focus to programmable pipelines instead of FFPs. *Programmable pipelines* allow the graphics processing unit (GPU) to run proprietary code (Owens et al., 2007). Such code can be used to implement new types of drawing commands and can even be used---although initially indirectly---to perform (non-graphics-related) computational tasks on a GPU, i.e., ``general purpose computation on the GPU'' or GPGPU (Thompson et al., 2002). The latter is useful because of the massive parallelism offered by GPUs as well as the ease with which GPUs generally handle vector and matrix operations; a direct result of the fact that 2D/3D transformations and projections within FFPs rely heavily on vector/matrix math.

GPU Programming---APIs and pitfalls: programming each level of the graphics card pipeline can be performed through different languages, such as NVidia's Cg, Microsoft's High-Level Shading Language (HLSL), and the OpenGL shading language (GLSL). Other specialized languages exist to do specific data processing: CUDA, OpenCL. If we exclude specific data processing languages (CUDA and OpenCL) which use specific data structures, output data must be stored in image textures. Graphics cards propose massive parallel computing but some pitfalls must be avoided in order to take advantage of this worthwhile power.

- Graphics card are optimized to compute data in parallel and therefore sequential algorithms cannot be parallelized without insuring data integrity (memory protection). Reading and writing graphics memory is not possible at the same time; this avoids memory corruption (one process reading at the same time as another is updating the information).
- Synchronization features such as mutex or memory protection (atomic functions) must be avoided as much as possible. Specific computation techniquescan be applied such as MapReduce, a programming model for processing large data sets with a parallel, distributed algorithm on a cluster (He et al., 2008).
- Bottlenecks exist within the GPU processing, especially when transferring data between the CPU and the GPU. When this occurs, the graphics card needs to wait until every process has ended, and then start the memory transfer---a dramatically slower process. Memory transfer between the GPU and the CPU must be limited as much as possible.
- Many other pitfalls must be taken into account regarding each language, such as texture coordinates that differ between OpenGL and DirectX, debugging issues, and graphics card crashes that hinder the development process.

5.2.2 Instances of GPU usages

Given the above, I will list below GPU usages to address scalability interaction and visualization. The key to these techniques is how they overcome the limitations of the GPU mentioned previously to facilitate multivariate graph exploration---they use multi pass read-write cycles, minimize CPU-GPU memory transfer, and accommodate variation in graphical hardware:

Rendering: Graphics cards can render numerous items on the screen and thus can display large datasets. In the following examples, GPUs are used to display data and to perform image based rendering techniques. Auber developed Tulip (Auber, 2004), an information visualization framework dedicated to the analysis and visualization of relational data. This software uses GP-GPU techniques to render large multivariate graphs. McDonnel (McDonnel and Elmqvist, 2009) developed a framework and an application using shaders to display multivariate data based on the dataflow model with a final image based stage. In this final step, the multivariate data of the visualization are sampled in the resolution of the current view. A more specific rendering technique is used by Holten (Holten, 2006) to improve edge visualization by an interesting variation on standard alpha blending, i.e. how color transparency is combined. Scheepens (Scheepens et al., 2011) used the GPU to compute density maps and then apply shading techniques to emphasize multivariate data on the density map of moving vessels.

Computation: Graphics cards can perform fast and parallel data processing, and can be used to process information at the data level. As previously explained, FromDaDy (Hurter et al., 2009b) uses the GPU for interactive exploration of multivariate relational data. Given a spatial embedding of the data, in terms of a scatter plot or graph layout, the Mole View (Hurter et al., 2011b) uses a semantic lens which selects a specific spatial and attribute-related data range. The lens keeps the selected data in focus unchanged and continuously deforms the data out of the selection range in order to maintain the context around the focus. Animation is also performed between the bundled and the unbundled layout of a graph. Kernel Density Edge Bundling (KDEB) (C. Hurter et al., 2012) computes bundled layouts of general graphs. This technique is also applied on dynamic graphs (Hurter et al., 2013a). Other GPU bundling techniques also exist; Winding roads uses a voronoï diagram to compute Graph bundling and its density (A. Lambert et al., 2010). Finally, the GPU has been used directly for graph layout as well (Frishman and Tal, 2007).

Interaction: Interaction with the data is an important manipulation paradigm to perform data exploration. Graphics cards can be used to provide tools to help users to interact with large datasets. Rolling the dice (Elmqvist et al., 2008) helps the user to define the appropriated displayed variables with a smooth animation when changing visual configuration; Graphdice (Bezerianos et al., 2010) uses the same paradigms but with graphs. FromDaDy (Hurter et al., 2009b) uses related animation with GP-GPU techniques. In order to address the dataset size issue, FromDaDy loads the whole dataset within the graphics card, so that when changing visual configuration, no memory transfer is needed. This helps to improve interaction with fast and continuous animations. Furthermore, a GP-GPU technique is implemented to support brushing and data manipulation across multiple views. The user can then brush trajectories, and with a pick and drop operation he or she can spread the brushed information across views. This interaction can be repeated to extract a set of relevant data, thus formulating complex queries. Each trajectory has a unique identifier. A texture (stored in the graphics card) contains the Boolean selection value of each trajectory. When the trajectory is brushed, its value is set to true. The graphics card uses parallel rendering which prevents reading and writing in the same texture in a single pass. Therefore FromDaDy uses a two-step rendering process: firstly it tests the intersection of the brushing shape and the point to be rendered to update the selected identifier texture, and, secondly, it draws all the points with their corresponding selected attribute (gray color if selected, visual configuration color otherwise).

5.3 Color Tunneling: a scalable solution to large dataset manipulation with image based interaction

Thanks to GPGPU techniques, the scalability issues when visualizing and interacting with large datasets can be addressed. In order to extend the Mole View and Histomage techniques (Chevalier et al., 2012; Hurter et al., 2011b), I directly took advantage of one of the most advanced GPU usages (at that time): the feedback render buffer. This new type of shader technique (programmable graphic pipeline) allows one to store in GPU memories the output of the geometry shader. Usually, the programmable pipeline modifies the geometry of vertexes and uses the raster map to display it. This computation needs to be performed at every frame. The rendered back buffer allows the storing of vertex modifications and thus optimizes the computation of vertexes geometry. This process is mainly used to compute the location of particles in particles systems, but other usages are possible. Since the Mole View uses point sprites (close to particle system), I managed to speed up the geometry computation between two consecutive frames. The investigation of such a process took almost one month and I finalized the first prototype in 2012 with a GPU version of Histomage animation. Such a prototype could animate up to 20M pixel with a frame rate of more than 20 fps on a 480 GTX nvidia card. This technological improvement was the missing factor to address the scalability issue with the 3D moleView extension.

During the Dagstuhl seminar “Putting Data on the Map”, I had a long discussion with Sheelagh Carpendale who had intensively worked on lenses and data distortion (Carpendale et al., 1997). She invited me for 2 months to Calgary where I spent most of my time finalizing this prototype. During that visit, I met Dr. Russ Taylor who had investigated a 3D cube of astrophysical measurements of the large-scale structure of hydrogen gas intensities in the Milky Way Galaxy (Taylor et al., 2003). The investigation of this dataset was very helpful (Figure 38) to finalize color tunneling (Hurter et al., 2014b).

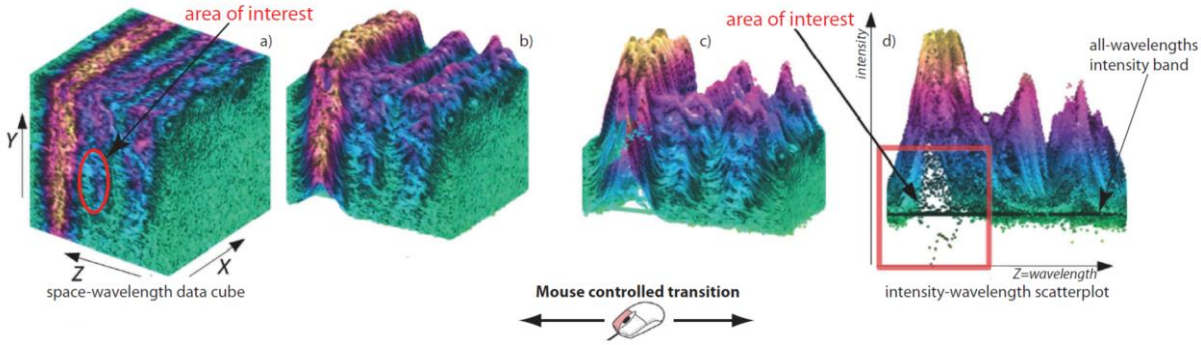


Figure 38: Color Tunneling (Hurter et al., 2014b), finding intensity outliers with isolated ranges in an astronomical data cube (Taylor et al., 2003).

During my stay in Calgary I also had the opportunity to work with John Brosz, Miguel Nacenta and Ricky Pusch on their project Transmogrification. This project was about developing a multi touch interactive system to deform visualization and thus leverage interactive data visualization and introduce the casual InfoVis. It is termed “casual” as the user is the one who transforms the view to find a better data visualization. As such, one can take the visualization of one’s heart beat (time series) and make it still on its path displayed on Google map (Figure 39). This is where the casual InfoVis intervenes where the user has the power to transmogrify data representation to suit his or her will. When I joined this project, the system was already working with an advanced prototype, but I found it difficult to fully understand the data transformation. The current version did not provide animation between the

transformation steps. Therefore I exported the animation principle from Color tunneling and Ricky Push integrated it into the working prototype (Brosz et al., 2013).

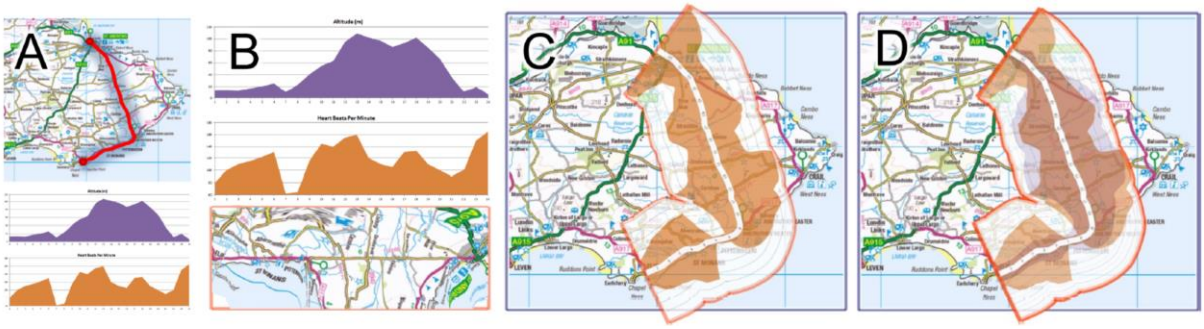


Figure 39: A) Cycling data: a map with a route and area graphs with average altitude (purple) and heart-rate (orange) at each kilometer. B) The rectified route map aligned under the linear graphs enables comparison of the measured variables to the map features. C) The heart-rate graph wrapped around the route in the map shows effort in spatial context. D) Same as C, but with multiple variables. Map used contains Ordnance Survey data c Crown

5.4 Conclusions

In this chapter I explained how the simple idea of using every pixel as a discrete item became a major challenge in my research. The extension of the Mole View was cumbersome and it took me almost three years to find the technical solution. I was also very lucky to encounter people who already had part of the solution. Animations are featuring alongside existing visualization software and are becoming a standard design feature. As an example, I can cite the recent work with graph and matrix exploration (Bach et al., 2014a) (Bach et al., 2014b). With Maxime Cordeil, my first PhD student, we managed to show how animation can display relevant information (Cordeil et al., 2013). We identified three expected benefits of such animation: tracking graphical marks, understanding their relative arrangements, and perceiving structural elements. We studied existing implementations of progressive 3D rotation (Elmqvist et al., 2008) and found problems that can reduce those benefits when dealing with dense scenes.

Color Tunneling takes full advantage of animations and we showed how it can be an efficient tool to dig into the dataset. Even if we provided many instances of concrete data exploration success, we still lack a proper evaluation to fully assess the power of animations to support data exploration.

As a conclusion, my previous investigations show how my research project is at the crossroads of Information Visualization, Visual Analytics, Computer Graphics and Human Computer Interaction (HCI). In the following chapter I will detail my HCI investigations with Air Traffic Controllers while investigating prospective interactions to support their activity.

6 Chapter
Strip'TIC: Striping for tangible interface for controllers

Strip'TIC (Stripping Tangible Interface for Controllers), is a system that has been developed to support air traffic controllers' activity with augmented paper and digital pen, vision-based tracking and augmented rear and front projection. In this section, I will detail the design process that lead our team toward the development of such a system.

The Strip'TIC story started in 2010 at CHI Atlanta, the premier international conference of Human-Computer Interface. At that time, I was about to defend my PhD but I was also working on developing tools to improve air traffic controller activities. I knew the work Wendy MacKay had done regarding the Chameleon system (MacKay, 1999). Mackay and Medini spent more than one year studying Air Traffic Control (ATC) activity and with the research center for French civil aviation (CENA[4], Athis-Mons, France) they developed a prototype called Chameleon to study possible usages of innovative technology to improve ATC activity. Following that, a member of the CENA developed Digistrip (Mertz et al., 2000), an electronic stripping system which uses an electronic metaphor of the paper strip to support ATC activity.

Conferences like CHI are the perfect opportunity to discover what researchers have managed to do with brand new available technologies. Even if the digital pen was not new, I discovered its usages with MouseLight (Song et al., 2010). I met Hyunyoung Song, PhD student, the first author of this paper and we had a very interesting discussion around the technical issues when using Anoto pens[5].

On the 16th April 2010, the Eyjafjallajokull volcano eruption transformed Western Europe and Scandinavia into an unprecedented no-fly zone. As a result I had to spend 5 extra days in Atlanta. Stéphane Conversy and I took advantage of this free time to travel to Tybee Island and Savannah to practice Kite surfing, our favorite sport. During the trip, we started a discussion around this Anoto pen idea for Air Traffic Controllers while we were eating slices of pizza (optimizing our lunch time to travel). An hectic brainstorming session started with many ideas and technology features to be tested. At that precise moment Strip'TIC was born, with the core ideas annotated on this pizza box. The first and the strongest ideas were to make the system aware of the handing information on the paper strip and to give feedback to the user.

Back in France, I contacted the Anoto Company to buy the devices. There was a long delay with grant agreements and Intellectual property issues and finally after 3 months I received a test set of 5 pens. The software development was very fast lasting just a few weeks and I ended up with a simple prototype where the pen could point at a paper strip and the system could tell the user name of the corresponding strip and highlight it on the radar screen.

I knew that a lot of work remained to be done, and I decided to start a student project to study ATC activity and to find areas that might be suitable for interaction with a digital pen. 4 students worked for 6 months on this project and we managed to publish the first Strip'TIC paper at the French

[4] http://fr.wikipedia.org/wiki/Centre_d'%C3%A9tudes_de_la_navigation_a%C3%A9rienne
[5] www.anoto.com

conference on human computer interaction (Hurter et al., 2011a). This initial version only contained a radar screen and paper strips (Figure 40).

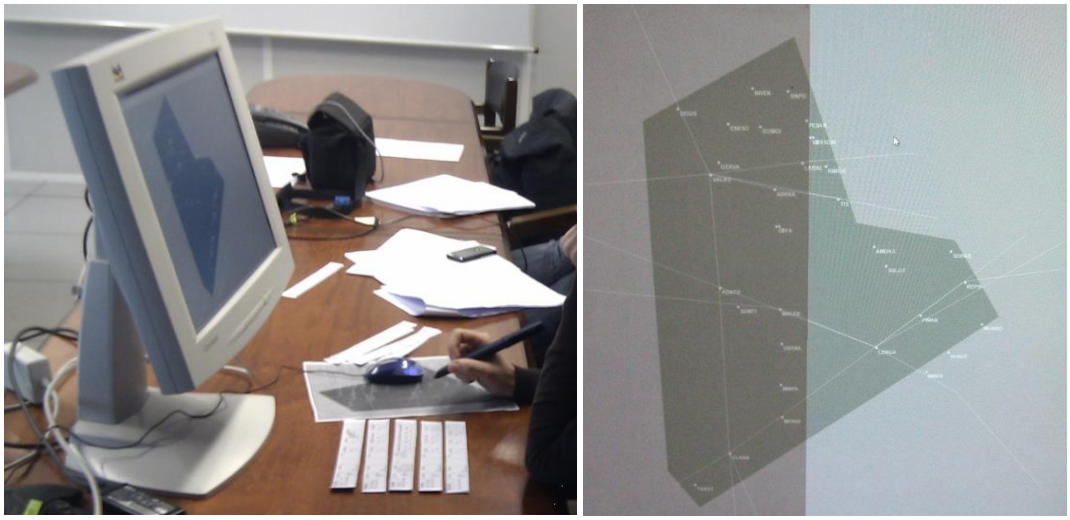


Figure 40: very first paper based prototype. Radar screen with and without the Anoto dot layer (right).

This initial work highlighted the need to improve the prototype with additional feedback. The user was not able to tell if his or her actions with the paper strip were correctly integrated by the system. We first used an audio feedback but this was a poor solution with too many audio notifications which spoiled the user's activity. During the trip to Savannah, we had thought of tracking the position of the paper strips with AR tags on their back in order to project video on top of them. During the summer I worked with Rémi Lesbordes, air traffic controller and François Régis Collin, engineer, to setup a strip tracking system. The main technological issue was the detection of the paper strip location. Therefore we spent more than one month testing different technological and physical setups. We finally managed to achieve suitable paper strip detection thanks to the ARToolkit[6] and a specific image processing algorithm (Figure 41).



Figure 41: Strip tracking system with bottom and top projection.

In order to improve usability and to capture user actions (hand writing, pointing), we thought of pointing at the radar screen and the strip board with the digital pen. We faced some issues with the pen detection. Even if the luminosity is reduced, we used a simple technique with a fine infrared reflective plastic layer (Hofer and Kunz, 2010) (Figure 41). Finally, we added a bottom beamer and summarized our investigation in a technical paper (Christophe Hurter et al., 2012) (Figure 42). Stip'TIC is not a monolithic system, but a set of interconnected modules. I developed a simple radar screen, the bottom and top projection and the supervision; I used an air traffic simulator called Rejeu, and the

[6] http://www.hitl.washington.edu/artoolkit/

middle ware Ivy (Buisson et al., 2002) to allow communication between modules. Additional modules can easily be developed to support additional features.

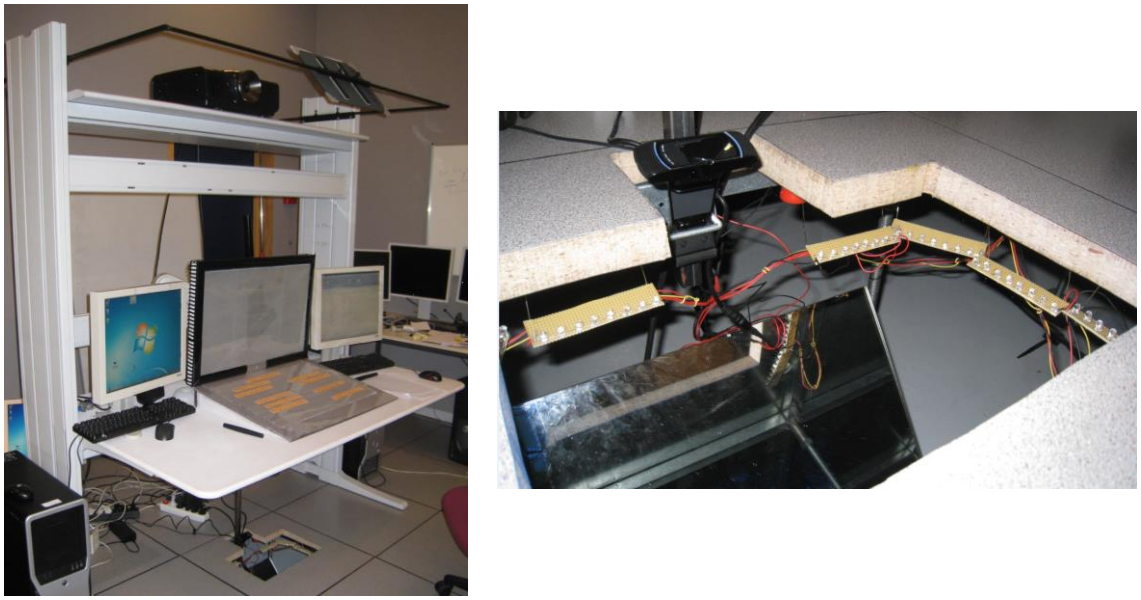


Figure 42: Radar screen with and without the Anoto dot layer.

The next steps were technological improvements such as avoiding digging into the floor to setup the back beamer (lens projection improvements), design improvements and additional ATC features developments. 4 Masters students worked on this project to assess its usages for air traffic controller in a control tower. Strip'TIC helped us to reflect upon the design of a paper-based tangible interactive space to support air traffic control.(Letondal et al., 2013) (Vinot et al., 2014). Cheryl Savery, PhD student from Queen's University in Canada, spent 3 mounts in our lab to develop and assess multi touch features with Strip'TIC (Savery et al., 2013). More recently 4 more Masters students have worked on this prototype to assess gesture integration into the system.

6.1 Strip'TIC and image based techniques

The Strip'TIC project has helped to better understand and to investigate ATC activity. It has contributed to the HCI community with augmented reality, tangibility and collaboration. To make this system work, I had to address many technical challenges. Even if none of them specifically contributes to the computer graphic domain, I had to use some image based techniques.

The first one was to handle the system calibration process in a simple manner to correct lens deformation when projecting on the top and on the bottom of the strip board. To this end, I used a simple homography computation (scaling and rotation matrix). Thanks to the graphic card, the projected image is first computed off line (in an object buffer) and then mapped as a texture on a grid to support the homography computation. This simple process avoids multiple projection matrix computation, and only the graphic can compute it when mapping the texture on a distorted grid.

I also developed a simple image based algorithm to improve AR toolkit pattern detection. Strip'Tic uses a complex environment with many light sources, and the complex arrangements are of both the plastic layer and material. To track the paper strip, a webcam captures the image below the stripboard. The luminosity there is not uniform, which hinders the ARtoolkit process. With a simple noise and gradient scale removal, I managed to find a way to greatly improve pattern detection. At that time, I could not find an easy way to process the images, so I performed the computation in C++ with an image

processing algorithm (Christophe Hurter et al., 2012). Today this computation can be easily managed with API like OpenCV, and we could greatly reduce the computation time devoted to pattern detection.

6.2 Conclusion

Strip'TIC does not directly take part in my image based research, even if it embeds image processing techniques. This project is the only one where I have tried to explore in detail the Air Traffic Control activity with a Human Centered Design process. I spent countless hours discussing with controllers, studying their activity, trying to understand design requirements and exploring innovative tools. The Strip'TIC has been my main research project for almost two years. This system is nowadays a suitable tool to support innovation in ATC and can also be used for more practical activities like evaluations. My next researches has stepped away from this project to concentrate my efforts on an image based algorithm devoted to data exploration. Strip'TIC is a concrete example of achievement based on people's willingness, outside of any planned project with proper funding. This pluridisciplinary project gathered many researchers, engineers and students and was an unique occasion to work together without temporal constraints but with reduced funding.

7 Chapter
Research program

During my research I have investigated visual designs, interactive techniques, methods and algorithms to outperform existing methods and more especially data manipulation and knowledge discovery. My work has been inspired by exchanges and interactions with other researchers, engineers and application specialists mostly in the aeronautical field. My research is built upon technological improvements and advances in graphic card powers and is directly linked to the Ben Shneiderman mantra « overview first » (Shneiderman, 1996). When I started my PhD thesis, I felt frustrated not being able to manipulate or even display one day of a recorded aircraft trajectory dataset with a reasonable frame rate. This is the reason why I developed FromDaDy (Hurter et al., 2009b), MoleView (Hurter et al., 2011b), and Color Tunneling (Hurter et al., 2014b). Today visualization of a large dataset is still a challenge and the bottleneck lies in the available pixels on the screen which are not numerous enough to visualize every piece of information in large datasets. Therefore, I investigated aggregation techniques to simplify the visualization of trails and graphs with edge bundling techniques. In the meantime, I developed various interactive techniques to manipulate and to link data thanks to image based algorithms. Smooth animations and continuous space interactions are the cornerstone of seamless data exploration techniques and decision making.

My contribution is to provide interactive scalable tools that are built upon image based algorithms. These techniques provide a wide area of investigation where much still remains to be done. I am only scratching its surface by picking some application domains where these algorithms can improve users' activities. In the following, I will discuss their assets, their limitations and give my plan for future usages.

7.1 Computed graphic and raster data

In this section I report on evidence of image based usages. I will take the example of the physics of light model for computer graphic rendering, followed by identified usages in data manipulation and exploration.

The physics of light is a rendering process with modern graphic cards. SIGGRAPH is the premier international forum for disseminating new studies in computer graphics and interactive techniques. In this conference many papers deal with the rendering process of a 3D scene with computer graphics. Among the rendering technical challenges, the computation of illuminations and shadows is still an active investigation area. The ray casting (Roth, 1982) technique relies on the reverse propagation of light. This technique can take into account reflection, distraction and absorption of material and produces realistic rendering. This is especially true with soft shadow computation, optical lens distortions which can be easily computed thanks to the physical modification of the light path. The ray casting algorithm is a simplified model of light propagation which omits the fact that every object in a 3D scene can reflect a given amount of light in every direction and thus be another light source. This simplification can prevent the illumination of object when no direct light source is applied and thus produces areas without any shade. To address this issue more complex techniques have been developed such as the defining rendering equation (Kajiya, 1986) with uses light emission in many directions. This technique is far more computationally challenging compared to ray casting, even with

some simplification and optimization (Purcell et al., 2002). Nevertheless the rendering results are high quality images and are among the current techniques to produce theater movies.

However, these techniques have a drawback: their computation time prevents rendering in real-time even with modern graphic cards. Therefore other simplification models have been developed and especially one that relies on rasterisation and texturing. Textures are composed of juxtaposed colored pixels which form a rectanglar shape. Texture can be applied to a 3D object and thus map specific shading or lightning features. These textures are the cornerstone of the simplified light rendering process. Instead of computing the complex physique of light, the rendering space is discretized and restricted to textures. A light model like Phong shading (Phong, 1975) is applied to vertexes (and then interpolated) or by pixels (fragment shader). Shades are computed by changing the point of view and then projecting the result on a texture(Engel, 2006). An environment map (Heidrich and Seidel, 1998) is computed and applied to a 3D object to simulate reflection. Many of these textures or pixel based techniques are currently used to produce fast rendering in video games. The computation process consists of a succession of texture rendering and compositions with raster map processing.

These raster usages produce a drastic simplification with much inaccurate or even incorrect rendering but are today essential rendering techniques. Fast, real-time rendering is achievable with rasterisation techniques, and we still rely heavily on them.

Data exploration and manipulation with image based techniques.

Taking into account that computer displays are actual raster screens composed of juxtaposed color pixels, an image based technique naturally fits their rendering process. Even if many image based techniques do exist, few of them are used to perform data exploration. Color tunneling is one such system, where every piece of data is mapped to a discrete interactive item.

Recent advances in discretization and rasterisation have introduced point cloud visualizations (Rusinkiewicz and Levoy, 2000). Rather than displaying texture surfaces, the point cloud technique uses each pixel as an individual object (i.e. color Tunneling (Hurter et al., 2014b)). These techniques arose with the development of 3D scanners and the management of large quantities of 3D pixels.

These techniques allow flexible data deformation and visualization. Data deformation can be performed with image based techniques such as SBEB (Ersoy et al., 2011) and KDEEB (C. Hurter et al., 2012) to simplify graph and trail visualization. These recent techniques have also proven to be usable and scalable even if they rely on a significant data space simplification: data rasterisation.

7.2 Raster data inaccuracy

Raster data is less accurate than continuous forms; in computer graphics, floating values are more accurate than integers. Using this data simplification benefits computation time, but hinders accuracy. Now the question is to assess if this gain of performance worth this data inaccuracy.

As previously explained, video games benefit from such a simplification and provide high visual quality video gaming with an interactive frame rate. If we take into account the KDEEB algorithm, small definition accumulation maps can perform bundling, but we can easily see these discretization artifacts.



Figure 43: Raster map size effects. Original graph (left), bundled version with a small raster map (middle) and with a large raster map (right).

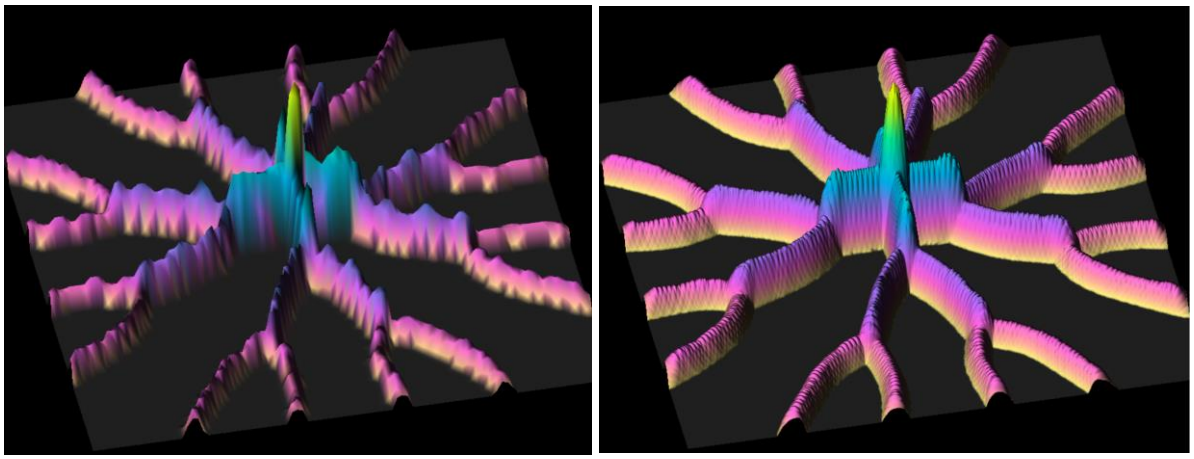


Figure 44: small size density map (left), large size density map (right).

We also use very high quality accumulation maps where the produced results are very smooth, but a lot of computation time is wasted by computing data from very similar locations. Parameters settings are open questions, and I do not have an easy way to compute the size of the accumulation map. Image based bundling techniques show an interesting use case where drastically reduced accuracy (accumulation map of 100 pixels width) can produce exploitable bundling results. It is not strictly true that this algorithm is only pixel based, since accumulation and data density are the solely pixel based techniques, and once the computation of the density is completed, computation is performed in a double floating accuracy with the computation of the gradient value. Then the pixel based/ rasterisation is applied, since vertex will move according to the raster map, which also produces artefact, and then again we switch into double accuracy by smoothing the result. This example showed how the high frequency data distortion can be reduced by using low level filtering. To summarize, low accuracy produces high frequency artefact that can be easily filtered out and this is still worthwhile if computational speed is a requested feature.

7.3 Technical challenges

My work has tried to prove the efficiency of such Image based techniques: they can provide a scalable solution and can also rely on technological advances such as faster and larger available memory for storage. The current bottleneck regarding GPGPU computation is the memory transfer between the GPU and the CPU, but we can forecast that future architecture improvement will address this issue.

In terms of technological improvements, one can expect a simpler way to program these powerful GPU processors, by providing high level API, and more advanced instruction. The major flaw today is the

lack of easy debugging tools. Such a GPU program, especially with shaders, needs special care to debug. It appends less frequently, but the system can lock. There is no easy way to investigate memory value and the debugging of high parallel threads is challenging even in a CPU and particularly so in the GPU.

Another interesting area of investigation lies in the use of massive memory and one can envisage memory as powerful as that of computation. This investigation needs to be validated, but since memory is cheap and available in large quantity, it is a valuable resource. For instance, multithreaded computers are often in the idle state. We can take advantage of this waiting period to process information and store the result for future usages. This optimization process of available resources is a well-known technique, but cannot be generalized.

7.4 Personal image based road map

In this section I will provide research investigation areas with image based techniques. These extensions are part of my ongoing project but also addresses identified research questions, technical challenges which remain to be addressed.

Image based techniques can be applied in many research areas, but I will emphasize some on them. I will especially focus on an existing technique which uses raster data or algorithms and thus will directly benefit from these image based techniques.

Bundling techniques are powerful tools for graph simplification. Bundlings have shown their ability to visually aggregate the links between nodes and thus to produce empty spaces to improve a graph's global readability. This aggregation helps to retrieve information, like highlighting main flow, and is often associated with interactive methods to improve this process.

Bundling techniques are useful for visual simplification of graphs and address issues regarding data density. Recent bundling techniques use graphic card power to support interactive visualization (OpenGL /WebGL /DirectX) or their ability to perform parallel computation (OpenCL / CUDA). The latest bundling techniques use image based technique to reduce computation time with a complexity close to O(Edges).

The aesthetic criteria is undeniable with the production of smooth curves and the addition of image processing such as bump mapping (table 2). These aesthetic criteria have never been evaluated in terms of an asset to investigate datasets and it remains an open question.

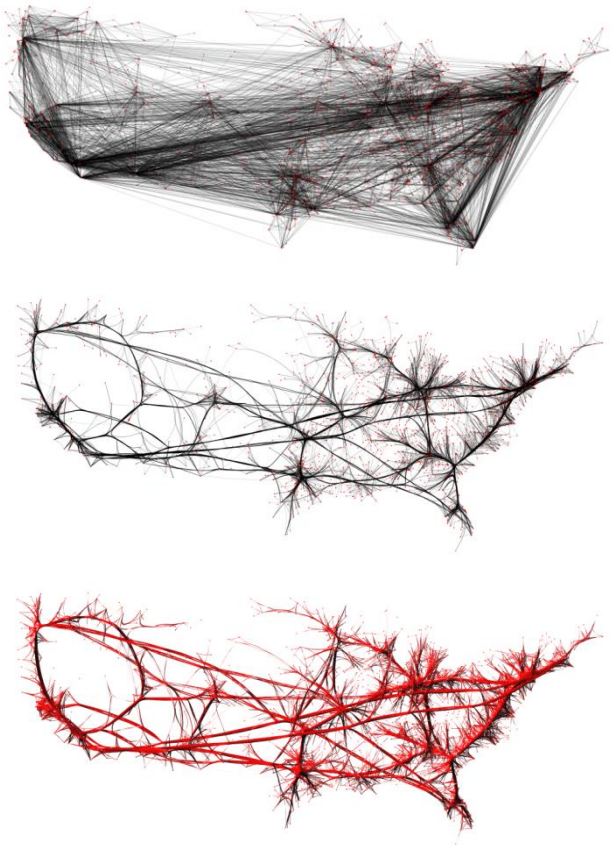


Figure 45: bundling, original US migration grap (top), KDEEB bundled version (middle), with bump mapping (bottom).

Nevertheless bundling techniques are not an all-purpose tool, and their ability to simplify visual graphs has many limitations. The following sections list bundling technique open questions that have not yet been investigated.

7.4.1 Tasks

There is no available task taxonomy that can help to assign specific bundling methods to a given task. For instance, if one is interested in investigating a small world graph (Auber et al., 2003) (van Ham and Wattenberg, 2008), which bundling method will provide the best results?

Taking into account the taxonomy of graph tasks (Amar et al., 2005)(Lee et al., 2006), a bundling algorithm only allows the following one:

- follow a link or a path , but only under certain conditions with reduced density and with interactive tools (e.g mole View),
- see an attribute (only the density) of a link.

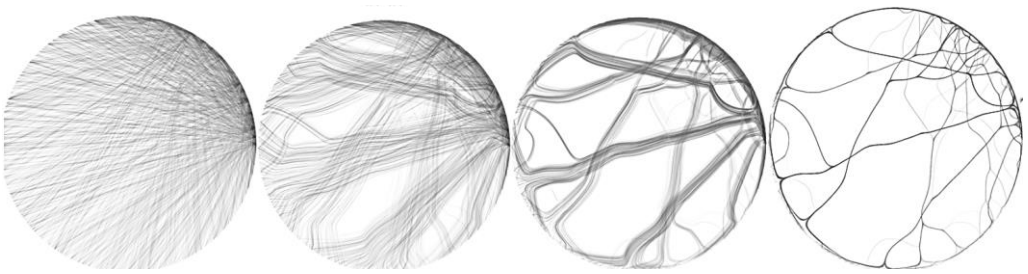


Figure 46: interactive tools to explore the dual layout of a graph(bundled and unbundled)

Bundling is a tool for the exploration of graphs at the macroscopic level and will not allow you to find information that concerns only a small number of links. For example, the following tasks are not fully supported by the bundling algorithm:

- finding an outlier (e.g. a single link),
- Ordering the nodes by their number of connections.

7.4.2 Dynamic graphs

Dynamic graphs have not yet been intensely investigated. They need fast bundling computation to ensure that the bundle shows the up to date graph layout. Since a graph evolves over time, their bundled version must be updated without abrupt changes. Only KDEEB (C. Hurter et al., 2012) is able to offer a solution in this area. KDEEB has proved its effectiveness with a computation time of only a few milliseconds of calculation and ensures a continuous evolution of the graph layout thanks to the Mean shift algorithm (Comaniciu and Meer, 2002). FDED (Holten and van Wijk, 2009) cannot ensure such continuity and the graph bundle can then show abrupt changes of structure (Nguyen et al., 2013a). Nevertheless, the lack of dynamic bundle exploitation tools remains and today there is no effective method provided. One can only watch the animation to interpret it.

7.4.3 Algorithm setting

My experience in the use of bundling algorithms has shown that the bundled end result is highly dependent on the values of algorithm parameters. This is particularly the case with the pre -processing and clustering parameters that will strongly influence the final result. The bundled graph can thus be very highly or partially aggregated, and it is difficult to determine the correct settings without extensive testing.

Furthermore, bundling parameters are complex and linked to the algorithm. To some extent, these parameters should not be related to the bundling algorithm, but to the task the user wants to perform. For instance, one can adjust the ink ratio and the algorithm produces the corresponding bundling result.

7.4.4 Bundling faithfulness and accuracy

No previous work has investigated the accuracy of the displayed bundled information. As such, no metrics have been provided to assess the bundling algorithm quality. This limitation hinders their usages and the validation of new algorithms.

Finally and most controversially we can consider the faithfulness of the bundled results (Nguyen et al., 2013b). In this sense, there is no guarantee that the bundle version is representing information that is actually embedded in the data sets. For example, the processing of a random graph with KDEEB will highlight aggregates links. This can be partially explained, because it is not possible to provide a uniform density graph link (blurring of the edges of the screen, Figure 47), but still shapes emerge.

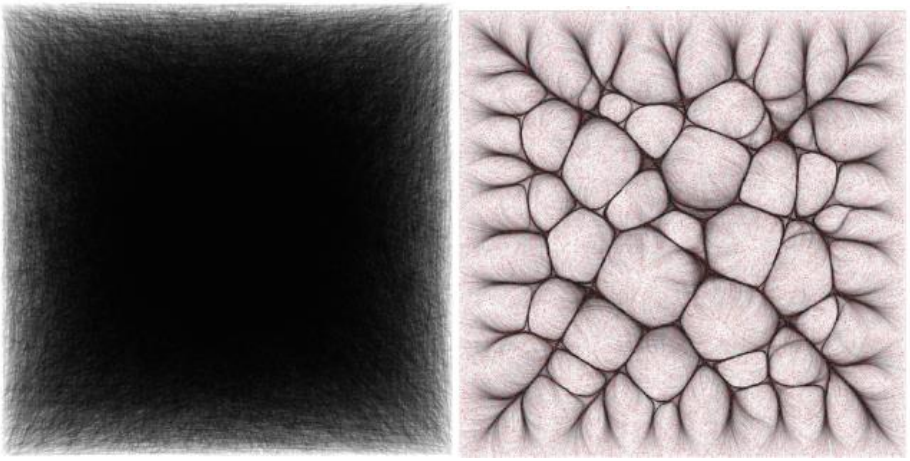


Figure 47: Graph pseudorandom and bundle version KDEEB (C. Hurter et al., 2012)

7.5 Proposal to improve bundling techniques

These proposals are the result of my experience with the use of bundling. The following list is extracted from the previous section (Personal image based road map).

Provide or evaluate metric bundle links: Today, there is no metric to quantify the result of a bundling algorithm. The only metric available is the ration ink/background proposed by Tufte (Tufte, 1986), which far from fully qualifies the result of an algorithm. Figure 48 shows examples of bundling results with very different visual renderings on the same datasets.

Provide reference data sets: each new implementation of a bundling algorithm uses new datasets which makes their comparison difficult. Mingle (Gansner et al., 2011) used sets of reference data (Davis and Hu, 2011) but provided no indication of any information to be extracted. It would therefore be useful to collect a set of validated data sets with ground truth data such as reference calculation speed, rendering quality and data mining references.

Link bundling algorithms to tasks: To improve Edge bundling usage, one should be able to choose suitable technique which match best the requested graph simplification tasks. This is a difficult problem since each algorithm has technical constraints that are not correlated to specific tasks. For example, hierarchical data is mandatory to apply HEB (Holten, 2006), data clustering must be applied before processing them with SBEB (Ersoy et al., 2011), and FDEB (Holten and van Wijk, 2009) needs rules of proximity.

Improve the parameterization of algorithms: each bundling algorithm has its own complexity with a different setof parameters. HEB (Holten, 2006) is by far the simplest, but also the most constrained with the need for a hierarchical data structure. Mingle is by far the most complex to implement with the use of the GPU to export a dependency graph. All these algorithms use many parameters for their computation (except HEB that uses only the B- Spline parameter). These parameters are often too abstract and related to the technical constraints of the algorithm. For example, KDEEB (C. Hurter et al., 2012) uses the size in pixels of a density map that has no direct connection with the exploration task of a graph. Finally, each bundling technique should be provided with a set of reference data sets and their set of optimal parameters. For example, KDEEB uses multiple bundling iterations with changing parameters.

Directional graphs: Edge bundling techniques help to reduce visual clutter of graphs. Many algorithms exist but seldom treat directed graphs. When dealing with dense and occluded directed graphs, edges with different directions should not be bundled together. It would lead to information loss and is an important issue in all applications where the exploration task(s) needs to take directionality into account, such as trajectory exploration. Most existing techniques cannot take edge direction into account without adding extra algorithm complexity. Examples of such graphs are trails or eye movement recordings. The most common visualization metaphor for directed graphs is node-link diagrams but the graphs often contain many hundreds and thousands of edges that inevitably lead to visual clutter. This research area is promising and could possibly be addressed with image based bundling methods (e.g. extension of KDEEB) and visualization methods (e.g. particle system).

Multidimensional bundling algorithms: There are still many opportunities to develop new bundling algorithms. No bundling algorithm can handle multidimensional bundling. Only a few techniques have been extended to take into account 3D (Antoine Lambert et al., 2010). FDEB is flexible enough for bundled multidimensional datasets, but with a prohibitive computation time. For example, the bundling of eye traces could be greatly improved if we take into account their temporality and direction: " I want two links to be attracted if they are close geographically, temporally and in the same direction". This problem remains an active and promising research area.

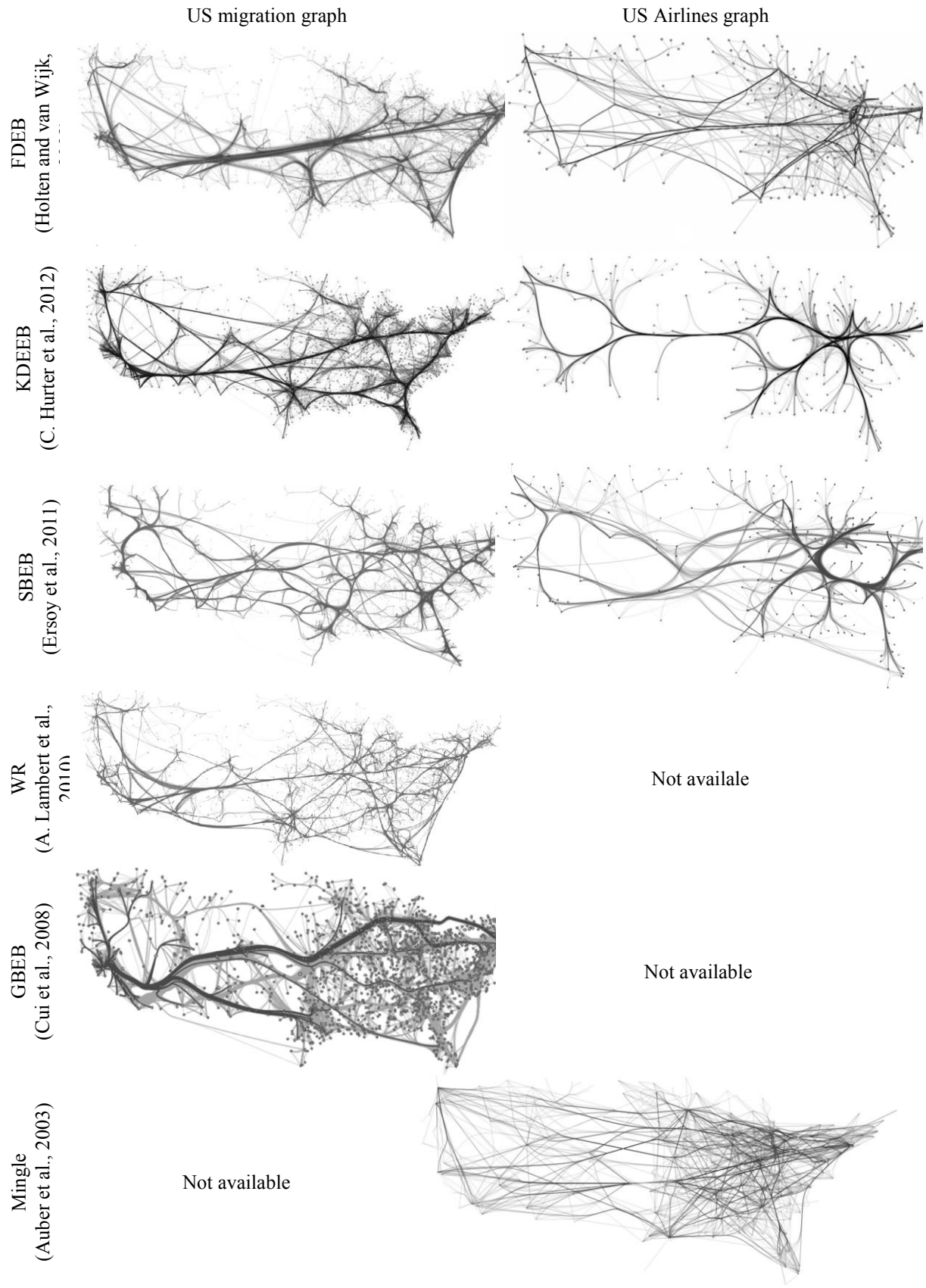


Figure 48 : examples of bundling results with very different visual renderings on the same datasets.

7.5.1 Particle system

The particle system uses the physique of particles with a life time, a position, velocity and acceleration (Blaas et al., 2009). These techniques help to show particle directions, and provide a synthetic view of global directions. I have already investigated the visualization of aircraft trajectories with such a technique. The animations produced reveal the direction of airways, but also provide an interesting visual technique which is overlapping resistant. In certain conditions, one can still identify opposite direction flow even if they fully overlap Figure 49.

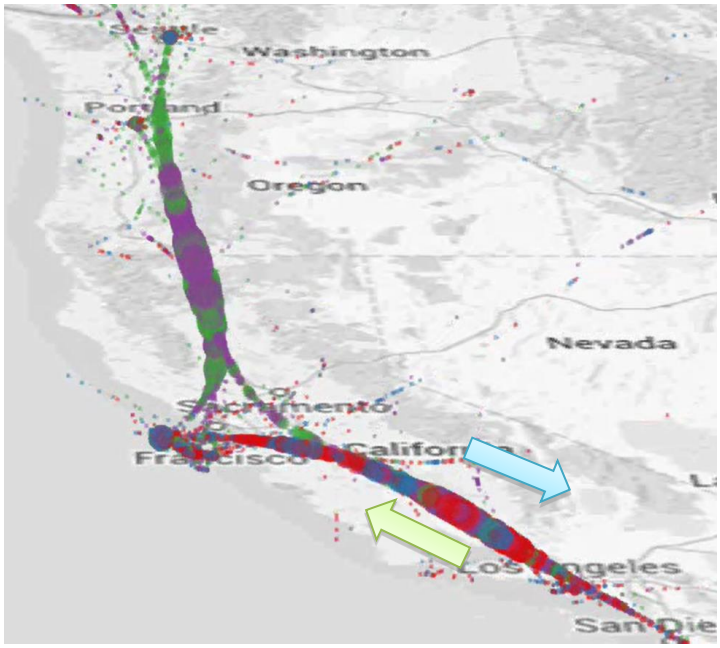


Figure 49: directional graph visualization with a particle system. Particles can overlap but one can still visualize their direction

To make such a visualization technique effective many interactions and technical challenges need to be addressed:

- Particle systems need a large amount of displayed entities which can hinder the interactive frame rate. Currently without specific optimization a few million items can be displayed,
- New interactive tools must be implemented and validated, such as brushing, data deformation and time navigation,
- Particle system assets must be clearly identified and validated with controlled experimentation.

Among the possible usages, investigation of eye tracking data is promising, but a design study must be performed beforehand to identify why and how particle systems can outperform existing visualization techniques. One of the major challenges is to prove that animation, which is an intrinsic part of a particle system, can speed-up the data retrieval process. One needs to watch the particle animation before being able to extract their direction. This question has already been investigated (Shanmugasundaram et al., 2007) but not with this visualization technique.

7.6 Image based algorithm in application domains

In this section, I detail project perspectives where I plan to use image based algorithms to support specific activities. These projects are an extension of my previous works but applied to different domains. This will give me the opportunity to answer research questions with respect to concrete

examples (internal validity). For instance, I will investigate new metrics to qualify bundles and to use the results to analyze cognitive maps. Finally, application domains will also provide a new research opportunity with new open questions (external validity). For instance the exploration and the visualization of a large point cloud remains a technical challenge. Air traffic analysis is also an inexhaustible source of inspiration.

7.6.1 Cognitive maps

In the following, I will report on an unexpected usage of edge bundling technique with the Memory project. This project is about to start in September 2014 with the collaboration of neuroscientists and computer graphics experts.

According to a ministerial review of 2004, approximately 860 000 people are affected by Alzheimer's disease in France. This estimation will possibly reach 1.3 million in 2020 and 2.1 million in 2040. This is why the study of Alzheimer's disease has been identified as a major society challenge. In The PAQUID cohort study, 3777 elderly people performed a lexical evocation task by orally producing the most possible city names within 3 minutes. They were longitudinally followed during a 22 year period and some of them developed the dementia of Alzheimer (Rondeau et al., 2009), (Tessier et al., 2001), (Nejjari et al., 1997). Among the different tasks and questionnaires reported in this study, we will especially investigate the lexical evocation task. This task is directly related to the concept of the cognitive map introduced in 1948 by Edward Tolman in his article «Cognitive maps in rats and men» and echoed by O'Keefe and Nadel in their book «the hippocampus as a cognitive map» published in 1978. The cognitive map is considered, among other things, as a sort of matrix in which the episodes of life can be stored. Formally, the cognitive map is "the capacity of a subject to reorganize spatial information in order to develop cognitive representations of the environment beyond its field of perception." It locates mentally several geographical points against each other, and organizes them into a coherent configuration. The analysis of the production of city names provides a unique opportunity to study the spatial mental representation of French geographical space for elderly people before and after developing the dementia.

We can visualize this cognitive map by connecting cited cities with a line. Since this visualization becomes cluttered with the numerous lines, we applied the KDEEB bundling technique (Figure 50). We also compared the maps produced at different time periods. Our first investigation showed a stronger graph implication with an elderly person suffering from dementia. This first investigation needs to be validated. The objective of this project is to develop techniques and tools to study the cognitive map of elderly subjects in the years preceding the onset of Alzheimer's disease. The developed tools will use image based technique (e.g. bundling) and will help to better understand the processes involved in the building and the deterioration of cognitive maps.

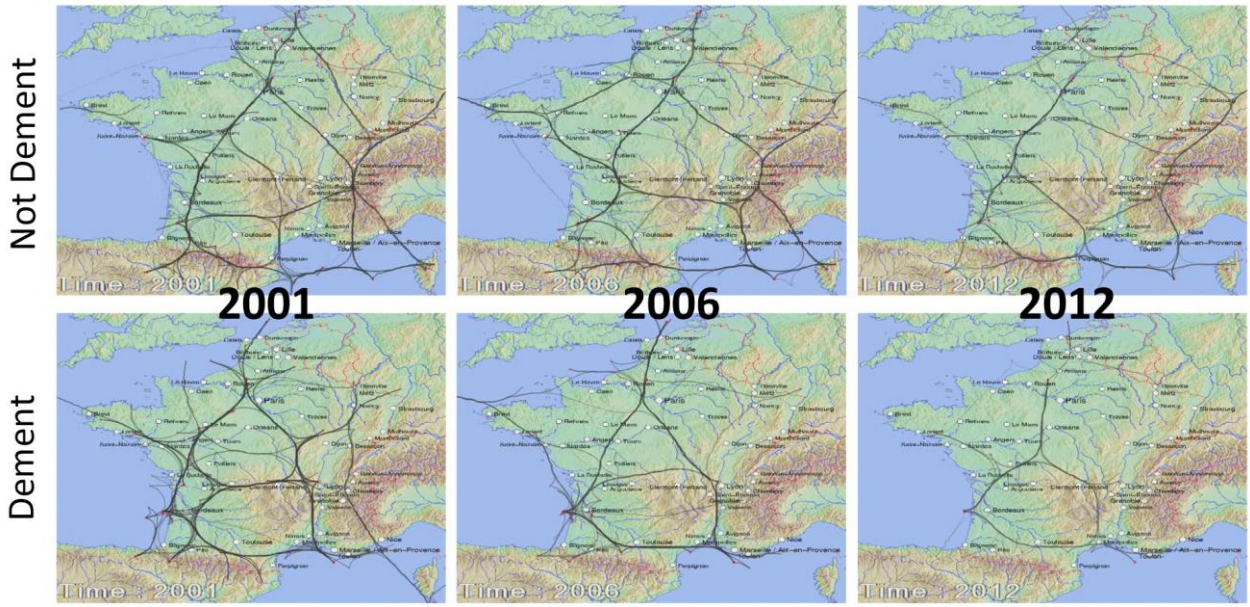


Figure 50: Visualization of cited cities over time. We notice a significant simplification of the network for demented subjects.

7.6.2 Eye tracking

Eye tracking systems help to capture user gaze and are becoming a very popular data source to analyze users' activity. As an example, we stated a research project with the CReA, the research center of the French army, to investigate pilot training with such a technique. During their training, pilots need to learn specific gaze scanning patterns, such as looking outside to prevent collision with other aircraft, then looking inside the cockpit to gather and analyze flight parameters (altitude, speed, direction, engine setting). While this pattern is simple, it remains an issue for pilots beginning their training who spend most of their time looking inside the cockpit. This behavior is especially true with simulator training [Johnson et al. 2006] and it must be corrected as fast as possible to reduce negative transfer from ground-simulation to real flight. Therefore, an experiment has been developed to improve ground-simulation teaching. In this experimentation (Figure 51), the pilot is notified with different modalities (audio or visual alarms) when he or she has an incorrect ocular behavior (e.g. looking for more than 2 seconds inside the cockpit). Depending on the notification modality, the time to learn a specific skill is expected to be different. Hence, we will seek the most appropriate notification. For instance one notification will hide cockpit information, and the pilot will have to look outside to make this information reappear. Another modality will be to emit a warning audio message, and then the user will decide to look back outside or not.

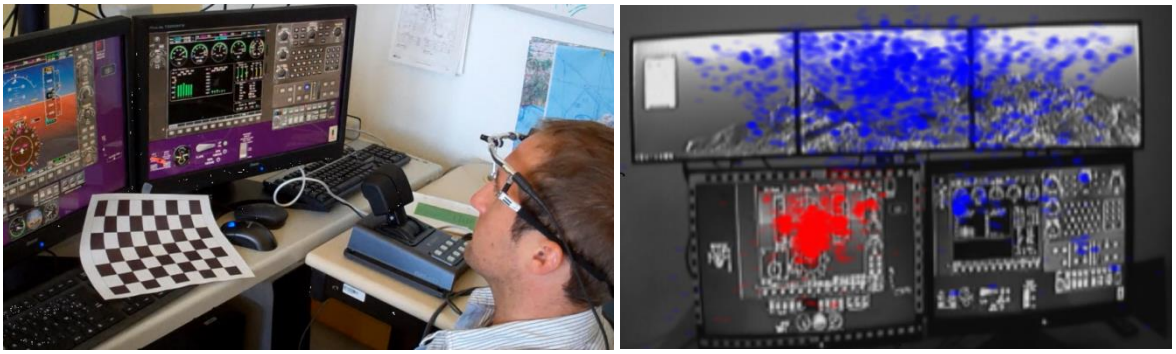


Figure 51: Simulation environment with head mounted eye tracking system (left). Visualization of fixation points inside the cockpit when gathering flight parameters in red, and elsewhere in blue.

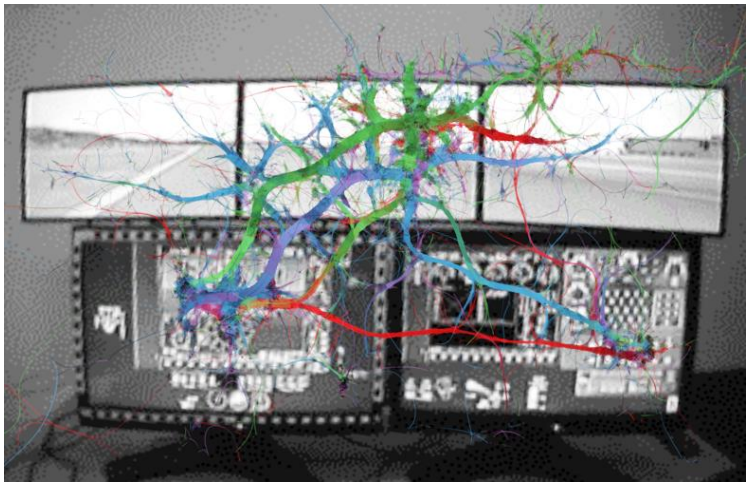


Figure 52: gaze analysis with bundling technique [hurter et al. 2012] [hurter et al. TVCG 2014]. Color corresponds to the gaze direction. Trail width corresponds to the density of the path.

In this early project, image based technique is used at two levels: data recording and data analysis (Figure 52).

Data collection: head mounted eye tracking systems do not provide third part software to capture in real-time user gaze (only post processing). Therefore team members are currently developing a real-time calibration system based on image processing and GP-GPU technique. The system needs further investigation and improvements to be validated and to generalize its usages.

Data analysis: collected data (gaze location and fixation time), must be processed to investigate user behavior. As such, bundling methods help to simplify eye trails and thus help their analysis (Figure 52).

In this project we plan to extend this work in many directions with quantitative analysis of bundled trails, directional bundling computations and statistical data.

7.6.3 Image processing: skin cancer investigation

The diagnosis and prognosis of skin tumors, such as melanoma, is an increasingly important health issue. Given the growing prevalence of such tumors, designing efficient and effective methods to analyze imaged skin lesions is an important goal, e.g. for the automatic measurement of the ABCD (Asymmetry, Border, Colors, and Dimension) diagnostic score.

Color tunneling (Hurter et al., 2014b) already investigated image processing techniques with the manipulation of skin tumor images. Figure 53 show how this software helps to segment the image with a pixel based technique. The user can navigate between view configurations, brush specific area and select skin-mole frontier. See corresponding video for details.

These initial results will be extended with additional interaction techniques such the ABCD score evolution over time.

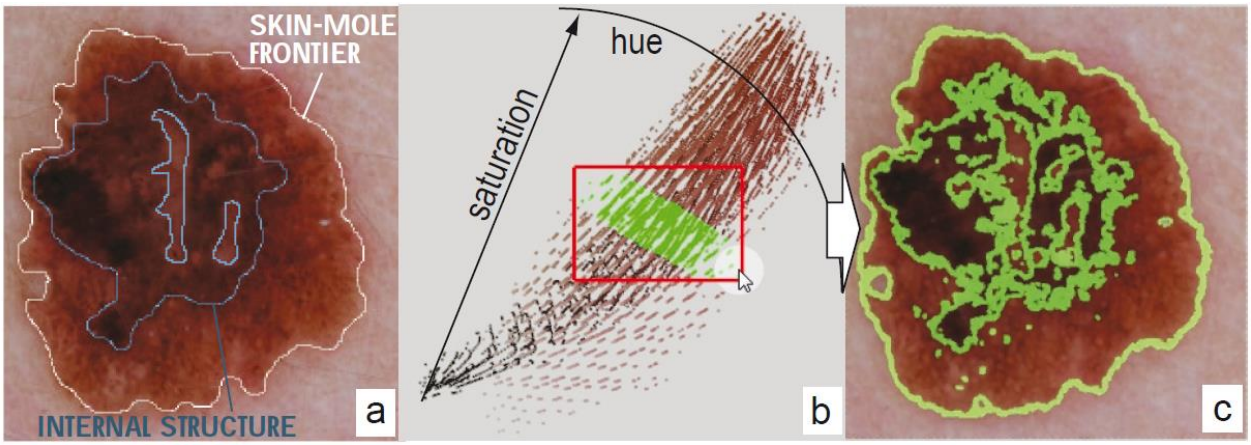


Figure 53: Skin tumor investigation with Color tunneling (Hurter et al., 2014b). One can navigate between data configuration and brush pixel to select a skin-mole frontier.

7.6.4 Point cloud display

3D scanners are measurement instruments capable of collecting a large number of points in three-dimensional coordinates. To collect points, a laser beam is emitted in every direction and thanks to the beam reflected time, the system can compute the distance of the targeted item. These points can be collected from any surface and from large or small size objects (e.g. a mechanical part of a few centimeters high or an archaeological site of several hectares). The accuracy of the measurement can vary from a few millimeters to a few centimeters depending on the distance and the targeted surface. These collected records form a point cloud and can be displayed. This new source of information, in additional to those already used for topological statements (e.g. GPS), open new fields of application with accurate data in the areas of civil engineering and infrastructure, architecture, construction and preservation of cultural heritage. Academic research and professional usages of these data have already started but they are mostly limited to the 3D modeling process of the scanned surfaces. Today, 3D scanners are more accessible in terms of price and usability; as such an increasing number of point clouds are available for potential usages.

Point clouds are composed of discrete elements and for this reason image based technique can be applied to process this type of dataset. In the conventional approaches to 3D modeling, computing power is dedicated to the 3D reconstruction performed by the machine. Thanks to image based techniques we can adopt an innovative approach using the computing power not for the automatic modeling but to provide interactive data manipulation (Figure 54). This approach offers the advantage of giving more flexibility to the 3D modeling process. In other words, the computing power of machines will still be used, but it will be the operator, with interactive tools that will guide the data processing. Building such a tool is challenging since a large data set must be displayed and manipulated in real-time. Color Tunneling was developed for this purpose but with limited interactions (digging into the dataset). Point cloud visualization and manipulation is an interesting application domain to investigate and will provide a way to assess how interactivity can potentially outperform automatic methods, and thus study the optimal amount of automatic as opposed to manual algorithm to provide in order to support efficient 3D modeling.

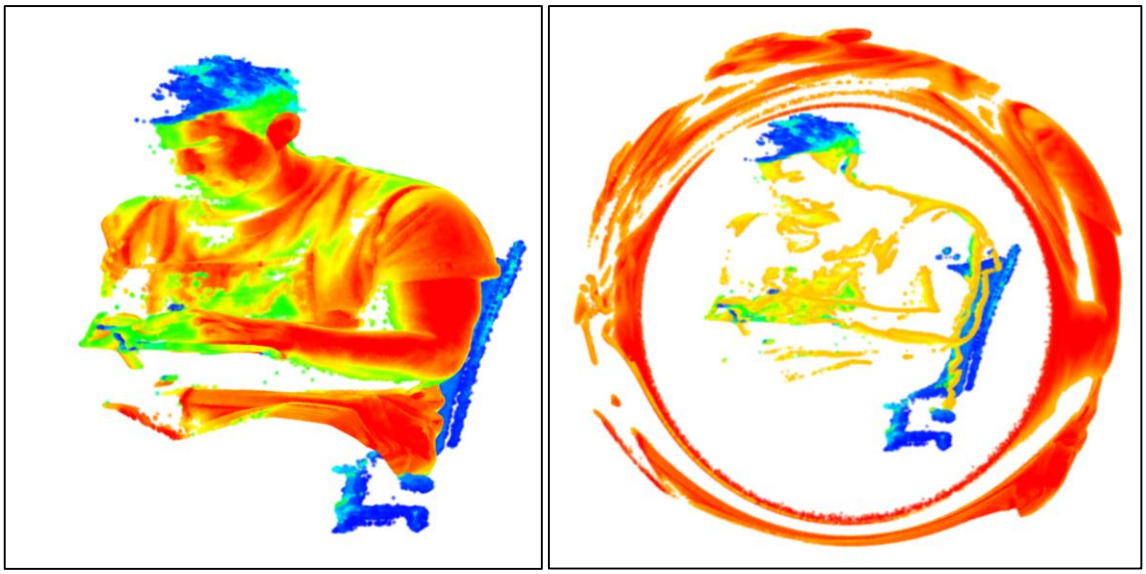


Figure 54: 3D point scan (left), with edge filtering (right, MoleView (Hurter et al., 2011b)).

7.6.5 Movement data analysis

Albeit visually scalable, current pixel-based techniques have not yet been widely applied to time-dependent datasets. Classic solutions, such as animation, work well for scientific datasets where the shapes to detect and track in time are natural and well-known. However, for InfoVis, we need to design new representations whose visual dynamics is both salient (so they can be easily detected) and suggestive (so their visual dynamics suggest to the user the desired types of events to be detected). For instance in recent work we used animation to detect merging and splitting of groups when the shapes are bundle merges/splits, so these are easy to detect, and also suggestive (Hurter et al., 2013b).

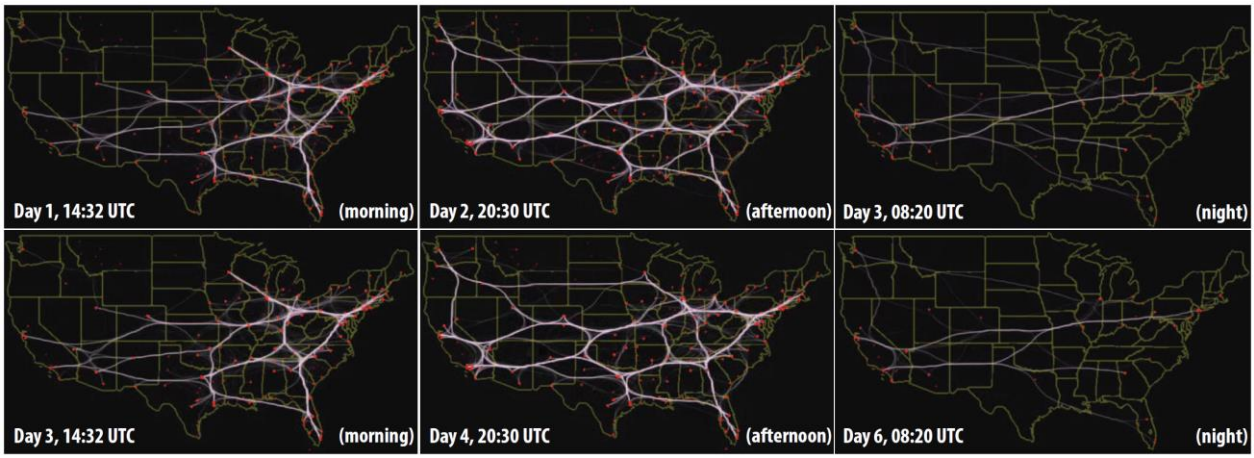


Figure 55: small multiples of US air lines over one week.

Movement data, which are multidimensional time-dependent data, describe changes of spatial positions of discrete mobile objects. Automatically collected movement data (e.g. GPS, RFID, radars, and others) are semantically poor as they basically consist of object identifiers, coordinates in space, and time stamps. Despite this, valuable information about the objects and their movement behavior as well as about the space and time in which they move can be gained from movement data by means of analysis. Analyzing and understanding time-dependent data poses additional non-trivial challenges to information visualization. First, such datasets are by their very nature several orders of magnitude larger than static datasets, which underlines the importance of relying on efficient interactions with

multiple objects and fast algorithms. Secondly, while patterns of interest in static data can be naturally depicted by specific representations in still visualizations, we do not yet know how to best visualize dynamic patterns, which are inherent to time-dependent data. This type of data proposes fascinating perspectives with two research challenges to be addressed.

7.1 Conclusion

In the document presented, I have summarized my investigations with GPU usages to support data visualization and manipulation. In the following, I will list the expended benefits of such techniques and I will describe their limitations and future challenges.

Benefits from the use of graphic cards and their massive memory and parallel computation power: Many of my projects use such benefits; FromDaDy (Hurter et al., 2009b) was the first one, the MoleView (Hurter et al., 2011b) and Color tunneling (Hurter et al., 2014b) which also deal with larger datasets thanks to graphic card computation power. Recent advances in graphic card power introduced dedicated programming languages which are not restrained to graphic visualization but to general computation (compute shader, CUDA, OpenCL…). These new languages will not replace existing rendering pipeline since we still need to display images, and shaders propose an efficient and mature transformation process. However, these new languages provide interesting additional features which are complementary to existing visualization processing. They can be combined to existing rendering pipeline and thus leverage interactive visualization computation. For instance, 3D models can be produced with standard programmable pipeline while physiques of these objects (intertia, accelerations…) can be computed with compute shaders. Regarding Information visualization, the exact same process can occur; color tunneling takes advantage of the programmable rendering pipeline to produce point based rendering, while the points' movements are computed with a render feedback buffer (former version of the compute shader). In the future, this kind of architecture may be generalized but it has a major constraint: it is more complex in terms of program architecture. As such it is far more challenging to produce a GPGPU technique than a standard CPU algorithm. The GPUGPU technique often needs to be refined to support parallel computation and memory concurrency (it is valid to read and to write at the same time in the same memory location). In conclusion, before starting a GPUGPU implementation, one needs to consider if the extra amount of work is worth the extra computational power.

The Image processing field offers many algorithms that are worth applying to image-based information visualization. For instance, KDEEB (C. Hurter et al., 2012) takes advantages of the Kernel density estimation algorithm. Smooth animated bundles use the mean shift algorithm (Comaniciu and Meer, 2002) (clustering algorithm). Data densities can be displayed by simple lighting computation. By synthesizing color, shading, and texture at a pixel level, we achieve a much higher freedom in constructing a wide variety of representation that is able to depict the rich data patterns we aim to analyze. Many image-processing algorithms are available and can be tested with data exploration technique. The goal is to provide efficient tools in terms of data exploration, interaction and visualization. The expected benefits rely on shorter computation processing time while extracting relevant information. Since these image-processing algorithms operate on "pixels" or discrete data space, we need to identify if this large data inaccuracy does not spoil data exploration. As an example, KDEEB bundling turned out to be the fastest bundling algorithm with valid bundling results. Even if DKEEB (Hurter et al., 2013a) uses a discrete accumulation map with a reduced number of cells, the visual bundling results are still valid. In the transformation process, the density map helps to compute a gradient map which is not a discreet data space any more. As an open question, we do not yet know the validity of these data simplifications and when such a process can be applied.

The use of memory instead of computation can reduce algorithm complexity: I experienced this benefit in many projects. FromDaDy uses a texture to store trail identifiers and to drastically reduce

brushing computation. The MoleView uses a gradient map (stored in the GPU memory) to distort trails or pixels. KDEEB uses the same gradient map to compute the bundled version of a graph. It is obvious that storing computation results can speed up the algorithm since it is faster to read a memory cell than to perform multiple computations. However, this expected benefit does not only rely on data storage, but also on algorithm complexity. As such, KDEEB transforms the initial bundling complexity from O(E.(E-1)) to O(E) where E are edges of a graph. This algorithm simplification works thanks to the second expected benefit (applying image processing algorithm; in this case the mean shift clustering), but it also introduces a new data processing pipeline where memory storage plays the central role. This new pipeline may, under given circumstances, simplify algorithm implementations. This last expected benefit has not yet been deeply investigated and this is more a research question than an actual fact and it remains a promising investigation area.

As our knowledge of GPGPU and image-based algorithms improves, it also opens up new opportunities for visualization tool research. Such data exploration will become faster, more interactive and will handle larger datasets. As a final conclusion, image-based InfoVis will definitely improve data visualization and interaction to support efficient decision making and this opens up a large new investigation field for researchers.

Bibliography

Amar, R., Eagan, J., Stasko, J., 2005. Low-Level Components of Analytic Activity in Information Visualization, in: Proceedings of the Proceedings of the 2005 IEEE Symposium on Information Visualization, INFOVIS '05. IEEE Computer Society, Washington, DC, USA, p. 15–. doi:10.1109/INFOVIS.2005.24

Appert, C., Beaudouin-Lafon, M., Mackay, W.E., 2005. Context matters: Evaluating Interaction Techniques with the CIS Model, in: FSEDA, S.F.B., MA, LHG, MSc, P.M.Ms., Moore, D., MBCS, R.R.Bs., , CEng (Eds.), People and Computers XVIII — Design for Life. Springer London, pp. 279–295.

Auber, D., 2004. Tulip — A Huge Graph Visualization Framework, in: Jünger, M., Mutzel, P. (Eds.), Graph Drawing Software, Mathematics and Visualization. Springer Berlin Heidelberg, pp. 105–126.

Auber, D., Chiricota, Y., Jourdan, F., Melançon, G., 2003. Multiscale Visualization of Small World Networks, in: Proceedings of the Ninth Annual IEEE Conference on Information Visualization, INFOVIS'03. IEEE Computer Society, Washington, DC, USA, pp. 75–81.

Bach, B., Pietriga, E., Fekete, J.-D., 2014a. Visualizing dynamic networks with matrix cubes. ACM Press, pp. 877–886. doi:10.1145/2556288.2557010

Bach, B., Pietriga, E., Fekete, J.-D., 2014b. GraphDiaries: Animated Transitions andTemporal Navigation for Dynamic Networks. IEEE Trans. Vis. Comput. Graph. 20, 740–754. doi:10.1109/TVCG.2013.254

Baudel, T., 2004. Browsing Through an Information Visualization Design Space, in: CHI '04 Extended Abstracts on Human Factors in Computing Systems, CHI EA '04. ACM, New York, NY, USA, pp. 765–766. doi:10.1145/985921.985925

Baudisch, P., Tan, D., Collomb, M., Robbins, D., Hinckley, K., Agrawala, M., Zhao, S., Ramos, G., 2006. Phosphor: Explaining Transitions in the User Interface Using Afterglow Effects, in: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology, UIST '06. ACM, New York, NY, USA, pp. 169–178. doi:10.1145/1166253.1166280

Bertin, J., 1983. Semiology of Graphics. University of Wisconsin Press.

Bezerianos, A., Chevalier, F., Dragicevic, P., Elmqvist, N., Fekete, J.D., 2010. Graphdice: A System for Exploring Multivariate Social Networks, in: Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization, EuroVis'10. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 863–872. doi:10.1111/j.1467-8659.2009.01687.x

Bezerianos, A., Dragicevic, P., Balakrishnan, R., 2006. Mnemonic Rendering: An Image-based Approach for Exposing Hidden Changes in Dynamic Displays, in: Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology, UIST '06. ACM, New York, NY, USA, pp. 159–168. doi:10.1145/1166253.1166279

Blaas, J., Botha, C., Grundy, E., Jones, M., Laramee, R., Post, F., 2009. Smooth Graphs for Visual Exploration of Higher-Order State Transitions. IEEE Trans. Vis. Comput. Graph. 15, 969–976. doi:10.1109/TVCG.2009.181

Bothorel, G., Serrurier, M., Hurter, C., 2011. Utilisation D'Outils De Visual Data Mining Pour L'Exploration D'Un Ensemble De RÈGles D'Association, in: 23rd French Speaking Conference on Human-Computer Interaction, IHM '11. ACM, New York, NY, USA, pp. 12:1–12:4. doi:10.1145/2044354.2044369

Bothorel, G., Serrurier, M., Hurter, C., 2013a. Visualization of Frequent Itemsets with Nested Circular Layout and Bundling Algorithm, in: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Li, B., Porikli, F., Zordan, V., Klosowski, J., Coquillart, S., Luo, X., Chen, M., Gotz, D. (Eds.), Advances in Visual Computing, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 396–405.

Bothorel, G., Serrurier, M., Hurter, C., 2013b. From Visualization to Association Rules: An Automatic Approach, in: Spring Conference on Computer Graphics, SCCG ’13. ACM, New York, NY, USA, pp. 057:57–057:64. doi:10.1145/2508244.2508252

Bothorel, G., Serrurier, M., Hurter, C., 2013c. Mining aeronautical data by using visualized driven rules extraction approach. Presented at the ISIATM 2013, 2nd International Conference on Interdisciplinary Science for Innovative Air Traffic Management.

Brosz, J., Nacenta, M.A., Pusch, R., Carpendale, S., Hurter, C., 2013. Transmogrification: Causal Manipulation of Visualizations, in: Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology, UIST ’13. ACM, New York, NY, USA, pp. 97–106. doi:10.1145/2501988.2502046

Buisson, M., Bustico, A., Chatty, S., Colin, F.-R., Jestin, Y., Maury, S., Mertz, C., Truillet, P., 2002. Ivy: Un Bus Logiciel Au Service Du DéVeloppement De Prototypes De SystèMes Interactifs, in: Proceedings of the 14th French-Speaking Conference on Human-Computer Interaction (ConféRence Francophone Sur L’Interaction Homme-Machine), IHM ’02. ACM, New York, NY, USA, pp. 223–226. doi:10.1145/777005.777040

Card, S.K., Mackinlay, J., 1996. The Structure of the Information Visualization Design Space.

Card, S.K., Mackinlay, J.D., Shneiderman, B. (Eds.), 1999. Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Card, S.K., Newell, A., Moran, T.P., 1983. The Psychology of Human-Computer Interaction. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

Carpendale, M.S.T., Cowperthwaite, D.J., Fracchia, F.D., 1997. Extending Distortion Viewing from 2D to 3D. IEEE Comput Graph Appl 17, 42–51. doi:10.1109/38.595268

Chevalier, F., Dragicevic, P., Hurter, C., 2012. Histomages: Fully Synchronized Views for Image Editing, in: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, UIST ’12. ACM, New York, NY, USA, pp. 281–286. doi:10.1145/2380116.2380152

Cleveland, W.S., 1993. Visualizing Data. Hobart Press.

Cohen, M., Puech, C., Sillion, F., Haeberli, P., Segal, M., 1993. Texture Mapping as a Fundamental Drawing Primitive.

Comaniciu, D., Meer, P., 2002. Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. 24, 603–619. doi:10.1109/34.1000236

Conversy, S., Chatty, S., Hurter, C., 2011. Visual Scanning As a Reference Framework for Interactive Representation Design. Inf. Vis. 10, 196–211. doi:10.1177/1473871611415988

Cordeil, M., Hurter, C., Conversy, S., 2011a. AmÉLioration Du Circuit Visuel Des ContrÔLeurs aÉRiens Pour Relier Les DonnÉEs Entre Visualisations En Utilisant Des Transitions AnimÉEs, in: 23rd French Speaking Conference on Human-Computer Interaction, IHM ’11. ACM, New York, NY, USA, pp. 9:1–9:4. doi:10.1145/2044354.2044366

Cordeil, M., Hurter, C., Conversy, S., 2011b. Experimenting and Improving Perception of 3D Rotation-Based Transitions between 2D Visualizations, in: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (Eds.), Human-Computer Interaction – INTERACT 2011, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 531–534.

Cordeil, M., Hurter, C., Conversy, S., Causse, M., 2013. Assessing and Improving 3D Rotation Transition in Dense Visualizations, in: Proceedings of the 27th International BCS Human Computer Interaction Conference, BCS-HCI ’13. British Computer Society, Swinton, UK, UK, pp. 7:1–7:10.

Cui, W., Zhou, H., Qu, H., Wong, P.C., Li, X., 2008. Geometry-Based Edge Clustering for Graph Visualization. IEEE Trans. Vis. Comput. Graph. 14, 1277–1284. doi:10.1109/TVCG.2008.135

Davis, T.A., Hu, Y., 2011. The University of Florida Sparse Matrix Collection. ACM Trans Math Softw 38, 1:1–1:25. doi:10.1145/2049662.2049663

Elmqvist, N., 2005. BalloonProbe: Reducing Occlusion in 3D Using Interactive Space Distortion, in: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST ’05. ACM, New York, NY, USA, pp. 134–137. doi:10.1145/1101616.1101643

Elmqvist, N., Dragicevic, P., Fekete, J., 2008. Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation. IEEE Trans. Vis. Comput. Graph. 14, 1539–1148. doi:10.1109/TVCG.2008.153

Engel, W., 2006. Shader X4 Advanced Rendering Techniques, Har/Cdr edition. ed. Charles River Media, Hingham, Mass.

Ersoy, O., Hurter, C., Paulovich, F., Cantareiro, G., Telea, A., 2011. Skeleton-Based Edge Bundling for Graph Visualization. IEEE Trans. Vis. Comput. Graph. 17, 2364–2373. doi:10.1109/TVCG.2011.233

Fekete, J.-D., Plaisant, C., 2002. Interactive Information Visualization of a Million Items, in: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02), INFOVIS '02. IEEE Computer Society, Washington, DC, USA, p. 117–.

Frishman, Y., Tal, A., 2007. Multi-Level Graph Layout on the GPU. IEEE Trans. Vis. Comput. Graph. 13, 1310–1319. doi:10.1109/TVCG.2007.70580

Gansner, E.R., Hu, Y., North, S., Scheidegger, C., 2011. Multilevel agglomerative edge bundling for visualizing large graphs, in: 2011 IEEE Pacific Visualization Symposium (PacificVis). Presented at the 2011 IEEE Pacific Visualization Symposium (PacificVis), pp. 187–194. doi:10.1109/PACIFICVIS.2011.5742389

Harris, M., 2005. Mapping Computational Concepts to GPUs, in: ACM SIGGRAPH 2005 Courses, SIGGRAPH '05. ACM, New York, NY, USA. doi:10.1145/1198555.1198768

He, B., Fang, W., Luo, Q., Govindaraju, N.K., Wang, T., 2008. Mars: A MapReduce Framework on Graphics Processors, in: Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques, PACT '08. ACM, New York, NY, USA, pp. 260–269. doi:10.1145/1454115.1454152

Heidrich, W., Seidel, H.-P., 1998. View-independent Environment Maps, in: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware, HWWS '98. ACM, New York, NY, USA, p. 39–ff. doi:10.1145/285305.285310

Hofer, R.L., Kunz, A., 2010. Digisketch: Taming Anoto Technology on LCDs, in: Proceedings of the 2Nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS '10. ACM, New York, NY, USA, pp. 103–108. doi:10.1145/1822018.1822034

Holten, D., 2006. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. IEEE Trans. Vis. Comput. Graph. 12, 741–748. doi:10.1109/TVCG.2006.147

Holten, D., van Wijk, J.J., 2009. Force-directed Edge Bundling for Graph Visualization, in: Proceedings of the 11th Eurographics / IEEE - VGTC Conference on Visualization, EuroVis'09. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 983–998. doi:10.1111/j.1467-8659.2009.01450.x

Hurter, C., 2010. Caractérisation de visualisations et exploration interactive de grandes quantités de données multidimensionnelles. Université Paul Sabatier - Toulouse III.

Hurter, C., Conversy, S., 2008. Towards Characterizing Visualizations, in: Graham, T.C.N., Palanque, P. (Eds.), Interactive Systems. Design, Specification, and Verification, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 287–293.

Hurter, C., Conversy, S., Gianazza, D., Telea, A.C., 2014a. Interactive image-based information visualization for aircraft trajectory analysis. Transp. Res. Part C Emerg. Technol. doi:10.1016/j.trc.2014.03.005

Hurter, C., Conversy, S., Vinot, J.-L., 2009a. Temporal data visualizations for Air Traffic Controllers (ATC), in: Interacting with Temporal Data, CHI 2009 Workshop.

Hurter, C., Edouard, P., Gaits, V., Nadfaoui, H., Pailler, J., Letondal, C., Conversy, S., 2011a. Etude exploratoire du stylo électronique pour le Contrôle Aérien, in: 23rd French Speaking Conference on Human-Computer Interaction, IHM '11. ACM, New York, NY, USA, pp. 15:1–15:8. doi:10.1145/2044354.2044373

Hurter, C., Ersoy, O., Fabrikant, S., Klein, T., Telea, A., 2013a. Bundled Visualization of Dynamic Graph and Trail Data. IEEE Trans. Vis. Comput. Graph. Early Access Online. doi:10.1109/TVCG.2013.246

Hurter, C., Ersoy, O., Telea, A., 2012. Graph Bundling by Kernel Density Estimation. Comp Graph Forum 31, 865–874. doi:10.1111/j.1467-8659.2012.03079.x

Hurter, C., Ersoy, O., Telea, A., 2013b. Smooth bundling of large streaming and sequence graphs, in: Visualization Symposium (PacificVis), 2013 IEEE Pacific. Presented at the Visualization Symposium (PacificVis), 2013 IEEE Pacific, pp. 41–48. doi:10.1109/PacificVis.2013.6596126

Hurter, C., Kapp, V., Conversy, S., 2008. An Infovis approach to compare ATC comets, in: ICRAT 2008, International Conference on Research in Air Transportation.

Hurter, C., Lesbordes, R., Letondal, C., Vinot, J.-L., Conversy, S., 2012. Strip'TIC: Exploring Augmented Paper Strips for Air Traffic Controllers, in: Proceedings of the International Working Conference on Advanced Visual Interfaces, AVI '12. ACM, New York, NY, USA, pp. 225–232. doi:10.1145/2254556.2254598

Hurter, C., Serrurier, M., Alonso, R., Tabart, G., Vinot, J.-L., 2010a. An Automatic Generation of Schematic Maps to Display Flight Routes for Air Traffic Controllers: Structure and Color Optimization, in: Proceedings of the International Conference on Advanced Visual Interfaces, AVI '10. ACM, New York, NY, USA, pp. 233–240. doi:10.1145/1842993.1843034

Hurter, C., Taylor, R., Carpendale, S., Telea, A., 2014b. Color Tunneling: Interactive Exploration and Selection in Volumetric Datasets, in: 2014 IEEE Pacific Visualization Symposium (PacificVis). Presented at the 2014 IEEE Pacific Visualization Symposium (PacificVis), pp. 225–232. doi:10.1109/PacificVis.2014.61

Hurter, C., Telea, A., Ersoy, O., 2011b. MoleView: An Attribute and Structure-Based Semantic Lens for Large Element-Based Plots. IEEE Trans. Vis. Comput. Graph. 17, 2600–2609. doi:10.1109/TVCG.2011.223

Hurter, C., Tissoires, B., Conversy, S., 2009b. FromDaDy: Spreading Aircraft Trajectories Across Views to Support Iterative Queries. IEEE Trans. Vis. Comput. Graph. 15, 1017–1024. doi:10.1109/TVCG.2009.145

Hurter, C., Tissoires, B., Conversy, S., 2010b. Accumulation as a tool for efficient visualization of geographical and temporal data, in: AGILE Workshop Geospatial Visual Analytics: Focus on Time.

Jankun-Kelly, T.J., Dwyer, T., Holten, D., Hurter, C., Nöllenburg, M., Weaver, C., Xu, K., 2014. Scalability Considerations for Multivariate Graph Visualization, in: Kerren, A., Purchase, H.C., Ward, M.O. (Eds.), Multivariate Network Visualization, Lecture Notes in Computer Science. Springer International Publishing, pp. 207–235.

Kajiya, J.T., 1986. The Rendering Equation, in: Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '86. ACM, New York, NY, USA, pp. 143–150. doi:10.1145/15922.15902

Koffa, K., 1963. Principles of Gestalt Psychology. Harcourt, Brace & World.

Lambert, A., Bourqui, R., Auber, D., 2010. Winding Roads: Routing Edges into Bundles, in: Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization, EuroVis'10. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 853–862. doi:10.1111/j.1467-8659.2009.01700.x

Lambert, A., Bourqui, R., Auber, D., 2010. 3D Edge Bundling for Geographical Data Visualization, in: Proceedings of the 2010 14th International Conference Information Visualisation, IV '10. IEEE Computer Society, Washington, DC, USA, pp. 329–335. doi:10.1109/IV.2010.53

Lee, B., Plaisant, C., Parr, C.S., Fekete, J.-D., Henry, N., 2006. Task Taxonomy for Graph Visualization, in: Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization, BELIV '06. ACM, New York, NY, USA, pp. 1–5. doi:10.1145/1168149.1168168

Letondal, C., Hurter, C., Lesbordes, R., Vinot, J.-L., Conversy, S., 2013. Flights in My Hands: Coherence Concerns in Designing Strip'TIC, a Tangible Space for Air Traffic Controllers, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '13. ACM, New York, NY, USA, pp. 2175–2184. doi:10.1145/2470654.2481300

Lieberman, H., 1994. Powers of Ten Thousand: Navigating in Large Information Spaces, in: Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology, UIST '94. ACM, New York, NY, USA, pp. 15–16. doi:10.1145/192426.192434

MacKay, W.E., 1999. Is Paper Safer? The Role of Paper Flight Strips in Air Traffic Control. ACM Trans Comput-Hum Interact 6, 311–340. doi:10.1145/331490.331491

McDonnel, B., Elmqvist, N., 2009. Towards utilizing GPUs in information visualization: a model and implementation of image-space operations. IEEE Trans. Vis. Comput. Graph. 15, 1105–1112. doi:10.1109/TVCG.2009.191

McGuffin, M.J., Tancau, L., Balakrishnan, R., 2003. Using deformations for browsing volumetric data, in: IEEE Visualization, 2003. VIS 2003. Presented at the IEEE Visualization, 2003. VIS 2003, pp. 401–408. doi:10.1109/VISUAL.2003.1250400

Mertz, C., Chatty, S., Vinot, J., 2000. The influence of design techniques on user interfaces: the DigiStrips experiment for air traffic control.

Nejjari, C., Tessier, J.F., Baldi, I., Barberger-Gateau, P., Dartigues, J.F., Salamon, R., 1997. [Epidemiologic aspects of respiratory aging: contribution of the PAQUID survey]. Rev. Dépidémiologie Santé Publique 45, 417–428.

Nguyen, Q., Eades, P., Hong, S.-H., 2013a. StreamEB: Stream Edge Bundling, in: Proceedings of the 20th International Conference on Graph Drawing, GD'12. Springer-Verlag, Berlin, Heidelberg, pp. 400–413. doi:10.1007/978-3-642-36763-2_36

Nguyen, Q., Eades, P., Hong, S.-H., 2013b. On the Faithfulness of Graph Visualizations, in: Proceedings of the 20th International Conference on Graph Drawing, GD'12. Springer-Verlag, Berlin, Heidelberg, pp. 566–568. doi:10.1007/978-3-642-36763-2_55

Owens, J.D., Houston, M., Luebke, D., Green, S., Stone, J.E., Phillips, J.C., 2008. GPU Computing. Proc. IEEE 96, 879–899. doi:10.1109/JPROC.2008.917757

Owens, J.D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A.E., Purcell, T., 2007. A survey of general-purpose computation on graphics hardware.

Pajarola, R., Sainz, M., Guidotti, P., 2004. Confetti: object-space point blending and splatting. IEEE Trans. Vis. Comput. Graph. 10, 598–608. doi:10.1109/TVCG.2004.19

Phong, B.T., 1975. Illumination for Computer Generated Pictures. Commun ACM 18, 311–317. doi:10.1145/360825.360839

Purcell, T.J., Buck, I., Mark, W.R., Hanrahan, P., 2002. Ray Tracing on Programmable Graphics Hardware, in: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '02. ACM, New York, NY, USA, pp. 703–712. doi:10.1145/566570.566640

Rekimoto, J., 1997. Pick-and-drop: A Direct Manipulation Technique for Multiple Computer Environments, in: Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology, UIST '97. ACM, New York, NY, USA, pp. 31–39. doi:10.1145/263407.263505

Renso, C., Spaccapietra, S., Zimányi, E., 2013. Mobility Data. Cambridge University Press.

Rondeau, V., Jacqmin-Gadda, H., Commenges, D., Helmer, C., Dartigues, J.-F., 2009. Aluminum and silica in drinking water and the risk of Alzheimer's disease or cognitive decline: findings from 15-year follow-up of the PAQUID cohort. Am. J. Epidemiol. 169, 489–496. doi:10.1093/aje/kwn348

Roth, S.D., 1982. Ray casting for modeling solids. Comput. Graph. Image Process. 18, 109–144. doi:10.1016/0146-664X(82)90169-1

Rusinkiewicz, S., Levoy, M., 2000. QSplat: A Multiresolution Point Rendering System for Large Meshes, in: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 343–352. doi:10.1145/344779.344940

Sainz, M., Pajarola, R., 2004. Point-based Rendering Techniques. Comput Graph 28, 869–879. doi:10.1016/j.cag.2004.08.014

Savery, C., Hurter, C., Lesbordes, R., Cordeil, M., Graham, T.C.N., 2013. When Paper Meets Multi-touch: A Study of Multi-modal Interactions in Air Traffic Control, in: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (Eds.), Human-Computer Interaction – INTERACT 2013, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 196–213.

Scheepens, R., Willems, N., Van de Wetering, H., Andrienko, G., Andrienko, N., van Wijk, J.J., 2011. Composite Density Maps for Multivariate Trajectories. IEEE Trans. Vis. Comput. Graph. 17, 2518–2527. doi:10.1109/TVCG.2011.181

Shanmugasundaram, M., Irani, P., Gutwin, C., 2007. Can Smooth View Transitions Facilitate Perceptual Constancy in Node-link Diagrams?, in: Proceedings of Graphics Interface 2007, GI '07. ACM, New York, NY, USA, pp. 71–78. doi:10.1145/1268517.1268531

Shneiderman, B., 1996. The eyes have it: A task by data type taxonomy for information visualizations, in: In IEEE Symposium on Visual Languages. pp. 336–343.

Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. CRC Press.

Song, H., Guimbretiere, F., Grossman, T., Fitzmaurice, G., 2010. MouseLight: Bimanual Interactions on Digital Paper Using a Pen and a Spatially-aware Mobile Projector, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10. ACM, New York, NY, USA, pp. 2451–2460. doi:10.1145/1753326.1753697

Sutherland, I., 2012. The Tyranny of the Clock. Commun ACM 55, 35–36. doi:10.1145/2347736.2347749

Taylor, A.R., Gibson, S.J., Peracaula, M., Martin, P.G., Landecker, T.L., Brunt, C.M., Dewdney, P.E., Dougherty, S.M., Gray, A.D., Higgs, L.A., Kerton, C.R., Knee, L.B.G., Kothes, R., Purton, C.R., Uyaniker, B., Wallace, B.J., Willis, A.G., Durand, D., 2003. The Canadian Galactic Plane Survey. Astron. J. 125, 3145. doi:10.1086/375301

Telea, A., Ersoy, O., 2010. Image-based Edge Bundles: Simplified Visualization of Large Graphs, in: Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization, EuroVis'10. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 843–852. doi:10.1111/j.1467-8659.2009.01680.x

Tessier, J.F., Nejjari, C., Letenneur, L., Filleul, L., Marty, M.L., Gateau, P.B., Dartigues, J.F., 2001. Dyspnea and 8-year mortality among elderly men and women: The PAQUID cohort study. Eur. J. Epidemiol. 17, 223–229. doi:10.1023/A:1017977715073

Thomas, J.J., Cook, K.A., 2005. Illuminating the Path: The Research and Development Agenda for Visual Analytics. IEEE Computer Society Press.

Thompson, C.J., Hahn, S., Oskin, M., 2002. Using modern graphics architectures for general-purpose computing: a framework and analysis, in: 35th Annual IEEE/ACM International Symposium on Microarchitecture, 2002. (MICRO-35). Proceedings. Presented at the 35th Annual IEEE/ACM International Symposium on Microarchitecture, 2002. (MICRO-35). Proceedings, pp. 306–317. doi:10.1109/MICRO.2002.1176259

Thrower, N.J.W., 1969. Edmond Halley as a Thematic Geo-Cartographer. Ann. Assoc. Am. Geogr. 59, 652–676. doi:10.1111/j.1467-8306.1969.tb01805.x

Tissoires, B., 2011. Conception (instrumenté), réalisation et optimisation de graphismes interactifs (PhD). Université de Toulouse, Université Toulouse III - Paul Sabatier.

Tufte, E.R., 1986. The Visual Display of Quantitative Information. Graphics Press, Cheshire, CT, USA.

Van Ham, F., Wattenberg, M., 2008. Centrality Based Visualization of Small World Graphs, in: Proceedings of the 10th Joint Eurographics / IEEE - VGTC Conference on Visualization, EuroVis'08. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 975–982. doi:10.1111/j.1467-8659.2008.01232.x

Van Liere, R., Leeuw, W. de, 2003. GraphSplatting: visualizing graphs as continuous fields. IEEE Trans. Vis. Comput. Graph. 9, 206–212. doi:10.1109/TVCG.2003.1196007

Van Wijk, J.J., Telea, A., 2001. Enridged Contour Maps, in: Proceedings of the Conference on Visualization '01, VIS '01. IEEE Computer Society, Washington, DC, USA, pp. 69–74.

Van Wijk, J.J., van de Wetering, H., 1999. Cushion Treemaps: Visualization of Hierarchical Information, in: Proceedings of the 1999 IEEE Symposium on Information Visualization, INFOVIS '99. IEEE Computer Society, Washington, DC, USA, p. 73–.

Vinot, J.-L., Letondal, C., Lesbordes, R., Chatty, S., Conversy, S., Hurter, C., 2014. Tangible Augmented Reality for Air Traffic Control. interactions 21, 54–57. doi:10.1145/2627598

Von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J. j., Fekete, J.-D., Fellner, D. w., 2011. Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges. Comput. Graph. Forum 30, 1719–1749. doi:10.1111/j.1467-8659.2011.01898.x

Wilkinson, L., Wills, D., Rope, D., Norton, A., Dubbs, R., more, & 3, 2005. The Grammar of Graphics, 2nd edition. ed. Springer, New York.

Willems, N., van de Wetering, H., van Wijk, J.J., 2009. Visualization of Vessel Movements, in: Proceedings of the 11th Eurographics / IEEE - VGTC Conference on Visualization, EuroVis'09. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, pp. 959–966. doi:10.1111/j.1467-8659.2009.01440.x

Wise, J.A., Thomas, J.J., Pennock, K., Lantrip, D., Pottier, M., Schur, A., Crow, V., 1995. Visualizing the Non-visual: Spatial Analysis and Interaction with Information from Text Documents, in: Proceedings of the 1995 IEEE Symposium on Information Visualization, INFOVIS '95. IEEE Computer Society, Washington, DC, USA, p. 51.

Yi, J.S., Melton, R., Stasko, J., Jacko, J.A., 2005. Dust & Magnet: Multivariate Information Visualization Using a Magnet Metaphor. Inf. Vis. 4, 239–256. doi:10.1057/palgrave.ivs.9500099

Young, D., Shneiderman, B., 1993. A Graphical Filter/Flow Representation of Boolean Queries: A Prototype Implementation and Evaluation. J Am Soc Inf Sci 44, 327–339. doi:10.1002/(SICI)1097-4571(199307)44:6<327::AID-ASI3>3.0.CO;2-J

Selected research papers

Book Chapters:

- Jankun-Kelly, T.J., Dwyer, T., Holten, D., Hurter, C., Nöllenburg, M., Weaver, C., Xu, K., 2014. **Scalability Considerations for Multivariate Graph Visualization**, *in: Kerren, A., Purchase, H.C., Ward, M.O. (Eds.), Multivariate Network Visualization, Lecture Notes in Computer Science. Springer International Publishing, pp. 207–235.*
- Christophe Hurter, Gennady Andrienko, Natalia Andrienko, Ralf Hartmut Guting, Mahmoud Sakr. **Air Traffic Analysis**. Book chapter in Mobility Data: Modeling, Management, and Understanding, Cambridge University Press (September 30, 2013), 368 pages, ISBN-13: 978-1107021716.

Journals:

- Christophe Hurter, Stéphane Conversy, David Gianazza, Alexandru C. Telea. **Interactive image-based information visualization for aircraft trajectory analysis.** Transportation Research Part C: Emerging Technologies. doi:10.1016/j.trc.2014.03.005
- Christophe Hurter, Richard Alligier, David Gianazza, Stéphane Puechmorel, Gennady Andrienko, Natalia Andrienko, **Wind parameters extraction from aircraft trajectorie**. Computers, Environment and Urban Systems. doi:10.1016/j.compenvurbsys.2014.01.005
- Christophe Hurter, Ozan Ersoy, Sara Fabrikant, Tijmen Klein, Alexandru Telea, **Bundled Visualization of Dynamic Graph and Trail Data**. (TVCG) Visualization and Computer Graphics, IEEE Transactions on , vol.PP, no.99, pp.1,1.
- Gennady Andrienko, Natalia Andrienko, Christophe Hurter, Salvatore Rinzivillo, Stefan Wrobel. **Scalable Analysis of Movement Data for Extracting and Exploring Significant Places.** *IEEE Transactions on Visualization and Computer Graphics (TVCG)***,** 2013, v. 19.
- Christophe Hurter, Ozan Ersoy, Alexandru Telea **Graph Bundling by Kernel Density Estimation.** *(EuroVis 2012). Computer Graphics Forum 31, 3pt1 (june 2012) p*865-874.
- Christophe Hurter, Ozan Ersoy and Alexandru Telea. 2011. **MoleView: An Attribute and Structure-Based Semantic Lens for Large Element-Based Plots.** *IEEE Transactions on Visualization and Computer Graphics* (InfoVis 2011)17, 12 (December 2011), 2600-2609.
- Ozan Ersoy, Christophe Hurter, Fernando Paulovich, Gabriel Cantareiro, and Alex Telea. 2011. **Skeleton-Based Edge Bundling for Graph Visualization.** *IEEE Transactions on Visualization and Computer Graphics* (InfoVis 2011)17, 12 (December 2011), 2364-2373.

- Stéphane Conversy, Stéphane Chatty, Christophe Hurter. **Visual scanning as a reference framework for interactive representation design.** *Information Visualization 10, 3 (July 2011), 196-211.*
- Christophe Hurter., Benjamin Tissoires, Stéphane Conversy. **FromDaDy: spreading data across views to support iterative exploration of aircraft trajectories.** *In IEEE Transactions on Visualization and Computer Graphics xx(y), (Proceedings of IEEE InfoVis 2009). IEEE Transactions on Visualization and Computer Graphics (TVCG), vol. 15, no. 6, pp. xxviv-xxvv, Nov./Dec. 2009.*

International Conferences:

- Maxime Cordeil, Christophe Hurter, Stéphane Conversy and Mickaël Causse. **Assessing and Improving 3D Rotation Transition in Dense Visualizations.** In *Proc. of the 27th BCS Conference on Human Computer Interaction (HCI2013)*,
- John Brosz, Miguel Nacenta, Ricky Pusch, Sheelagh Carpendale, and Christophe Hurter. **Transmogrification: Casual Manipulation of Visualizations**. In *Proceedings of the 26th annual ACM symposium on User interface software and technology* (UIST '13). ACM,
- Cheryl Savery, Christophe Hurter, Rémi Lesbordes, Maxime Cordeil, and T.C. Nicholas Graham. **When Paper Meets Multi-touch: A Study of Multi-modal Interactions in Air Traffic Controls**. In *Proceedings of Interact 2013,(*INTERACT 2013), to appear
- Catherine Letondal, Christophe Hurter, Jean-Luc Vinot, Rémi Lesbordes and Stéphane Conversy. **Strip'TIC: designing a paper-based tangible interactive space for air traffic controllers**. In *Proceedings of ACM SIGCHI 2013,(*CHI 2013) ***honorable mention CHI 2013***
- Christophe Hurter, Ozan Ersoy, Alexandru Telea. **Smooth Bundling of Large Streaming and Sequence Graphs.** Proceedings of IEEE Pacific Visualization 2013 (PacificVis 2013). ***honorable mention.***
- Fanny Chevalier, Pierre Dragicevic, and Christophe Hurter. **Histomages: fully synchronized views for image editing.** In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12). ACM, New York, NY, USA, 281-286.
- Christophe Hurter, Benjamin R. Cowan, Audrey Girouard, Nathalie Henry Riche, **Active progress bars: facilitating the switch to temporary activities.** In *Proc. of the 26th BCS Conference on Human Computer Interaction (HCI2012)*, Birmingham, UK, September 2012.
- Christophe Hurter, Rémi Lesbordes, Catherine Letondal, Jean-Luc Vinot and Stéphane Conversy. **Strip'TIC: Exploring Automatic Paper Strip for Air Traffic Controllers.** In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (AVI '12), Genny Tortora, Stefano Levialdi, and Maurizio Tucci (Eds.). ACM, New York, NY, USA, 225-232.
- Gennady Andrienko, Natalia Andrienko, Christophe Hurter, Salvatore Rinzivillo, Stefan Wrobel. **From Movement Tracks through Events to Places: Extracting and Characterizing Significant Places from Mobility Data.** IEEE Visual Analytics Science and

Technology (VAST 2011) Proceedings, IEEE Computer Society Press. ***Best VAST 2011 paper.***

- Christophe Hurter, Mathieu Serrurier, Roland Alonzo, Jean-Luc Vinot, Gilles Tabart. **An automatic generation of metro-like maps to display Flight Routes for Air Traffic Controllers: structure and color optimization.** *Conference Proceedings of AVI'10, Advanced Visual Interfaces, Roma, Italy, ACM Press*.