



Interactive Transitions for Map Applications

María Jesús Lobo

► To cite this version:

María Jesús Lobo. Interactive Transitions for Map Applications. Human-Computer Interaction [cs.HC]. Université Paris Saclay (COmUE), 2017. English. NNT : 2017SACLS478 . tel-01698880

HAL Id: tel-01698880

<https://theses.hal.science/tel-01698880>

Submitted on 1 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Interactive Transitions for Map Applications

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université Paris-Sud

École doctorale n°580 Sciences et Technologies de l'Information et de
la Communication (STIC)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Orsay, le 5 décembre 2017 par

María Jesús LOBO

Composition du Jury :

Sarah Cohen-Boulakia
Maître de conférences, U. Paris-Sud
Christophe Hurter
Professeur, ENAC
Guillaume Touya
Directeur de recherche, IGN
Jason Dykes
Professeur, City, University of London
Emmanuel Pietriga
Directeur de Recherche, CNRS
Caroline Appert
Chargée de recherche, U. Paris-Sud

Présidente
Rapporteur
Rapporteur
Examineur
Directeur de thèse
Encadrante de thèse

Para Vicente, Clemente, Ignacio y Blanca

CONTENTS

Acknowledgements	vii
Resume	ix
1 Introduction	3
1.1 Context	3
1.2 Problem	5
1.3 Research Question	8
1.4 Thesis Outline	8
2 Related Work	11
2.1 Why provide a multi-layer representation?	11
2.1.1 Information Visualization Taxonomies	12
2.1.2 Spatial information taxonomies	13
2.2 How to combine different representations?	15
2.2.1 Spatial Multiplexing	15
2.2.2 Temporal Multiplexing	21
2.3 How to interact with a multi-layer representation?	26
2.3.1 Classifications of interaction techniques	26
2.3.2 Multiscale Navigation	27
2.3.3 Navigation between heterogeneous layers	30
2.4 Summary	32
3 An Evaluation of Interactive Map Comparison Techniques	35
3.1 Map comparison techniques	36
3.1.1 Juxtapose (JX)	37
3.1.2 Translucent Overlay (OV)	38
3.1.3 Swipe (SW)	38
3.1.4 Blending Lens (BL)	38
3.1.5 Offset Lens (OL)	39

3.2	User Study	39
3.2.1	Maps	40
3.2.2	Tasks.	41
3.2.3	Hypotheses.	43
3.2.4	Participants and Apparatus	44
3.2.5	Procedure	44
3.2.6	Results	45
3.3	Discussion	48
3.3.1	Validity	49
3.4	Summary	51
4	MapMosaic: dynamic spatial multiplexing	53
4.1	Background and Motivation.	54
4.2	Interviews with GIS experts	55
4.2.1	GIS Operators	56
4.2.2	GI Science Researchers.	58
4.2.3	Design Requirements	59
4.3	MapMosaic: Concepts and Features	61
4.3.1	Layers	62
4.3.2	Compositing Regions.	63
4.3.3	Filters	65
4.4	Implementation and Performance Evaluation	66
4.5	Comparison with MAPublisher and QGIS.	68
4.5.1	Viscosity, Premature Commitment and Progressive Evaluation	71
4.5.2	Abstraction, Role Expressiveness.	73
4.5.3	Hard Mental Operations, Error proneness	74
4.5.4	Visibility and Juxtaposability	74
4.6	Experts' feedback	75
4.6.1	Cartographic data maintenance	76
4.6.2	Crisis Management	78
4.7	Summary	80

5	Animation plans for before/after satellite images	81
5.1	Background and Motivation	83
5.1.1	Slideshow-style Animations	83
5.1.2	Supporting Process Animations	84
5.2	Animation Plans	85
5.2.1	Predefined Animation Primitives	89
5.3	User Study	91
5.3.1	Participants and Apparatus	92
5.3.2	Procedure	92
5.3.3	Results	94
5.3.4	Discussion	97
5.4	Authoring Tool	98
5.4.1	Tool Description	98
5.4.2	User Interface	98
5.4.3	Implementation	100
5.4.4	User Study	101
5.5	Discussion	106
6	Conclusion	109
6.1	Summary	110
6.1.1	Characterization and evaluation of existing map comparison techniques	110
6.1.2	<i>MapMosaic</i> : spatial multiplexing for vector and raster geographical layers	111
6.1.3	<i>Baia</i> : Time multiplexing for before-and-after satellite images	111
6.2	Perspectives and Future Work	112
6.2.1	Further Evaluations	112
6.2.2	Temporal Multiplexing for Vector Objects	114
	References	116
A	Appendix	133
A.1	Maps used in the first evaluation	133
A.2	Images and material used for the <i>Baia</i> evaluation	133
A.2.1	Images and animations used in the first experiment	133

A.2.2	Tutorial used for <i>Baia's</i> prototype evaluation	133
-------	--	-----

ACKNOWLEDGEMENTS

First, thanks to my advisors. Thank you for believing in me since my first internship in 2013, for introducing me to research and HCI and for teaching me so much. Thank you also for all the almost instantaneous mail answers, for the encouragement, and the jokes. Thanks to Caroline for her patience and for always finding the time to help me. Thanks to Emmanuel for working with me in Chile, for all his support and for teaching me to be rigorous. I feel really lucky that we could work together.

Also thanks to my jury, Christophe, Guillaume, Jason and Sarah for carefully reading my thesis, and all your feedback and questions.

I had the chance to work with a great team. Thanks to Olivier for advising me during my first internship and to Anastasia for the interest and the advice. And thanks to all the other students, Hugo, Marie, Rafa, Bruno, Anna and also Hande, Vit and Dylan for the good atmosphere in the open space. Thanks especially to Arnaud, for all the discussions, tea, coffee and "highly scientific" videos shared across these three years.

Thanks also to all our collaborators from IGN and GRED, Guillaume, Sidonie, Charlotte, Marion and Jean-François for all the feedback and insight.

This PhD would have been significantly harder without all the support from my family and friends. So thanks to my parents Christel and Carlos, and all my family for always supporting my decisions and even coming to visit. Thanks to Ignacio, Sandra, Laura, Despoina, Matias, JM, and Felipe for making the life in Paris easier. And thanks to all my friends left in Chile for making me feel at home whenever we saw each other.

Et merci à David pour être là, encore et toujours.

This research was supported by ANR project Map-Muxing (ANR-14-CE24-0011-02).

SYNTHÈSE

Les utilisateurs experts de SIG (Systèmes d'Information Géographique) doivent souvent mettre en relation et comparer des représentations hétérogènes d'une même région géographique. Par exemple, la mise à jour des bases des données géographiques, comme OpenStreetMap, nécessite une comparaison entre des cartes existantes et des images satellite récentes. Les moyens de combiner les cartes sont pourtant souvent limités à des techniques qui ne prennent pas en compte les données contenues dans les cartes, comme des techniques de superposition qui permettent de varier l'opacité de la couche supérieure ou des techniques de juxtaposition qui montrent les images cote à cote. Ces techniques ne sont pas efficaces pour accomplir des tâches dans certains domaines comme l'analyse de crime ou la planification urbaine. Cette thèse présente trois contributions afin de proposer des nouvelles transitions interactives pour combiner différentes représentations en une, soit de façon spatiale (multipléxage spatiale) ou temporelle (multipléxage temporel). Ces techniques ont comme objectif de permettre aux utilisateurs de comparer et mettre en relations des couches géographiques hétérogènes facilement.

Afin de mieux comprendre les limites des techniques existantes, la première contribution de cette thèse est une évaluation de cinq techniques de comparaison de cartes interactives. On caractérise ces techniques par rapport à leur niveau de perturbation visuelle, de division de l'attention et leur stratégie de recherche. Pour les évaluer, on demande aux participants de trouver des différences entre des images aériennes et des cartes topographiques (modifiées à la main en ajoutant six types de différences). Les résultats suggèrent que les techniques qui superposent les couches sont plus efficaces que les techniques qui les juxtaposent et qu'une stratégie de recherche motrice peut apporter des bénéfices pour certaines tâches.

D'après les résultats de l'évaluation et des entretiens avec des utilisateurs experts en GIS, on introduit MapMosaic, la deuxième contribution de cette thèse. MapMosaic est une nouvelle technique de multipléxage spatiale pour combiner des cartes. Ce modèle de composition dynamique permet aux utilisateurs de créer et manipuler des régions de composition locale de façon interactive, en considérant l'information sémantique et les attributs des objets et des champs. On a évalué MapMosaic en utilisant deux approches: premièrement, on compare son modèle d'interaction au

modèle de QGIS (un logiciel SIG très utilisé) et MAPublisher (un outil cartographique professionnel) en utilisant les “Dimensions Cognitives” et avec une comparaison analytique. Les résultats suggèrent que le modèle de MapMosaic est plus flexible et peut mieux appuyer les utilisateurs dans leur tâches. Ensuite, on rapporte des retours utilisateurs experts qui confirment le potentiel de MapMosaic, grâce à des cas d’utilisation précis.

Le multipléxage spatiale peut être très utile pour comparer différentes couches géographiques. Cependant, le multipléxage temporel pourrait être plus approprié pour la représentation des phénomènes dynamiques, puisque les changements peuvent être animés. Ceci pourrait être utile de façon particulière pour présenter des changements entre des images satellites, par exemple, pour montrer les effets du réchauffement climatique. Ainsi, la troisième contribution de la thèse est Baia: un cadre pour créer des transitions animées avancées, appelées plans d’animation, entre des couples d’images avant-après. Baia est basé sur un modèle de transition par pixel qui permet de créer des animations très variées. Il reste simple d’utilisation grâce à des primitives d’animations prédéfinies permettant de représenter des changements géographiques communs facilement. On décrit le modèle et l’outil d’édition d’animation associé et deux études avec utilisateurs. Le premier suggère que les animations créées avec Baia sont perçues comme plus réalistes et focalisent mieux l’attention des spectateurs que des animations basées sur un fondu homogène et le deuxième rassemble des retours sur l’outil d’édition d’animations.

PUBLICATIONS

1. Lobo María-Jesús, Appert Caroline & Pietriga Emmanuel. **MapMosaic: Dynamic Layer Compositing for Interactive Geovisualization.** International Journal of Geographical Information Science (IJGIS), 31(9), 1818-1845, 2017.
2. Appert Caroline, Chapuis Olivier, Pietriga Emmanuel & Lobo María-Jesús. **Reciprocal Drag-and-Drop.** ACM Transactions on Computer-Human Interaction (ToCHI), 22(6), Article No. 29, 36 pages, 2015.
3. Lobo María-Jesús, Pietriga Emmanuel & Appert Caroline. **An Evaluation of Interactive Map Comparison Techniques.** In CHI '15: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 3573-3582, ACM, 2015. Honorable Mention Award.
4. Lobo María-Jesús, Pietriga Emmanuel & Puech Claude. **Visualización de Big Data en Alta Resolución a Disposición de la Comunidad Científica y la Industria Chilena.** In Bits de Ciencia, 6 pages, Universidad de Chile, 2015.

1

INTRODUCTION

1.1. CONTEXT

Maps provide “*a presentation of the earth’s geographic surface*” through a graphical representation [153]. Before graphic displays, cartographers were constrained by the printed format. They aimed to create a perfect map for a given communication goal, such as road maps to help users navigate when traveling, or thematic maps for communicating geopolitical or economic statistics. Research faculties and national agencies were the firsts to see the opportunities of using computers to store and dynamically query geographic information. The first Geographical Information Systems were only intended as a software solution to accelerate the printed map production or as map measurement tools [94]. However, dynamic displays provide much more capabilities, and Geographic Information Science quickly became a research area on its own [49]. One of the questions that this domain addresses is how to take advantage of dynamic displays to better access and visualize geographic data. With interactive systems, users are no longer restricted to a simple and static representation. Interaction and different visualization techniques can help users understand and manipulate geographical data, both when using GIS to create maps, and when using those maps in diverse contexts.

Maps are used as visual thinking and analysis tools, as well as visual communication tools. They take advantage of people’s spatial reasoning abilities [150], serving four primary activities: *exploration*, *confirmation*, *synthesis*, and *presentation* to an audience [97]. *Exploration* maps enable users to gain insight about geographical data, and thus generate hypotheses about the available information. *Confirmation*

maps are created then to confirm these hypotheses, by *e.g.*, showing visual depictions of model results. *Synthesis* maps keep only the relevant information for the task at hand. Finally, *presentation* maps are created to present and convince other viewers of interesting findings. The goal of the map defines the most appropriate design and symbolization.

The field of geovisualization “*draws upon approaches from many disciplines, including Cartography, Scientific Visualization, Image Analysis, Information Visualization, Exploratory Data Analysis (EDA) and GIScience to provide theory, methods and tools for the visual exploration, analysis, synthesis and presentation of data that contains geographic information*” [37]. Geovisualization thus addresses the problem of choosing the correct design of digital maps depending on the type of users’ activities. In particular, interactive systems make users able to explore multiple representations quickly and might support visual thinking [129] during exploratory tasks. These tasks are especially challenging as they are open with no clear a priori objective. Digital representations can also be made dynamic to better convey a particular message on a presentation map.

Exploration maps are used as support for geovisualization applications in several specific domains, like crime analysis [133], geographic profiling [76] and visualization of citizen surveys [143]. In general, Geovisualization applications couple *multiple views* of the data in order to present multiple describing attributes of a geographic area. For example, GeoVISTA CrimeViz [133], provides support for conducting spatiotemporal analysis of criminal activity by coupling different views of the data: an interactive map view that enables users to change the base map and toggle different thematic layers, and a histogram with the number and type of crimes. This approach was useful for experts working with the data [134]. Some other tools also use *animation* to reveal spatiotemporal processes, for example, MapTime uses animation to study point based spatiotemporal data [144].

Presentation maps might also take advantage of interactivity, as they can convey dynamic information. For example, digital maps are becoming one of the most favored ways of creating presentation maps to tell stories that feature a spatial dimension. Examples range from following a character’s journey [122] to understanding the evolution of political conflicts [140]. The use of maps in data journalism goes beyond conflicts and politics: maps also provide good support to illustrate stories about environmental situations [23, 100], natural disasters [102], and people [19]. Map-based narratives [20] are typically organized as slideshows featuring animated transitions [89]. They help the audience focus on the important elements of the story

and feel more engaged, providing authors with means to convey their message effectively.

1.2. PROBLEM

Geographic Information Systems organize data into thematic layers that offer different perspectives on the geographic data. Those layers can be very heterogeneous in both nature and content, and can visually interfere with one another. They can hold any type of geo-located data, ranging from cloud cover to road networks and live traffic conditions. Distinct layers may contain different types of features (*e.g.*, roads, topographic contour lines), but may also show the same features, emphasizing different characteristics thereof (*e.g.*, road type vs. traffic conditions). For example, when using a GIS for risk management to identify bottlenecks in case of evacuation after an emergency, the system presented by Cova [33] uses two layers: one with the census data and one with the roads. Another example is when updating an existing map using OpenStreetMap (<https://www.openstreetmap.org>), existing objects in different layers, like building and roads, are overlaid on top of recent satellite imagery.

Geographical layers consist of either objects or fields. Objects are defined by their geometry, and are either points, areas or lines. These objects represent real world entities: buildings are usually represented by areas, and roads represented by lines. Objects are usually stored as vector data. Fields define spatial distributions over a geographical region without specification of the discrete objects they cover. As an example we can consider a satellite image, or a digital elevation model: there is a value for each pixel that corresponds to a specific geographic location, but there is no information about what kind of object this pixel belongs to. Fields are usually stored as rasters. Despite some theoretical efforts to unify these two families (objects and fields) [50], the intention and scale influence whether the geographical world is perceived as objects or fields [32]. Besides, most existing tools keep the distinction in the operations that they offer to users. For example, QGIS (<http://www.qgis.org>), provides two different set of operations for handling specifically raster or vector data, requiring users to make the distinction to manipulate their data.

Users often need to correlate data from different geographical layers varying in diverse aspects such as nature or theme [41]. One of the most popular examples dates as far back as 1854, when John Snow traced the source of a cholera outbreak to a water pump by plotting deaths on a map of London [80]. Figure 1.1 displays the map, where the deaths are presented as stacked bar plots per address, perpendicular to the streets. While this example relies on a single map, other scenarios require relat-

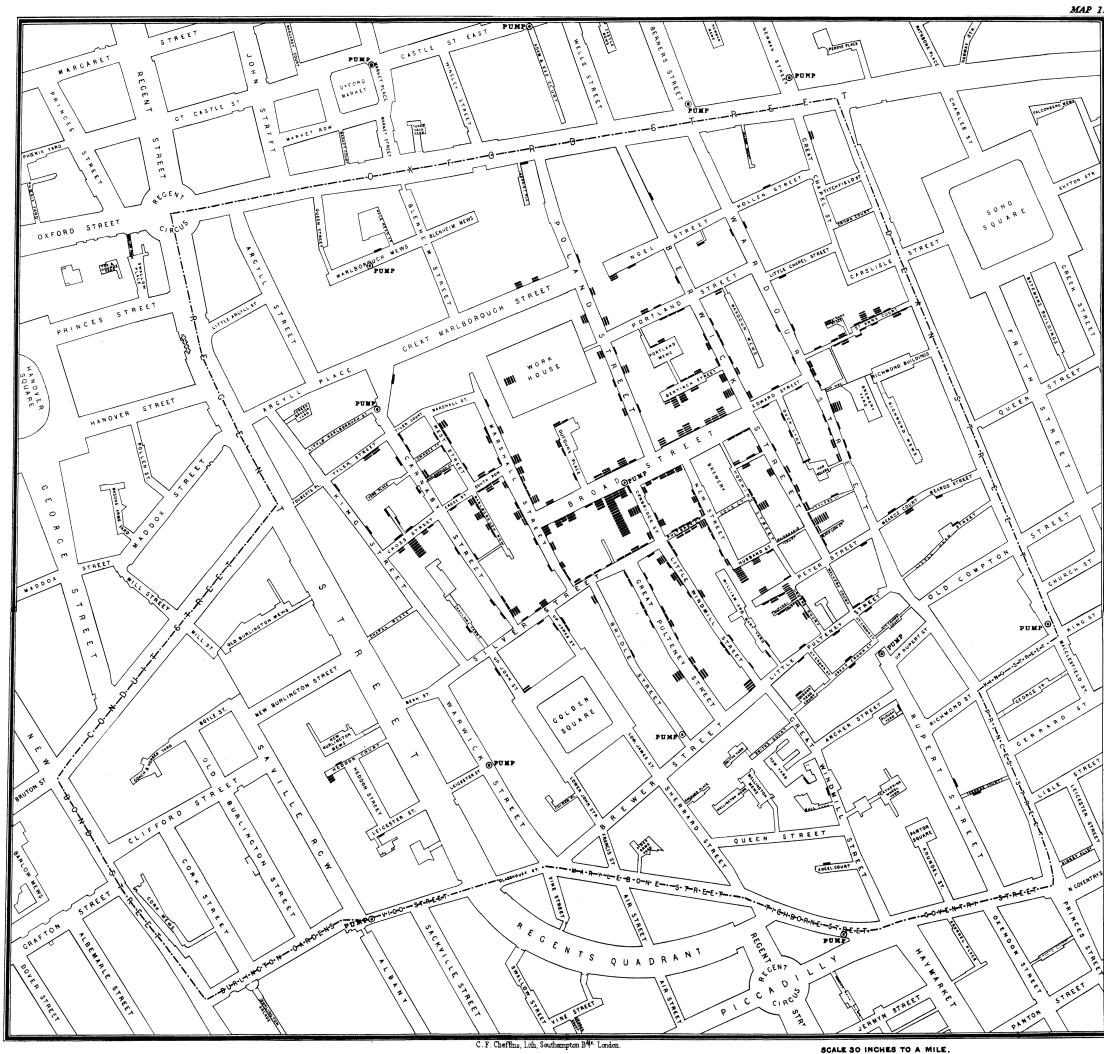


Figure 1.1: In 1854 Jon Snow mapped the deaths of cholera to trace the source to a water pump.

ing different maps across different data types. For example, to explore the evolution of a geographical entity, users might need to compare two raster layers containing satellite images taken at different times. This is the case for crisis mapping, after a natural disaster, images before are compared to images taken after to map the damaged areas. Also, two vector objects might be compared to find a relationship between the same attribute across different geographical entities. For example, users compare the number of votes per city for an election or a census to study possible spatial relationships [108]. Finally, some tasks require the combined use of both vector and raster information, like the already mentioned example of OpenStreetMap where users compare recent satellite imagery to the vector database to update the map.

The means to combine geographical layers in order to correlate and make sense of all the very heterogeneous datasets mentioned before remain limited. Consider-

ing the simple case of two simple layers: one base map (*i.e.*, a reference map showing only essential data that provide strong orientation cues, such as administrative boundaries) and an additional layer with sparse symbols (like touristic points of interest), combining the two layers can be as straightforward as overlaying the symbols on top of a topographic map, like many web-based mash ups [161]. However, when combining denser layers or comparing feature-rich maps of the same region, cartographic representation alone might be insufficient to access, correlate, and make sense of the heterogeneous datasets. User interaction can be seen as complementary to visual representation and helps alleviate this problem. It enables, for example, users to adapt the visual representation so as to emphasize relevant subsets of the data depending on the task at hand [2, 39, 127, 129]. Two main approaches exist for combining layers: multiple views can co-exist in the same space, or layers can be sequenced over time through animation.

Some geovisualization systems provide elaborate interactive data visualizations (see [24, 143, 145, 161] for representative examples), but those systems are domain-specific and they mostly enable interactions dedicated to the specific data managed by a particular application. More general geovisualization examples rely mostly on interaction techniques based on information visualization like multiple coordinated views, linked by highlighting and brushing [45, 53]. When considering general-purpose GIS user interface front-ends, from both research projects [35, 55] and industrial-strength products such as ArcGIS (<http://www.esri.com/software/arcgis>) and QGIS, once the map has been built, interactive navigation is simple and direct, but limited to basic interaction techniques such as pan & zoom, layer toggling or text search. These techniques consider the layers as flat images that can only be superimposed, juxtaposed, and sometimes drilled through. On the contrary, the creation and editing of more elaborate interactive transitions between layers, that take into account the semantics of the geographical information, is either cumbersome or not possible. This results in models of user interaction that offer poor support for the exploratory navigation and analysis tasks involved in, *e.g.*, simulation and planning, surveying & updating topographic features, or natural disaster management [76].

The existing techniques to combine multiple layers also offer poor support for the depiction of temporal evolution, such as comparing satellite images of the same region but taken at different times. These visualizations might be especially useful for displaying natural phenomena evolution. In fact, an increasing number of stories use remote sensing imagery, as both government agencies¹ and commercial businesses²

¹For instance NASA: <http://earthobservatory.nasa.gov>

²For instance DigitalGlobe: <https://www.digitalglobe.com>

provide this type of imagery to the public, offering both past and recent images of many geographic areas. However, as for exploratory maps, the means to display evolution across satellite imagery remain in most of the cases limited to side by side comparisons. Animations would offer a better solution, as they have the potential to represent the evolution of geographical areas through time. However, the use of animation in spatial histories is often limited to vector data, and is usually limited to appearance and disappearance phenomena, or to a simple monolithic blending.

In summary, geographical layers are very heterogeneous in both nature (object or field) and content. These layers often need to be compared or related for diverse tasks. In particular, users often need to compare different representations of the same region, at the same scale. However, the means to combine them are often limited to simply overlaying them, displaying them side by side, or basic animations. These techniques do not fully support users in their exploration or presentation tasks.

1.3. RESEARCH QUESTION

Most existing solutions combining geographical layers either are domain specific or do not take into account the information contained in the layers. We argue that digital maps are highly-structured, semantics-rich user interface components that have the potential to offer more elaborate transitions between geographical representations [136, 157]. We define a transition as a multiplexing operation, that takes as input two or more map representations of the same area, and outputs an interactive graphical scene that either combines those representations spatially (space multiplexing) or sequences them using smooth animations (time multiplexing).

This thesis addresses the following research question:

Can we create interactive transitions to help users relate data from heterogeneous layers by taking advantage of geographical data characteristics?

1.4. THESIS OUTLINE

This thesis presents three main contributions: an *evaluation* of existing interaction techniques to correlate heterogeneous layers and two novel approaches to map multiplexing: *MapMosaic* and *Baia*. Figure 1.2 illustrates how these contributions explore the design of map multiplexing.

Before detailing those contributions, this manuscript starts with a presentation

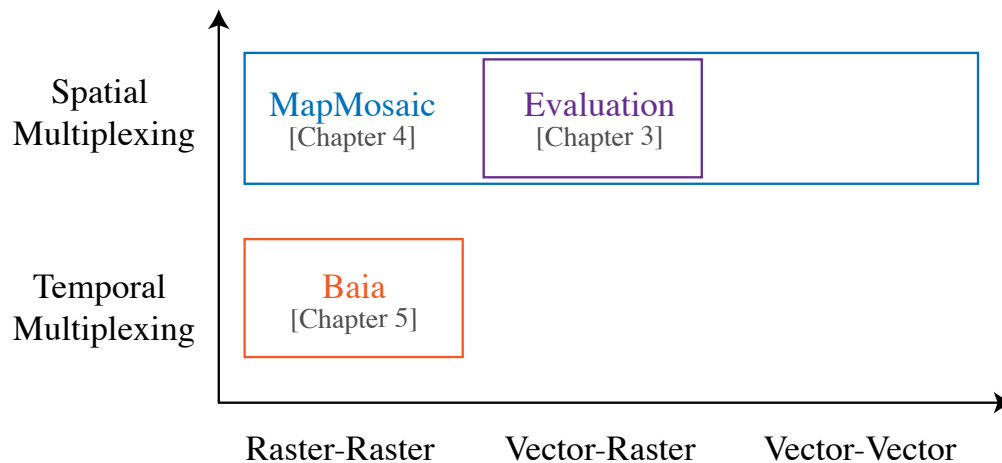


Figure 1.2: Classification of the contributions of the thesis according to the type of multiplexing and the kind of geographical data they handle.

of existing work in Geographical Information Science, Geovisualization and Human Computer Interaction that is related to transitions between geographical layers. First, we define what a multi-layer representation is, and present the tasks that require users to combine different layers in order to create or explore maps. Then, we describe the interaction and visualization techniques that exist to address this problem.

In order to better understand the limits of existing approaches, the third chapter presents our evaluation of existing interactive techniques for a map comparison task between a topographic map showing vector objects and satellite imagery. We begin by characterizing the techniques that we evaluate, and we report on a user study that showed that the most effective technique depends on the task at hand and that techniques that superimpose two representations are more effective than the ones that juxtapose them.

In the fourth chapter, we take into consideration the findings of this user study as well as interviews with experts working with multiple layers, to design *MapMosaic*. We present the dynamic compositing model that *MapMosaic* relies on, and how it takes advantage of layers' semantics and structure to enable advanced spatial multiplexing. We describe a prototype implementation, feedback gathered from workshops with GIS expert users, and an evaluation that suggests that *MapMosaic's* model is less cognitively demanding than existing models.

In the fifth chapter, we present an animation model to illustrate changes between before/after satellite images. We report on a user study that suggests that animations created using this model can be more realistic and better at focusing viewer's

1

attention than monolithic blending. We then present an authoring tool that enables a wide audience to create custom animations. A user study shows that the tool has a fast learning curve and that non expert users manage to create advanced animations after a short training.

Finally, we present the conclusion of the thesis and directions for future work.

2

RELATED WORK

In this chapter, we present the existing work related to multi-layer geographical representations. We start by describing the motivations to provide such representations by identifying the tasks where they are useful. Then, we present the existing techniques to combine different geographical layers: spatial and temporal multiplexing. Finally, we describe the current interaction techniques that enable users to manipulate those representations.

2.1. WHY PROVIDE A MULTI-LAYER REPRESENTATION?

Several representations exist to depict a geographical region. Geographical databases contain information about existing geographical entities, such as roads, buildings, administrative boundaries or points of interest. They can also contain fields providing information about elevation, or temperatures. Maps are created for a specific purpose, and depict information with a particular style or design, like topographic maps or thematic maps. In those maps, only a subset of the information contained in the database is kept, to support a specific activity. For example, MapBox (<https://www.mapbox.com>) proposes a map to facilitate street reading and another designed specifically for outdoor activities. The geographical database behind those maps is the same, but which entities are shown and how they are graphically rendered change. Besides vector and field information, realistic representations exist, such as satellite imagery or 3D imagery, like Google Earth. All of these representations can also be provided at different levels of detail, varying the resolution of raster images or generalizing the vector entities [148] in objects databases.

All these representations contain different information that might be better suited to different kinds of tasks [13, 41]. Cotlekin *et al.* [31] present an experiment where casual users were asked to choose between a topographic map and a realistic map for six different tasks related to route planning and tourism. Their results show that for most of the tasks users prefer topographic maps. However, they prefer a realistic map when exploring an area to find places of interest with specific characteristics, such as a good place to rest.

Even if a specific map exists for some tasks, for example, the MapBox outdoor map for hiking activities, this map might need to be combined with another, or with additional data. For some tasks, the required map and information not only depend on the task at hand but also depend on the user or the region of interest. We consider the scenario where a user wants to plan a hike. He needs a map showing with great detail the hiking routes but he also wants to be able to explore the area using satellite imagery to find good places to take a break, like the task mentioned before. In that case, a composition of both maps, depending on the geographical area might be the most suitable map. Several other applications require relating different and heterogeneous layers, for example, emergency response & crisis management, where multiple data sources have to be correlated for damage assessment and coordination of field agents. Another example is studying the evolution of urban areas or melting glaciers by comparing old topographic maps to recent satellite imagery.

A comprehensive list of tasks involving multiple map layers is not readily available in the literature [54]. We review existing taxonomies in both the geovisualization and information visualization communities to identify tasks where relating multiple layers depicting the same geographical region might be useful.

Roth [126] categorizes these taxonomies into *objective-based*, *operator-based*, and *operand-based*. *Objective-based* taxonomies identify the kind of tasks users may want to accomplish using the interface, *operator-based* identify the operations used to achieve those tasks, and *operand-based* the objects these operations and tasks act upon. In this section, we present some examples of *objective-based* taxonomies. A more detailed review can be found in [17] and [126].

2.1.1. INFORMATION VISUALIZATION TAXONOMIES

To identify tasks where relating multiple representations is needed, we start by studying the more general information visualization taxonomies. Even if these tasks do not specifically address geographical information, they can be applied to it.

In his survey, Roth [126] identifies two main tasks present in information visualization typologies: *identify* and *compare*. *Identify* refers to the examination of an object in the visualization, and *compare* to the exploration of similarities and differences between multiple geographical objects. Brehmer *et al.* [17] separate the tasks also according to their level of specificity: from less specific, considering why users *consume* and *produce* the visualization, to more specific, considering why users *query* the visualizations. For the less specific purpose they propose three reasons: *present*, *discover* and *enjoy*. *Present* refers to communicating information and *discover* to generating and confirming hypotheses. They might be considered equivalent to the map activities presented in the introduction: *presentation* and *exploration*. *Enjoy* refers to a casual use without a specific prior objective. For any of these objectives, users *search* elements of interest and *query* to *identify*, *compare* or *summarize* them. While searching, users might *explore* the dataset to look for a precise set of characteristics, without having prior knowledge about the identities of the specific objects or their location. These kinds of tasks (e.g., *compare*, *summarize*) might benefit from several representations of the data.

Yi *et al.* [164] present a taxonomy to classify users' intent when using interactive visualizations in seven categories: *select*, *explore*, *reconfigure*, *encode*, *abstract/elaborate*, *filter* and *connect*. *Select* is similar to the *identify* task mentioned earlier and involves marking an object of interest. *Explore* refers to changing the subset of visible data, for example panning in maps. *Reconfigure* enables users to alter the spatial representation of the visualization, for example rotating a 3D visualization. *Encode* refers to altering the visual representation, for example changing from a graphical representation to another, e.g., from a pie chart to a bar chart. *Abstract/Elaborate* refers to changing the level of detail of the map, for example by zooming, *filter* also changes the level of detail, but based on a criterion. Finally, *connect* refers to linking elements across different views. Several of these tasks, such as *connect* and *abstract/elaborate* involve different representations of the data.

2.1.2. SPATIAL INFORMATION TAXONOMIES

Some objective-based taxonomies explicitly address spatial data. As for the information visualization taxonomies mentioned earlier, *compare* is also a common task [34, 128], as are *identify*, *(re)order/(re)sort* and *abstract/elaborate*. Crampton [34] also adds a more complex task: *cause/effect*. This task involves finding a relationship between two different representations, for example, exploring high values on a scatter-plot view, to see if those values present a spatial relationship on a map view.

Roth [128] derives a taxonomy based on objectives, operators, and operands for interactive maps through a card sorting experiment with experts. He states several objectives: *identify*, *compare*, *rank*, *associate*, and *delineate* for three operand primitives: *space-alone* (geographic component of the visualization), *attributes-in-space* (how an attribute varies in space) and *space-in-time* (how geography varies across time). However, in most of the cases, the specific tasks are aimed at a single map representation, for example, comparing one attribute across two different geographical entities. The only objective in which several map representations are needed is when comparing space-in-time, to study some entities' evolution.

Andrienko *et al.* [3] present a classification of tools and techniques to explore specifically spatiotemporal data according to three kinds of changes occurring over time: *appearance* or *disappearance* of objects, changes of *spatial* properties and changes of *thematic* properties. They present a task typology based on identification and comparison of entities, across time and space. The tasks are defined based on the question they try to ask: *when*, *what* and *where*, and their level of detail: *elementary* for one single object or time moment, or *general*, for all the objects and moments. The tasks that address *when* correspond to evolution across time, for example comparing behaviors of the same geographical location at different times, such as weather evolution during different months. These tasks involve correlation of representations, where each representation corresponds to a particular moment in time.

Elias *et al.* [41] specifically address the problem of the tasks that involve relating multiple rendered maps (*i.e.*, raster maps, with no vector information available) through interviews with experts. The authors categorize maps according to two axes: *time* (the moment that the map depicts) and *category* (the type of objects visible on the map, for example, natural or human-made features). Maps can be more or less *familiar* to each user, according to their knowledge. They derive three main tasks from the expert's interviews: *familiarization*, *evolution*, and *fusion*. *Familiarization* refers to relating a familiar map to an unfamiliar one. For example, relating an old map, that contains the target information, such as the location of a shipwreck, to a current one. *Evolution* refers to studying evolution across time, for example, to explore the changes of a shoreline. Finally, *fusion* refers to the synthesis of two unfamiliar maps, the example of the John Snow map described in the introduction falls in this category.

To our knowledge, Elias *et al.* [41] are the only ones to mention tasks that require relating multiple cartographic layers explicitly. However, several taxonomies mention tasks that could involve relating multiple representations, such as *compare* or *filter*. Also, experts working with geographical information systems often need

to relate different layers for their work, as we will show in Chapter 4. These tasks require visualization techniques to facilitate relating and comparing different geographic representations. We review now the existing approaches that are based on *spatial multiplexing*, when layers are combined into one interactive scene, or *time multiplexing*, when an animation is used to transition from one representation to another.

2.2. HOW TO COMBINE DIFFERENT REPRESENTATIONS?

2.2.1. SPATIAL MULTIPLEXING

CLASSIFICATIONS

In this section, we first describe some existing categorizations to be able to identify and classify the existing spatial multiplexing techniques.

Javed and Elmqvist [73] present a categorization of techniques to create composite visualizations and Gleicher *et al.* [47] present four different designs to support visual comparison between visualizations. Both works consider *Juxtaposition*, *i.e.*, putting representations side-by-side, and *superimposition* *i.e.*, overlaying one representation on top of another. Javed and Elmqvist propose also two other operators: *overloading* and *nesting*. *Overloading*, like *superimposition*, uses the same space for both representations, but without keeping a spatial relationship between them. *Nesting* uses the marks in one visualization to host the other, for example using bar charts instead of points in a scatterplot. Gleicher *et al.* also include *explicit encoding*, created by computing a relationship between the objects to be compared and depicting it in the representation.

Geographical representations are inherently tied to their spatial location and dimensions, so we will focus on the spatial multiplexing techniques that preserve them: *juxtaposition* and *superimposition*.

MULTIPLE VIEWS AND SMALL MULTIPLES

Different geographical layers can be spatially combined by juxtaposing them. Presenting multiple alternative representations of the same dataset enables users to get different perspectives and explore the data. In the general approach, several information visualization and geovisualization applications use multiple views to present different representations of one dataset, for example, different statistic representations. GeoWizard [45], a visualization tool to explore Sweden statistics provides four

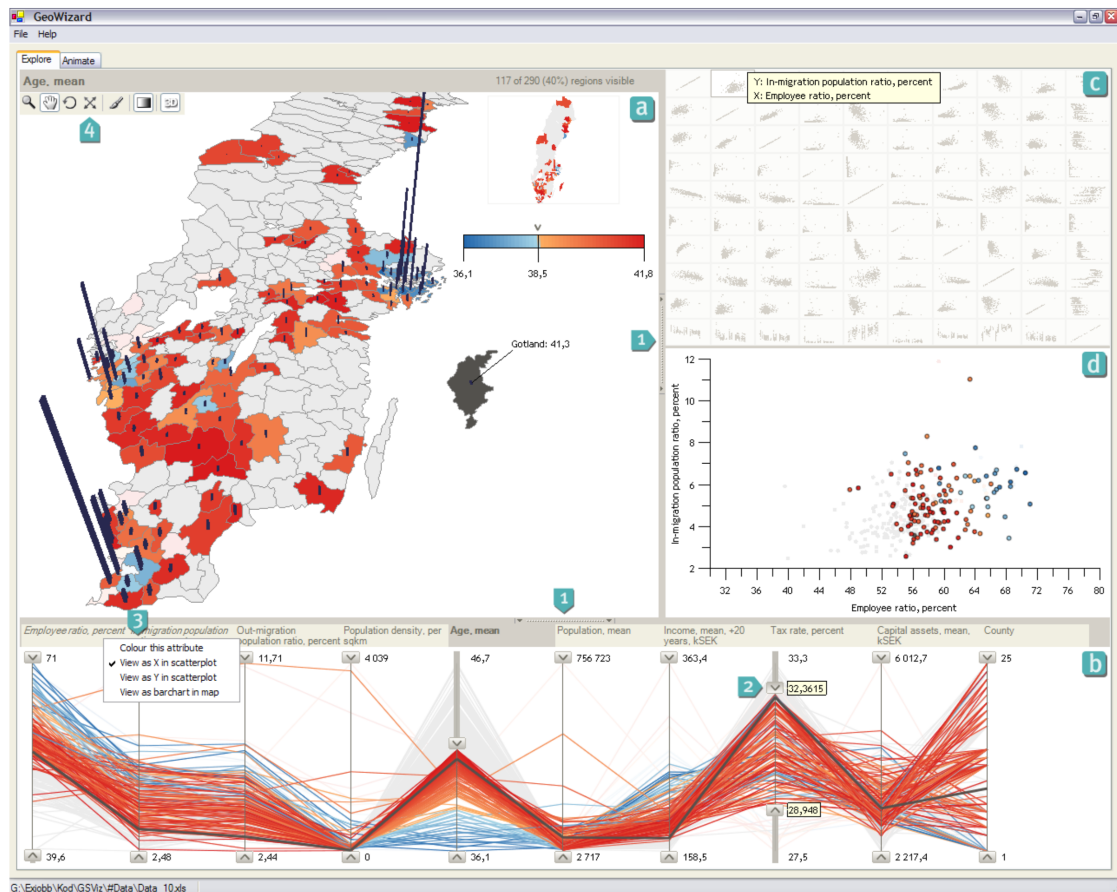


Figure 2.1: The Geowizard system use multiple views to explore Sweden statistics. Source [45].

dynamically linked views: a choropleth map and three different visualization for statistical parallel coordinates (scatterplot matrix and 2D scatter plot). Figure 2.1 shows this system.

Multiple views can also be used to relate different maps of the same region, for example displaying two or more satellite images of the same area taken at different moments side-by-side. This technique is often known as small multiples.

Tufte defines small multiples as “*a series of graphics, showing the same combination of variables, indexed by changes in another variable*” [149]. They use juxtaposition to compare and relate different data perspectives or data views. Small multiples are convenient when the information cannot be integrated into the same view but might make comparison difficult as they reduce the size of each map and they introduce spatial separation [98].

One of the aspects to consider when using small multiples is their spatial arrangement. For example, GeoVista Studio arranges views in a matrix of different kinds of bivariate views [96] such as a *MultiForm Bivariate Small Multiple*. This matrix contains different bivariate views such as choropleth maps and scatterplots. The first

variable is the same for each visualization, and the column of the matrix determines the second one. Each row presents different types of views. This enables users to find relationships between the variables and their geographic location.

Another example is how Wood *et al.* [160] use small multiples to identify how the name position in a ballot for an election in London might affect the number of votes a candidate receives. They create a visualization that uses one small multiple per borough in London, and order in each of them the candidates votes according to their position on the ballot. This visualization enables them to find both an effect for the position on the ballot and the geographical location in the number of votes.

Meulemans *et al.* [103] study alternative arrangements for small multiples in a grid (for example, according to geography) and how gaps (or white spaces) can be used to convey information (for example, a white space can be used when no data is available for a given location). They found that arranging small multiples according to spatial properties can be helpful, but the optimal arrangement solution depends on the geographical data to be displayed.

Juxtaposing alternative representations of the data offers different perspectives and is useful in several scenarios. However, juxtaposition forces users to go from one view to the other, potentially causing problems of divided attention. Superimposition offers an alternative that does not suffer from this problem but might introduce visual interference, as representations compete for the same representation space.

OVERLAY AND MASH UPS

Spatially combining two aligned geographical layers can be as straightforward as superimposing the layers, plotting one on top of the other. GIS tools often rely on the metaphor of stacked layers and render several layers at a time, enabling users to vary the opacity of each layer. Web mapping services also provide possibilities now to bring multiple datasets together, through mash ups.

Mash ups are defined as the mix of several different projects into one. Google Maps can serve as support for geographical mash ups, as it enables users to import different kinds of data and plot them on top of base maps [104]. Base maps can be schematic, showing only essential data that provide strong orientation cues such as administrative boundaries, relief and water bodies. Base maps can also use orthoimages, providing a more realistic – but possibly less useful [150] – background. Using Google Maps to create mash ups is simpler and more accessible than using traditional GIS and it has been used in many scenarios, like disaster mapping and real state [78, 104]. Roth and Ross [132] used it to create DC CrimeViz: an application

based on Google Maps to analyze crime activity supporting multiple context layers, animation, and filters.

2

Wood *et al.* [162] overlay spatiotemporal data representing spatial queries to a US-based mobile service on top of Google Earth in order to assess the possibilities of Google Earth as an exploratory tool. Besides plotting the queries source geographical locations and their results, they use other visualizations like tag clouds and tag maps to show the relationships between the queries content and the spatial locations. They conclude that mash ups enable more rapid prototyping and early exploration than developing more low-level and target-specific solutions.

Some tailor made geovisualization applications also use superimposition to combine complementary layers. One example is the application introduced by Roth *et al.* [131]: a web-based visualization to depict water level changes related to climate change for the Great Lakes region of North America. The system overlays several layers, such as water depth and the exposed area of the lake, on top of a base map which can be satellite imagery or a topographic map. The prototype of the tool was evaluated with experts working in various domains, and they appreciated the presence of different layers, as they might be useful for different tasks. For example, satellite imagery helps to identify the land use, and topographic maps depict the landforms. Figure 2.2 shows a screenshot of the system.

Google mash ups and some geovisualization applications provide support for bringing data from heterogeneous sources together through superimposition. However, they work only in cases where the data to be overlaid is sparse and does not interfere with the background data in the base map. To overlay another opaque map, for example, a rendered topographic map on top of satellite imagery, or a heat map depicting the relief would be unsatisfactory, as one layer would completely occlude the other. More advanced techniques to overlay these kinds of data are detailed next.

STATIC ADVANCED COMPOSITING

Recent work in cartographic research has produced methods to combine different maps by overlaying ortho-rectified imagery and topographic maps or other vector data. Such methods make use of digital compositing [117], where an image is obtained by assembling multiple images using alpha blending to control the visibility and opacity of the source images in different regions of the final image.

Luz and Masoodian [95] study the effect of using translucency to combine two superimposed layers. They vary the foreground layer opacity and ask participants to read both the foreground and background maps. They found the optimal level of

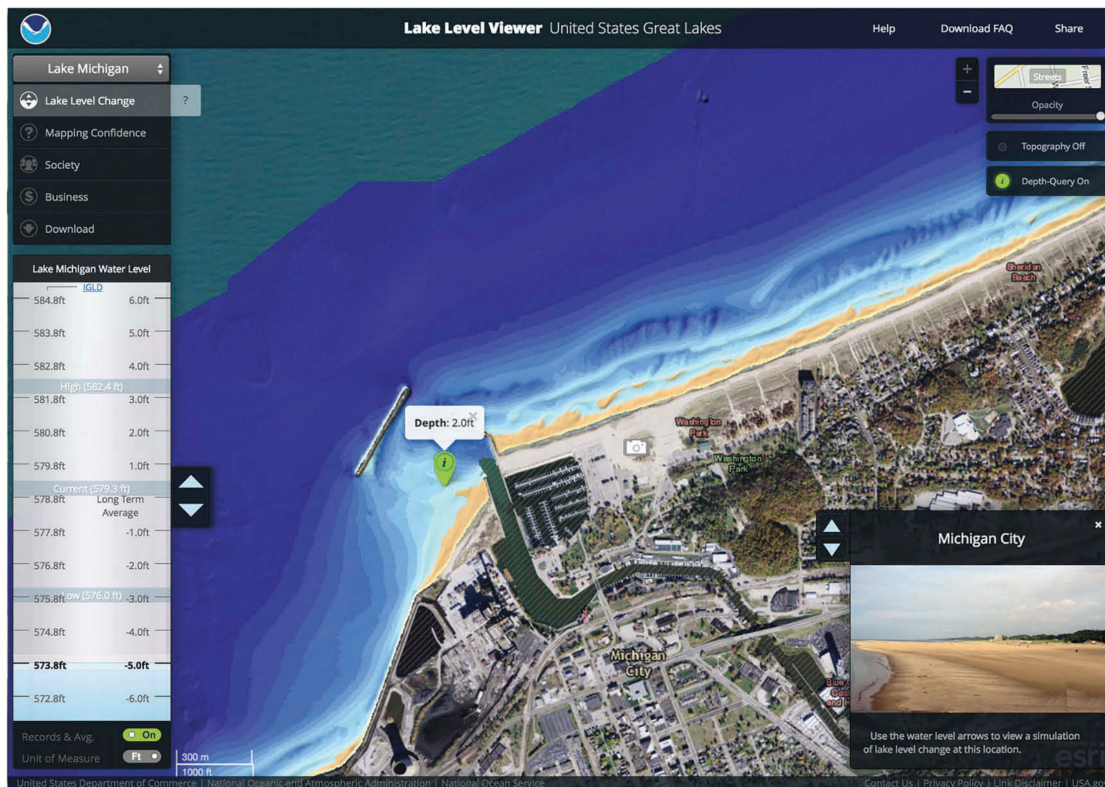


Figure 2.2: Web-based geovisualization application that uses superimposition of layers to depict water level changes. Source [131].

translucency for the foreground map to be 50% (for both kinds of tasks) and that the optimal level also depends on the kind of map displayed in the background layer, with backgrounds dense in content making the task more difficult.

Raposo *et al.* [120] compare different designs that overlay topographic map data on an orthoimage background. They propose two new designs that aim at increasing realism in maps while keeping topographic information readable. The first one depicts topographic features using saturated color with a degree of translucency on top of an orthoimagery background, and a dark translucent shaded relief. The second one increases realism by using a grayscale orthoimagery, and a colored shaded relief. Both designs were compared to the US standard product (topographic features overlaid on top of an orthoimagery background without altering the symbolization and without relief depiction) by asking participants to answer questions related to landscape features. Participants were more accurate with both novel designs than with the standard product. The authors think that their superiority might be due to the relief depiction and argue that the more readable design also depends on the characteristics of the depicted location.

The described studies suggest that the effectiveness of compositing techniques



Figure 2.3: Continuum from a topographic map to a realistic map. Source [67].

depends on characteristics of the represented region. Hoarau *et al.* [66, 67] propose more advanced control over the photo realism of maps by creating hybrid visualizations that mix realistic orthoimagery and symbolized vector data in one single visualization. The authors use *interpolation blocks* that interpolate between colors and textures of the symbolized vector data and the orthoimagery. For the final result, each geographical layer (*e.g.*, roads, water) can be interpolated differently, according to the cartographer's design. This results in continuums from one representation to the other, as depicted in Figure 2.3.

On the industry side, MapStack (<http://mapstack.stamen.com>) also provides some compositing capabilities depending on the type of object depicted. Users can choose from a set of map styles and limit them in scope to three predefined areas (water, buildings, parks). They can also vary their opacity. Such simple approaches target lay users who want to easily create basic personalized maps [6, 27]. Regarding more advanced GIS tools, such as QGIS or ArcGIS, users are able to select the layers to be rendered and vary their opacity. They can mask raster layers with some vector object, but those operations are often cumbersome and do not support flexible manipulation.

The resulting spatial techniques might be quite elaborate in terms of graphics rendering, but are designed to produce static maps. They do not support interactive exploration tasks, like the ones we will discuss in the next sections.

2.2.2. TEMPORAL MULTIPLEXING

An alternative to *spatial multiplexing* is using *temporal multiplexing* by sequencing different representations in time through animations.

TYPES OF ANIMATION

Animations, and in particular map animations, have been used for diverse purposes. We report on some studies that classify the different roles of animation in order to see when they are useful for relating different geographical layers.

In user interfaces in general, animations serve different purposes. Chevalier *et al.*'s [26] list five main uses of animation: *keeping in context*, *teaching*, *improving user experience*, *data encoding* and *visual discourse*. *Keeping in context* serves in navigation tasks like panning and zooming by animating the camera from one point to another instead of moving it abruptly. *Data encoding* helps reveal data relationships, for example, to understand causality between two phenomena.

Several studies try to understand how animation might be useful in geographic visualization. Harrower and Fabrikant [62] identify two types of animations in cartography: *non temporal animation* and *temporal animation*. *Non temporal* animations show attributes changes that are not temporal, for example morphing projections, like the animations provided by the web application WorldMapCreator (<http://www.worldmapcreator.com/>). *Temporal animation* display dynamic events happening over time. Harrower states that "*timeless maps are problematic because they portray the world in an 'eternal present' and eliminate the concept of process*" [62] suggesting that *temporal animations* might be particularly useful for displaying geographical processes.

Lobben [90] classifies cartographic animation into four categories: *time-series*, *areal*, *thematic* and *process* animations according to three variables that might change during the animation: *time*, *variable* and *space*. *Time-series* animations are static in *variable* and *space*, meaning that the symbols depicted do not change and neither does the region represented. *Areal* animations present only a dynamic *space*, by changing the observer's viewpoint. For example, "Fly-by" animations smooth the change from one position to another. *Thematic animation* involves changes in *time* or *values* which are reflected in changes in the symbols (for example, an animation of choropleth maps through time). Finally, a *process* animation depicts dynamic *time* and *value* by representing motion and trajectory (for example, the evolution of the shape of a geographical entity).

Animations in geographical visualization and user interfaces might be classified

according to their purpose and the characteristics they animate. As proposed by [62], we classify them according to *non temporal animations* and *temporal animations*, and report on how *non temporal animations* relate different representations of the data and how *temporal animations* display evolution across representations showing different moments in time.

NON TEMPORAL ANIMATIONS

Non temporal animations are used in general to help users relate different representations. We present some examples where they have been used for *keeping in context* and *data encoding* [26].

One of the most often encountered examples of *non temporal map animation* is fly-by animations, where the camera position is animated from one focus region to another. For example, when selecting a location using Google Earth, the camera will be animated to be centered on that position. Another example is how Matsuoka *et al.* [101] use a fly-by animation of before and after scenes of the Iran Earthquake in 2003 to better estimate and illustrate the damages.

Heer *et al.* [65] study animated transitions between statistic graphics. They introduce Dynavis, a visualization tool that enables several animated transitions such as *timesteps* that animates graphical depictions to reflect variations in time such as changing the height of a bar plot according to its new value. It also animates changes of visualization type, such as a bar chart being transformed into a donut chart. They conducted two experiments and found that animated transitions are advantageous for syntactic and semantic tasks, and that staging (*i.e.*, sequencing different animations in time) in the animation can also be beneficial.

Transmogrifiers [18] enable users to transform a visualization from one source shape to a target shape through an animated transition. The shapes are specified by touch interaction. The resulting visualization enables users to more easily compare different shapes, for example, to compare river lengths. Morphing algorithms have also been used to introduce advanced visualizations through analogies with more common visualizations, for example transforming a data table into a parallel coordinates plot. This teaching method enables novice users to use advanced visualizations [135].

TEMPORAL ANIMATIONS

Animations are natural candidates for representing spatiotemporal data and processes [60]. Tversky *et al.* [151] state that: “*the structure and content of the exter-*

nal representation should correspond to the desired structure and content of the internal representation.”, meaning that changes in display time should be appropriate to depict real changes in time. Early exploratory visualization applications such as HealthViz [98] have been using basic forms of animation to visualize time-series, swapping between snapshots sequentially. More recent examples, such as the wild-fire simulation by Kim *et al.* [79], interpolate between snapshots to create smoother animations.

Harrower [58] introduces a geovisualization tool that uses animation to explore a time series of satellite imagery for change detection. He argues that common change detection methods only quantify the changes, but do not reveal the processes involved and that using animation might show change patterns not visible otherwise. VoxelViewer enables users to set both the spatial and the temporal resolution, as some patterns might be visible only for specific periods, at specific resolutions. He showcases his system with NDVI data in the Sahara desert.

Other applications combine animations with other views. Blok *et al.* [12] tried to synchronize two animations to find relations between two spatiotemporal phenomena. They tested different methods (overlaying the two animations or juxtaposing them) in various scenarios such as studying the relationship between rainfall and vegetation. Both animation views can be interactively controlled to facilitate comparison. Opach *et al.* [107] link an animation view with other components that can be both animated, for example showing an overview of the same animation, or static, for example depicting a fixed moment to study landscape evolution.

Animation can also be used to tell spatial stories. Vector-based story maps can be created with tools conceptually similar to slide show presentation authoring applications. These applications feature means to easily create smooth transitions between slides, and to animate changes to the slides’ contents, which consist of the addition or removal of vector graphics objects, or the modification of their visual properties. Some web-based tools, developed by, *e.g.*, ESRI [43] or Northwestern University [82], have been specifically designed for authoring story maps, enabling authors to illustrate elaborate stories.

DESIGN CONSIDERATIONS

Different parameters might influence how effective an animation is for a particular task. We review research projects that have studied which factors are important when designing animations, to inform our design of animations to combine geographical layers.

Maggi *et al.* [99] compare animations with different characteristics for an air traffic control task where airplanes are depicted as squares. Participants have to find the airplane that is accelerating. They vary the speed of aircrafts and whether the animation is dynamic or semi dynamic (refreshed once every four seconds) and compare performance across different levels of expertise. They found that experts are as efficient with dynamic and semi dynamic displays, but novice were more accurate with semi dynamic displays. For dynamic displays, they focused on the most salient element *i.e.*, the plane moving faster, rather than the target of the task, the plane that was accelerating.

Defining the transitions between two maps, *i.e.*, how to interpolate between two snapshots of the animation is not clear either [7]. Transitions might depend on three aspects of the variables depicted: *level of measurement*, *visual variables*, and *classification*. For example, the *visual variables* can be classified as quantitative and qualitative and are based on the visual variables used in static maps: spacing, size, perspective height, hue, lightness, saturation, orientation, shape, arrangement. The authors suggest that transitions should be aligned with the represented states in the legend, for example, to avoid introducing an interpolated color that does not exist in the legend. Some visual variables also seem more appropriate to animated transitions such as size and color, than orientation or shape. Dong *et al.* [36] further studies different dynamic symbols to represent traffic flow, comparing line thickness and color and find that line thickness is more efficient as dynamic variable.

Some studies also explore how animation compares to static representations such as small multiples. Griffin *et al.* [52] compared the use of small multiples juxtaposed in space to animation in order to detect moving clusters. They found that animations work better when the cluster is subtle, and enable participants to identify clusters faster than small multiples in space. Slocum *et al.* [144] propose both small multiples and animations in their MapTime system to visualize spatiotemporal data. Their interviews and focus group with users with different degrees of expertise suggest that animations and small multiples might be useful for different purposes: the former to visualize general trends and the latter to compare moments separated in time.

Lobben [91] argues that the efficacy of animations compared to static displays depends on data characteristics. She conducts an experiment where the Mongol history of military conquests is depicted either statically or dynamically and then asks participants to answer questions related to different properties changing over time: time, attribute and location. She found that participants in the animation condition answered more questions correctly than in the static condition, but the difference

is mainly for the time-related questions. This suggests that time is the property that might benefit more from a dynamic representation. This finding is aligned with Tversky *et al.* [151] congruence principle.

Even if these studies reveal a benefit for animations, they should be used with caution, as they have also been observed to cause problems related to visual perception and cognitive load. For instance, users may be blind to some changes because of divided attention problems [48]. Their working memory may also be overloaded by the continuous flow of information [59, 61]. In addition, overly complex animations may be difficult to understand. When creating an animation, designers should also follow Tversky *et al.*'s *apprehension principle*, and create animations that are “[...] *slow and clear enough for observers to perceive movements, changes, and their timing, and to understand the changes in relations between the parts and the sequence of events*” [151].

The studies reported suggest that temporal multiplexing can be useful in some scenarios, but the animations should be carefully designed. Using animations to represent spatial changes through time seems to be particularly useful, as long as Tversky *et al.*'s [151] principles are respected and the dynamic symbols are chosen appropriately.

Spatial and temporal multiplexing provide different alternatives for combining geographical layers. Spatial multiplexing uses superimposition, for example stacking layers with varying opacity, or juxtaposition, positioning representations side-by-side. Temporal multiplexing enables users to animate from one representation to another, in order to relate them, or to depict evolution over time.

2.3. HOW TO INTERACT WITH A MULTI-LAYER REPRESENTATION?

2

In order to facilitate using multi-layer representations, interaction techniques exist to relate the views. Most examples enable users to navigate through scale and between heterogeneous layers.

2.3.1. CLASSIFICATIONS OF INTERACTION TECHNIQUES

In order to study how interaction can be useful when working with multi-layer representations, we start by reviewing some categorizations of interaction techniques and their goals for interactive maps.

Crampton [34] defines four interactivity types for geovisualization applications: *Interaction with the Data Representation*, *Interaction with the Temporal Dimension*, *Interaction with the Data* and *Contextualizing interaction*. He describes specific operators for each one of the type. Interaction involving multiple data representations are found in the four types. For example, to study temporal changes, *Toggling* enables users to switch abruptly between layers representing different times.

Edsall [40] classify existing GIS techniques according to four different goals: compensate for the indispensable deficiency of a computer display, discover unobvious patterns in data and explore particularly large databases. *Focusing* serves for the first goal, to emphasize a specific subset of the data. The authors mention *Toggling visibility* of different layers as an interaction intended at exploring large databases, that enables users to compare different maps. *Brushing and linking* serve to select a subset of data through a spatial query, determined by the user through direct manipulation, for example using a lasso tool. The spatial selection will be then applied to all the linked windows.

In his review of existing taxonomies of interaction techniques, Roth [126] mentions that *brushing*, *focus*, *highlighting* are the most common interaction primitives in information visualization and geovisualization. The other primitives can be categorized into manipulating the representation and manipulating the user's viewpoint. In his empirically-derived taxonomy [127], he classifies the operators as *enabling* and *work* operators. *Work* operators can be then classified as operators to manipulate the *kind*, *layout*, and *order* of maps, manipulate the *design* of the map, manipulate the *user's point of view* and finally further *examine features* within the map. Some operators especially support combining different layers, for example *arrange*, to manip-

ulate different alternate views, or *overlay*, to adjust the features that are displayed on the map.

We are interested mainly in techniques that enable users to transition between geographical representations. We start by reviewing techniques that manipulate the user's point of view through multiscale navigation. Then, we review techniques that manipulate the design of the map, to enable navigation between heterogeneous layers.

2.3.2. MULTISCALE NAVIGATION

One of the most closely related subject to our problem in the HCI field is multiscale navigation, which can be seen as navigation in a representation that features multiple layers with varying levels of detail. Multiscale navigation enables users to navigate more efficiently in both space and scale. Three main techniques exist: pan and zoom, overview + detail, and focus + context [29].

PAN AND ZOOM

Classical pan & zoom uses time multiplexing to present users with different views (possibly at different scales) on the map. Only one view is shown at a time, putting some burden on users' memory. These techniques are the most common in most current map interfaces such as GIS desktop solutions and web mapping applications.

Some studies explore how to design faster pan and zoom interaction techniques than the classical dragging and scrolling provided in most interfaces. For example, Cockburn et al [30] compares automatic speed-dependent zooming, a technique that adjusts the zooming rate to the speed at which the user scrolls, to traditional panning and zooming methods in standard interfaces. He finds that the speed-dependent zooming is more efficient in map browsing tasks than horizontal and vertical scrolling and clicking for zooming. Different inputs have also been explored. Bourgeois et al. [16] study how to couple pan and zoom actions. They compare one-hand and two-hand interactions and find that simultaneous interaction techniques (i.e. simultaneous pan and zoom) make target selection faster.

OVERVIEW + DETAIL

Overview + detail interfaces are dual-view representations of the same area at two different levels of detail. In the case of geographic data, they show both a detailed view and an overview of the map simultaneously, but in distinct presentation spaces. They are usually used in combination with pan and zoom techniques. For example, Google

Earth provides users with a zoomable 3D view of the globe and an overview in 2D where a red rectangle indicates the current viewport in the detail view. The viewport is updated each time the user pans or zooms. The study conducted by Hornbaek *et al.* [69] shows that this arrangement is appreciated by users in map navigation tasks. However, if the scale difference between the views is too big users might encounter difficulties as a panning action in the overview view might result in an abrupt change in the detail view.

Harrower [64] compares pan and zoom to overview + detail specifically for maps. He identifies different interaction techniques for pan and zoom such as the classic drag and drop and keyboard controls. The author suggests that the overview technique has potential to be more efficient than the other techniques as it enables users to pan and zoom with only one gesture (by drawing a rectangle in the overview). It also facilitates navigations tasks. However, the best interaction technique probably depends on the task at hand and the user characteristics.

Some other more advanced techniques use multiple views to enable multi scale navigation. The Polyzoom [75] technique enables users to create a hierarchy of multiple focus regions, where each region is a zoomed region of a parent viewport. A viewport might have several different children. This hierarchy enables multiscale and multiple focus navigation without using overlapping or distortion. An example of the technique is presented in Figure 2.4. The links between children and parent windows are depicted graphically. A user study showed that users were faster when using Polyzoom to perform a search task and a multifocus comparison than when using standard pan and zoom.

FOCUS+CONTEXT

Focus+context techniques display an inset zoomed region in the context. For example, the DragMag [155] technique enables users to create a zoom window on top of a base window. The magnified region is depicted graphically in the base image, and the user can change its position. When using a DragMag, there is a trade-off between the spatial separation of the zoom window and the region to be zoomed, and the occlusion the DragMag causes in the context [21].

Another approach consists in using lenses that magnify in place and distort the transition area to integrate the zoomed representation in the context. For example, the Sigma Lens framework [4, 113], provides a solution that enables implementation of different transitions between focus and context, based on displacement and compositing, such as fish-eyes that displace the pixels in the border of the focus region,

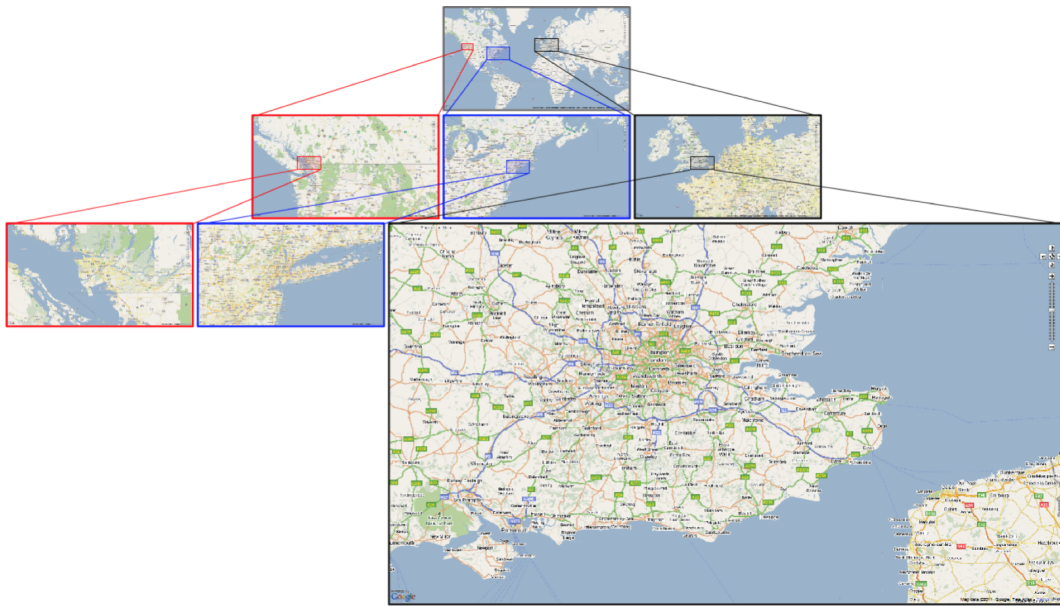


Figure 2.4: The Polyzoom technique enables users to create a hierarchy of multiple focus regions. Source [75].

and blending lenses, that alpha composite these pixels. Problems with this representation might arise, however, such as those due to spatial distortion, that can impede interpretation and target selection. Such lenses have also been used for geographic data, for example to annotate metro maps [15].

Other techniques, like JellyLens [115], take into account the object being inspected by dynamically adapting its shape to match the geometry of objects of interest and generating a smooth transition between the focus and the context like fish-eye lenses. This technique was observed to be more efficient than fish-eye lenses in a multi-scale visual search task. Figure 2.5 illustrates this technique. Lenses might take account of the underlying information by modifying their behavior as well. RouteLens [1] enables users to follow paths in maps more easily by attracting the lens to the followed path when moved. This technique proved to be more efficient than conventional magnification lenses in a path following task.

Mélange [42], uses a space folding metaphor from a 2D space to a 3D space to show distant focus regions simultaneously and keep the context visible. This technique is useful when comparing distant entities in the visual space. A user study revealed that Mélange was faster for estimating distances in a scenario of social network analysis than existing techniques, and users were more accurate when using it than when using a technique with a split view at different detail levels.

Multiscale navigation techniques are targeted at relating different views of the



Figure 2.5: The JellyLens technique adapts to the shape that is being inspected. (a) and (b) present two examples of fisheye lenses, and (c) displays the JellyLens technique. Source [115].

same geographical space at different scales. However, they do not address our problem of relating multiple representations of the same geographic region at the same scale. In the following section we review techniques that enable users to navigate between heterogeneous layers at the same scale by altering the design of the maps.

2.3.3. NAVIGATION BETWEEN HETEROGENEOUS LAYERS

RELATING GRAPHICALLY MULTIPLE REPRESENTATIONS

When dealing with multi-layer representations, the relationships between the different layers might not be obvious, for example the matching between objects across juxtaposed layers. We review below some interaction techniques that help users relate layers or representations on-demand.

Brushing involves both the selection of an object of interest and a manipulation of it to differentiate it from the other objects. Highlighting [38, 124], for example, uses visual methods to mark observations across different views. The most common approach is using color to differentiate the selected element, but some new static and dynamic approaches seem promising, like drawing lines between the different observations or using transparency. Griffin *et al.* [53] observed in their empirical study that *leader lines* (drawing a line between the mouse cursor and the different equivalent objects) is as effective as color highlighting for a search task across views and could thus be preferred when color is used to display an attribute.

Another example uses *focusing* and *brushing* in temporal displays. In the system to teach Earth Science introduced by Harrower *et al.* [63], an animated display combines several layers: cloud cover, land temperature, and water temperature in order to find spatiotemporal patterns and relationships between them. For example, how are temperature and clouds related, and when the relationship appears. They use

temporal *focusing* and *brushing*: temporal focusing enables users to restrict the animation to a specific period, and temporal brushing enables users to select a specific time of the day in a cyclic legend. This enables users to compare representations of the same hour at different days to look for patterns or to filter out unnecessary information.

Besides relating juxtaposed representations, graphic cues can be useful to relate off-screen content with the current display. In this case, we can consider the visible region and the rest of the map as different layers. Gustafson *et al.* [56] introduce a technique to visualize offscreen context by using *wedges*, isosceles triangles having their tip on the offscreen location and designed to not overlap. Other techniques, like Halo [8], also provide a cue about the distance of the offscreen object.

These interaction techniques serve to relate different views that present different representation of the data. They are specially useful in the case of juxtaposed views. Other techniques change the representation to help users to combine multiple layers through superimposition or animation.

CHANGE THE REPRESENTATION

Most GIS commercial systems, like QGIS and ArcGis enable users to combine layers by superimposing them, and toggling their visibility or adjusting their opacity. ArcGis also supports more advanced compositing techniques, like swiping between layers (*i.e.*, reveal the bottom layer by moving a slider, following a drawer analogy). Creating more advanced composites, for example using vector objects as masks is quite cumbersome as it involves interacting with a series of dialog boxes. These systems also support changing the symbolization of geographical information and querying the spatial database, but the interaction model is also cumbersome.

Lenses can also be used to display an alternative representation in a local area. Bier *et al.* [10] introduced the concept of *Magic Lens* that alter an existing graphical representation in a local region in different ways, for example, to preview changes in a graphic scene, or to facilitate selection through a representation that slightly shrinks cluttered objects. Since then, several visualization tools have used the concept of lenses [147].

Other lens-based techniques use semantic information to present a more relevant view in context. MoleView [70] enables users to explore large and dense 2d relational data (like graphs and images) by creating semantic lenses. They can filter elements based on attributes, pushing other items away, for example, filter flights in France according to their altitude. Users can also explore bundled graphs, by en-

abling them to bundle and unbundle them.

Wetpaint [14] enables users to *scratch* through layers. Initially thought for art restoration tasks, where users might need to compare the different layers in a painting, the system uses touch based interaction to reveal the stacked layers one by one. The underneath layer is revealed wherever the user touches the interface. If he touches again, the following layer is revealed. This technique seemed to be more efficient than a fading controlled by a slider between two layers (satellite imagery and topographic map) in a path tracing task.

Animation can also be used to combine representations. The simplest case, termed blitting, uses temporal multiplexing enabling users to toggle between two superimposed maps. The MapMorphing [121] technique presents a more advanced approach, by enabling users to morph between two maps significantly different from each other such as maps with different projections or very different symbolizations. Users can use a slider to go from one fully opaque representation of the first map to a fully opaque representation of the second map. A user experiment revealed that MapMorphing enabled participants to better understand the relationships between the maps than juxtaposing them.

Several interaction techniques exist to relate different geographic layers, for example to navigate between layers that represent the same space at different scales and to navigate between heterogeneous representations. These techniques usually do not take account of the underlying representation or are domain specific.

2.4. SUMMARY

Different geographic representations might be useful for different tasks. Cartographers specifically choose symbolizations for maps for different activities, in order to only keep the relevant information and make it as readable as possible. However, users and experts often need to correlate heterogeneous geographical datasets or different maps. The tasks in information visualization *compare* and *filter*, and the ones in the taxonomies targeted at spatial information, like finding *cause/effect* relationships, or comparing geographic evolution across time often involve multiple representations of the data.

The ways of presenting these multiple representations can be classified into *spatial multiplexing* and *time multiplexing*.

Spatial multiplexing combines several depictions in one scene, either by super-

imposing them or by juxtaposing them. Both strategies present drawbacks, superimposing several layers might lead to information occlusion and visual interference, and juxtaposition forces users to go back and forth between layers, causing problems of divided attention. The strengths of each strategy, and in which case each one might be more appropriate, are not yet fully understood.

Temporal multiplexing has been used to transition between representations, to enable users to better understand them, for example from one viewpoint of the camera to another, or from one type of visualization to a different one. It might be especially useful for representing temporal evolution, following Tversky's congruence principle [151]. However, animations are mostly used for vector data, or to present snapshots of realistic imagery, and their potential to represent processes has not been fully explored.

Interaction techniques can support users when manipulating these multi-layer representations. These techniques support multiscale navigation through pan & zoom, overview+detail and focus+context interfaces. Other techniques support navigating between heterogeneous representations, by relating them graphically or by changing the representation. However, most interaction techniques to navigate between different representations at the same scale do not take into account the semantics of the datasets presented, and restrain users to toggling layers or swiping between them [136].

3

AN EVALUATION OF INTERACTIVE MAP COMPARISON TECHNIQUES

Main portions of this chapter were previously published at CHI 2015 and got an Honorable Mention [93].

Diverse tasks in multiple domains, such as crisis mapping and crime analysis, require comparing different geographic representations. Spatial multiplexing through map composition supports comparison, unifying all these representations in a single one. It can yield powerful insights, and it can help users identify patterns and anomalies [97] or find casualty relationship. As more and more data become available, and as the level of sophistication of geovisualization tools increases, designing efficient interactive composition techniques has become a critical issue. Such techniques should enable compositions of base maps and data layers that optimize the legibility of the end result and that ease inter-layer comparison. So far, this problem has received relatively little attention. Several techniques exist, some of them are even widely used (Figure ??), but at this point most are little more than ad-hoc designs whose strengths and weaknesses are not well-understood.

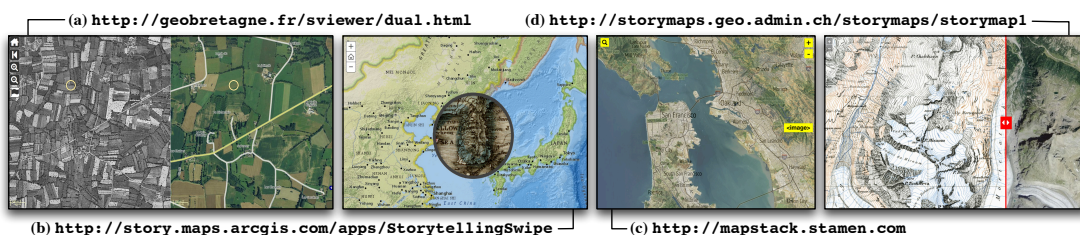


Figure 3.1: The four main composition strategies: (a) synchronized juxtaposed views, (b) magic lens, (c) translucent overlay, (d) swipe.

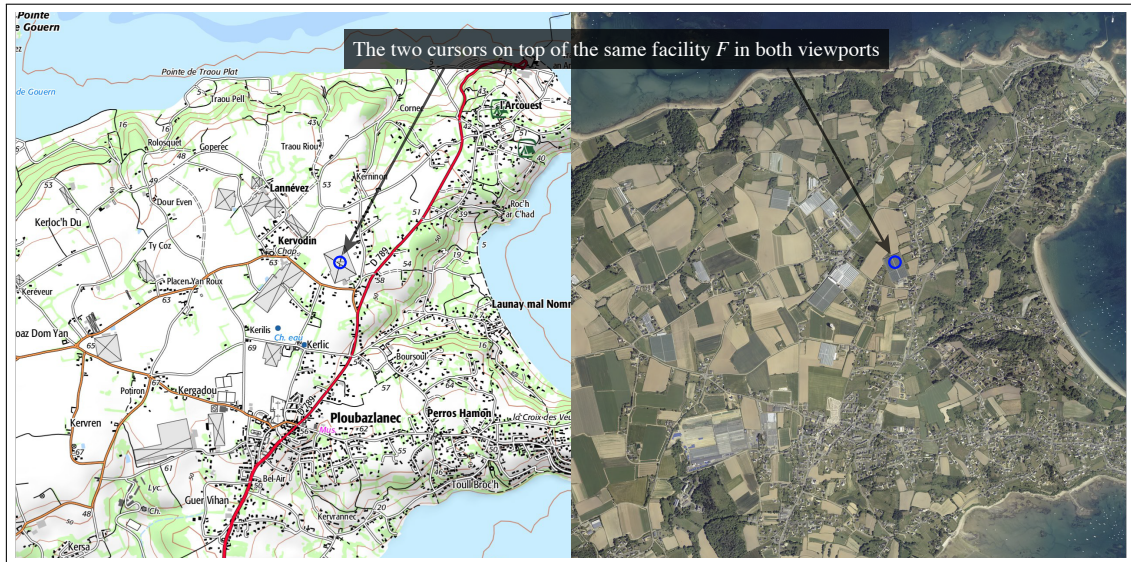


Figure 3.2: Technique IX: The two images are juxtaposed, showing the same geographical region in both viewports. Any pan and zoom action applies to both of them. Cursors are instantiated in each viewport. Their position is synchronized to help relate features in the two images.

Examples of these techniques are the already mentioned techniques such as, *e.g.*, techniques to select specific items in dense sets [163], techniques to visualize off-screen content [56], and techniques to enable multi-scale navigation, such as pan & zoom, overview+detail and focus+context [29]. These techniques provide users with multiple views on the same space (map or image) and can help them compare the content of these different views. But they are designed to show the *same region* at different scales, or different regions but of the *same map*. They are not adapted to our problem, which is to composite *heterogenous representations of the same geographical region at the same scale* so as to help users relate and compare features across them.

In this chapter, we identify and characterize techniques for the interactive composition of image layers in geovisualization applications (Figure ??). We empirically evaluate representative techniques on map comparison tasks using a set of 20 orthoimages and corresponding topographic maps, in which we purposefully introduced six types of differences with respect to the reference orthoimage. We present our operationalization strategy, and discuss the relative merits of these techniques in terms of visual interference, user attention and scanning strategy.

3.1. MAP COMPARISON TECHNIQUES

We surveyed the existing techniques by reading the literature, talking with GI science experts and reviewing prominent Web sites. A first observation is that those tech-

niques do not only apply to geographical maps, but to any images that are spatially aligned. Some techniques are used in Medical Imaging to explore brain scans, in Exogeology to compare surveys of, *e.g.*, planet Mars, made at different times or in different parts of the electromagnetic spectrum. The same applies to Astronomy, where comparisons also include looking at two images generated from the same FITS file (a high-dynamic-range format) using different transfer functions and color mappings.

Map and image comparison techniques predominantly employ two operators from Javed & Elmqvist's composite visualization design space [74]: juxtaposition and superimposition, which are called juxtaposition and superposition in Gleicher *et al.*'s taxonomy of visual designs for comparison [47]. These techniques can be found both in professional Geographical Information Systems, such as ArcGis and Qgis, and in web-based map applications, in the form of widgets that enable users to combine layers interactively. For example, QGis enables users to toggle layer's visibility and adjust their translucency. The web application *Remonter le temps* (<https://remonterletemps.ign.fr>) from IGN features five interaction techniques to compare maps across time.

The following four techniques are the most commonly used in map and image comparison tools. We characterize them in terms of visual interference and divided attention (which should both be minimized) and type of scanning. A motor-driven scanning strategy requires users to reposition elements on screen with their pointing device to compare different regions. A vision-driven strategy relies more on visual search and does not require so much interaction. The characterization is displayed in Figure 3.5.

3.1.1. JUXTAPOSE (JX)

As illustrated in Figures ??-a & 3.2, the two images can be juxtaposed, showing the same geographical region in both viewports. This technique does not cause any problem of visual interference, but suffers from problems of divided attention [57] as users have to repeatedly inspect each viewport. In addition, this technique can only display half the geographical area that superimposition techniques can show, requiring users to pan the view more often. Additional motor actions include moving the dual cursor to relate both views (Figure 3.5).

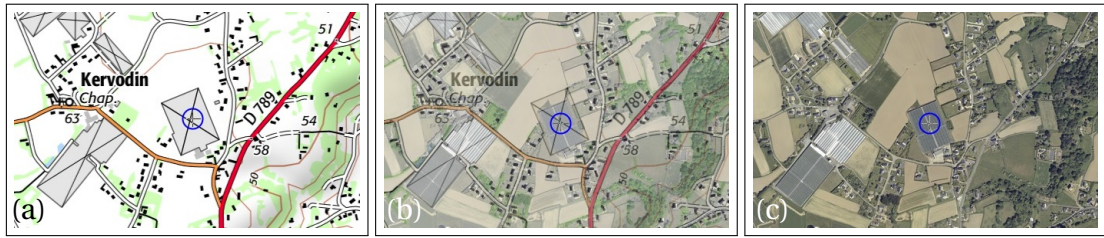


Figure 3.3: Technique OV: The two images are overlaid in the same viewport. Users can see the image in the bottom layer *through* the image in the upper layer by adjusting the latter's opacity. (a) Upper layer fully opaque: only the topographic map is visible. (b) At 50% translucency (adjusted using the mouse wheel), the orthoimagery is made partially visible through the map. This time, the two representations of facility *F* (the same as in Figure 3.2, under the cursor) are spatially aligned. (c) Upper layer fully transparent: the orthoimagery is fully revealed.

3.1.2. TRANSLUCENT OVERLAY (OV)

Techniques that use a superimposition strategy can display a geographical region twice as large, but have their own drawbacks. The first technique consists in overlaying both maps and enabling users to adjust the opacity of the upper layer, so as to see the lower layer through it (Figures ??-c & 3.3). As the Macroscopic [86], OV can cause significant visual interference depending on the maps' content and translucency settings, but it avoids the problems of divided attention from which JX suffers. Motor actions are limited to translucency control, enabling users to adopt a mostly vision-driven scanning strategy (Figure 3.5). Supporting image blitting is a simple matter of adding shortcuts to 0 and 100% opacity.

3.1.3. SWIPE (SW)

Another popular superimposition technique consists in swiping between the two maps (Figures ??-d & 3.4). As with OV, the two layers are superimposed, but in this case both remain fully opaque. Users "swipe" (or push and pull) the upper layer above the lower one, revealing more of one or the other layer. This technique minimizes both divided attention and visual interference (Figure 3.5). However, even if, as with OV, motor actions are limited to a simple uni-dimensional control, the nature of the composition creates a stronger dependency between motor actions and visual scanning.

3.1.4. BLENDING LENS (BL)

Yet another strategy when the two images are superimposed consists of using a magic lens [10] to show the lower layer in a locally-bounded region around the cursor (Figure ??-b). Figure 3.6 shows a somewhat more elaborate variant called a Blending Lens [113]. This variant features a smooth transition between the lens' focus and the

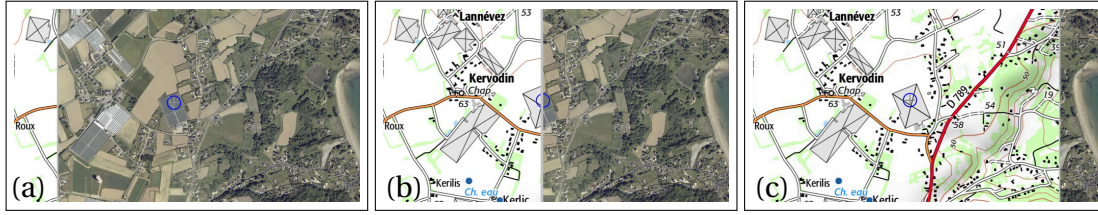


Figure 3.4: Technique sw: The two images are overlaid in the same viewport. Users can see the image in the bottom layer by swiping horizontally. (a) Moving the separator to the left reveals more of the orthoimagery. (b) & (c) Moving the separator to the right reveals more of the topographic map. Comparing a feature such as facility *F* across the two layers requires moving the separator back and forth in its vicinity (b).

3

context thanks to a transparency gradient. As most of the upper layer remains fully opaque, providing a stable context, this technique does not suffer from visual interference as much as OV. There is no problem of divided attention. However, users have to place the lens directly on top of the region in which they want to compare the maps, which forces them to adopt a mostly motor-driven scanning strategy (Figure 3.5).

3.1.5. OFFSET LENS (OL)

The last technique, called OffsetLens, is a variant of the DragMag [155]. We identified it as a potentially interesting solution in the design space. OL combines the juxtaposition and nesting operators [74], trying to strike a balance between visual interference and divided attention. As illustrated in Figure 3.7, it follows the same general approach as BL, except that the two images are juxtaposed instead of being spatially aligned. This makes it possible to look at fully opaque renderings of the region of interest from both maps, minimizing visual interference compared to BL. However, OL reintroduces the problem of divided attention, though less so than JX since the distance between the two viewports is much smaller. As with BL, users have to adopt a mostly motor-driven scanning strategy (Figure 3.5). As a side note, a typical DragMag would not have the secondary viewport follow the lens automatically, and would allow users to manually reposition it independently. But this additional flexibility would impose more motor actions on users to minimize divided attention.

3.2. USER STUDY

To our knowledge, the five techniques introduced above have never been compared empirically on well-defined map comparison tasks. The following studies are related to our work, but try to answer other questions. Luz and Masoodian evaluated the

readability of maps when translucent layers containing widgets (sliders) were superimposed on them [95], building upon Harrison *et al.*'s study of the effect of transparent overlays on user attention [57]. WetPaint [14] was compared to OV on a path following task mixing satellite imagery and a schematic subway map. Plumlee & Ware compared zooming vs. multiple DragMags for multi-scale comparison tasks [116]. Finally, Raposo & Brewer [120] compared eight different topographic designs that statically overlay orthoimages and maps.

Our goal here is to better understand the strengths and weaknesses of comparison techniques based on their characterization in terms of visual interference, user attention and scanning strategy. To this end, we designed a controlled experiment in which users had to compare orthoimagery with topographic maps. We chose this particular configuration because it is representative of many scenarios [41], including map making (in, *e.g.*, OpenStreetMap), cadastral & land management [84], and crisis management after a natural disaster, where satellite imagery can provide up-to-date views of the situation but is not sufficient on its own to perform analyses.

3.2.1. MAPS

We used 1:25000 topographic maps (SCAN25) and the corresponding orthoimagery (ORTHO), both produced by IGN (*Institut National de l'Information Géographique et Forestière*, in France), to create 20 perfectly-aligned pairs amenable to comparison;

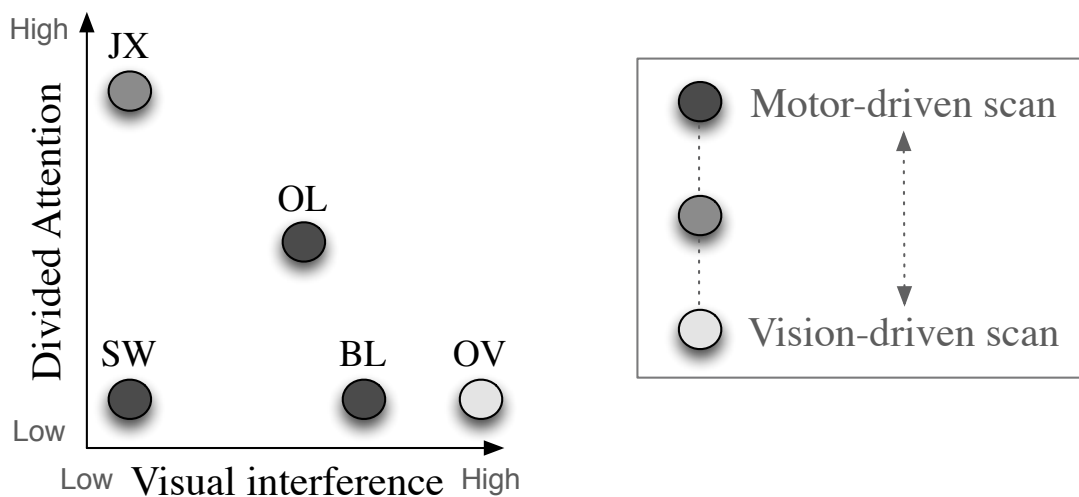


Figure 3.5: Coarse characterization of the techniques evaluated in the following study, in terms of visual interference and divided attention (which should both be minimized) and type of scanning. A motor-driven scanning strategy requires users to reposition elements on screen with their pointing device to compare different regions. A vision-driven strategy relies more on visual search and does not require so much interaction. This is a continuum: none of the techniques is either purely vision-driven or purely motor-driven.



Figure 3.6: Technique BL: When the two images are overlaid in the same viewport, the lower layer can be revealed in a locally-bounded region only – instead of the entire viewport – using a Blending Lens. The lens *punches a hole* through the upper layer, revealing the lower layer. (a) & (b) The lens approaches from the left the same facility F as in previous figures. (c) The lens is centered above it, revealing its appearance in the orthoimagery. (d) The lens’ translucency can be adjusted using alpha blending, so that both images are visible simultaneously, as with the overlay technique (OV).

see Figure 3.2 for a representative example. In each case, we used the orthoimagery as the reference image, and customized the map, purposefully introducing errors, as detailed below. This produced a total of 120 maps. All SCAN25 maps were manually edited in Adobe Illustrator, carefully removing features, introducing new ones, or modifying the geometry of existing ones so that those edits could not be detected without comparison to a reference. While using real maps in our study makes it much more challenging to control precisely all relevant factors, it is also essential in terms of external validity, as discussed later.

3.2.2. TASKS

As mentioned in the previous chapter, Elias *et al.* observed that many queries require correlation of multiple maps, and identified three main user-task categories in their taxonomy [41]: familiarization (“*relating the primary features of a map to a context understood by the map user*”), evolution (“*[synthesis of] maps in a common category across time periods*”), fusion (“*synthesis of two maps from categories that are both unfamiliar to the user*”). Many of their examples involve some sort of geometric comparison performed on either lines or surfaces. We position our tasks at this latter level of abstraction, so as to make our findings as generalizable as possible (given the map types considered).

Our tasks test the *compare* objective primitive in Roth’s taxonomy of primitives for interactive cartography and geovisualization [128]. These tasks are aimed at opera-

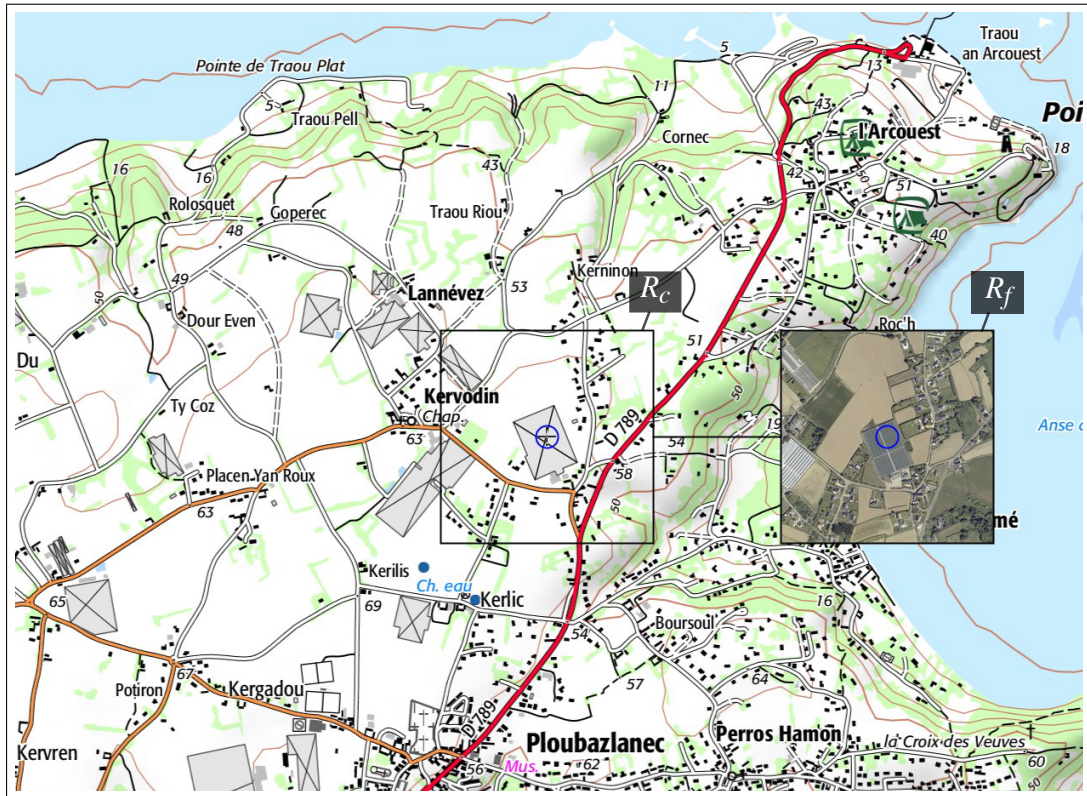


Figure 3.7: Technique OL (Offset Lens): Users see one representation in the main viewport (here the map). A fixed-size rectangle R_C , always centered on the cursor, delimits a region of interest (showing here the same facility F as in previous figures). A similar rectangle, R_f , offset to the right side of R_C , displays the same region in the orthoimagery. R_f gets smoothly offset to the left of R_C when it comes too close to the main viewport's right edge.

tionalizing comparisons performed on the two main map drawing primitives: surfaces and paths, using buildings and roads as representative instances, as detailed below. We believe the chosen tasks are reasonably close to actual comparison tasks such as, *e.g.*, a cartographer updating an outdated topographic map from recent orthoimagery (photomapping).

We propose a set of 3 tasks, performed on 2 distinct types of geographic features: *road* (line), and large man-made *facility* (surface). There are thus 6 distinct types of differences, illustrated in Figure 3.8. All are separately instantiated on each of the 20 map pairs. To limit task difficulty, any facility involved in a difference is displayed as a gray polygon, at least 1cm^2 on screen, with a black stroke and cross inside them (Figure 3.8-A). Modified facilities (Figure 3.8-C) can be either of the correct size, but offset in position (at least 1.5cm), or they can have an incorrect size (by a factor of at least $1.5\times$). Roads to inspect are those painted orange (*routes départementales*). Missing roads (Figure 3.8-E) are always connected to an orange road at one junction, and are at least 3cm long. The distance between the modified segment of a road and the original one is at least 1cm (Figure 3.8-F). All these indications were explicitly

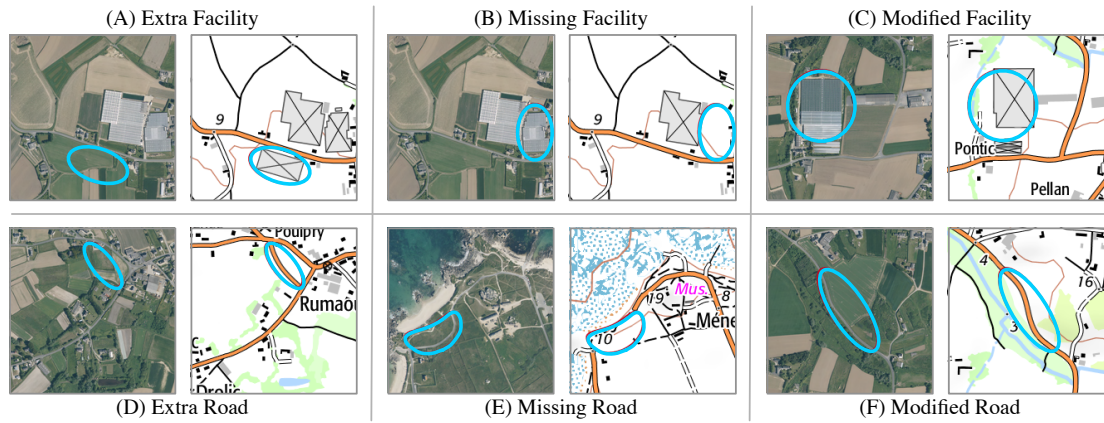


Figure 3.8: Six types of differences: (A) the map contains a facility not present in the orthoimage; (B) a facility in the orthoimage is missing on the map; (C) a facility shows a geometry mismatch between the map and the orthoimage. (D), (E) and (F) illustrate the same three differences for roads.

given to participants. All 120 differences were evaluated in a pilot study, which helped us identify and replace a subset of trials that were too difficult to perform, no matter the technique used to achieve the task.

3.2.3. HYPOTHESES

Figure 3.5 summarizes the positioning of all five techniques considered in the experiment, with respect to how much divided attention and visual interference they cause relative to one another, and whether users' scanning strategy is predominantly motor- or vision-driven. This characterization of techniques led to the following hypotheses:

- *H1*: Techniques that superimpose images (BL, OV, SW) will perform better than those that juxtapose them (JX, OL) because problems of divided attention will be more detrimental to task performance than problems of visual interference, even more so given that the latter can be mitigated via interactive parameter adjustments.
- *H2*: This difference will be stronger for tasks that require precise geometrical comparisons (DIFF = *modified*) than for the other tasks (DIFF = *extra* or *missing*).
- *H3*: Vision-driven scanning should be more efficient than motor-driven scanning, implying that OV will perform best.
- *H4*: OL will provide a good compromise for tasks that do not require precise geometry comparison, as it tries to mitigate problems of divided attention and

visual interference.

3.2.4. PARTICIPANTS AND APPARATUS

Fifteen unpaid volunteers (six females), daily computer users, age 20 to 37 year-old (average 28.5, median 28), served in the experiment. All had normal or corrected-to-normal vision and did not suffer from color blindness. We conducted the experiment on an HP workstation running Linux Fedora 19, equipped with a 3.2 GHz Intel Xeon CPU, 16GB RAM, and an NVidia Quadro 4000 driving a 27" LCD (2560x1440, 100 dpi). We used a standard HP optical mouse set to 400 dpi resolution and default system acceleration. The experiment was implemented in Java 1.7 using the ZVTM toolkit [111].

3.2.5. PROCEDURE

We followed a $[5 \times 2 \times 3]$ within-subject design with 3 primary factors: $TECH \in [BL, JX, OL, OV, SW]$, and the type of task, decomposed into two factors: the type of feature $GEOENT \in [road, facility]$, and the type of difference $DIFF \in [extra, missing, modified]$. Measures include task completion *Time* and incorrect selections, treated as *Errors*.

We grouped trials into 5 blocks, one per $TECH$. $TECH$ presentation order was counterbalanced using a Latin square (5 orders). Within each $TECH$, trials were organized into 6 sub-blocks, one per task ($GEOENT \times DIFF$). Task presentation order was counterbalanced independently for each block, using $15 \times 6 = 90$ orders randomly picked out from the $!6 = 720$ possible ones. Within each sub-block, participants performed 1 training trial followed by a series of 3 measures. Maps were grouped in batches of 4 per $TECH$, always presented in the same order. Using this counterbalancing strategy, illustrated in Figure 3.9, we ensured that every $GEOENT \times DIFF$ combination was performed on every map with every technique by 3 distinct participants. Proper counterbalancing is important: using real-world data means that we cannot fully control the content – and thus the complexity – of our maps and orthoimages. We thus have to handle this potential source of bias.

Participants could rest as long as they wanted between trials, but were asked to perform the task as fast as possible once a trial had started, while avoiding errors. To complete a trial, they had to put the mouse cursor on top of the difference (within 100 pixels from its center), and validate their answer by pressing the space bar. They were reminded about the type of difference before the start of each trial ($DIFF+GEOENT$, with the corresponding image pair from Figure 3.8) to avoid any confusion. After

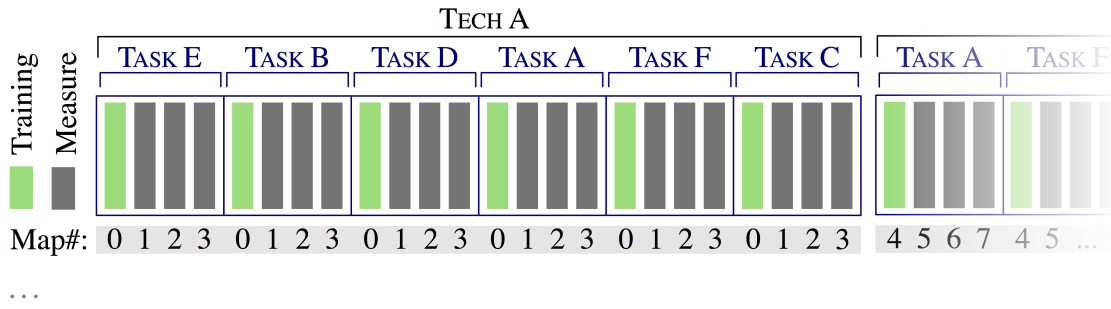


Figure 3.9: Counterbalancing example: blocking by TECH; the same 4 maps are used for all 6 tasks (1 training + 3 measures) in the same block.

each series of 4 trials, participants had to call the experimenter, who asked them to rate the difficulty of the task with each technique using a 5-point Likert scale. The experimenter then made sure participants understood that the task was about to change, and what the new task was.

A correct answer was indicated by a large green tick mark; an incorrect one by a large red cross. To avoid participants rushing through the experiment, incorrect answers were counted as errors, but did not end the trial; participants still had to identify the right feature. To avoid participants getting stuck, we set a 90-second timeout for each trial. The average duration of the experiment was 75 minutes.

3.2.6. RESULTS

A total of 1071 trials were successfully completed out of the 1350 performed, *i.e.*, participants identified the difference within the 90s time limit in 79.3% of all trials. Figure 3.10 shows the distribution of successful trials that remain for analysis. We can see that, despite our initially-balanced design, collected measures are not properly balanced among $\text{TECH} \times \text{GEOENT} \times \text{DIFF}$ conditions. While successful trials are equally distributed among the different TECH conditions, their number is much lower in the *road* \times *missing* than in all other $\text{GEOENT} \times \text{DIFF}$ conditions. Finding a missing road (Figure 3.8-E) was too difficult a task for some participants. In order to conduct analyses on datasets having approximately-equal sample sizes per condition, we filtered out all *road* \times *missing* trials and considered the following two datasets:

- D_{facility} , containing the 642 successful trials in conditions $\text{GEOENT} = \text{facility}$ and $\text{DIFF} = \{\text{extra}, \text{missing}, \text{modified}\}$;
- D_{road} , containing the 426 successful trials in conditions $\text{GEOENT} = \text{road}$ and $\text{DIFF} = \{\text{extra}, \text{modified}\}$.

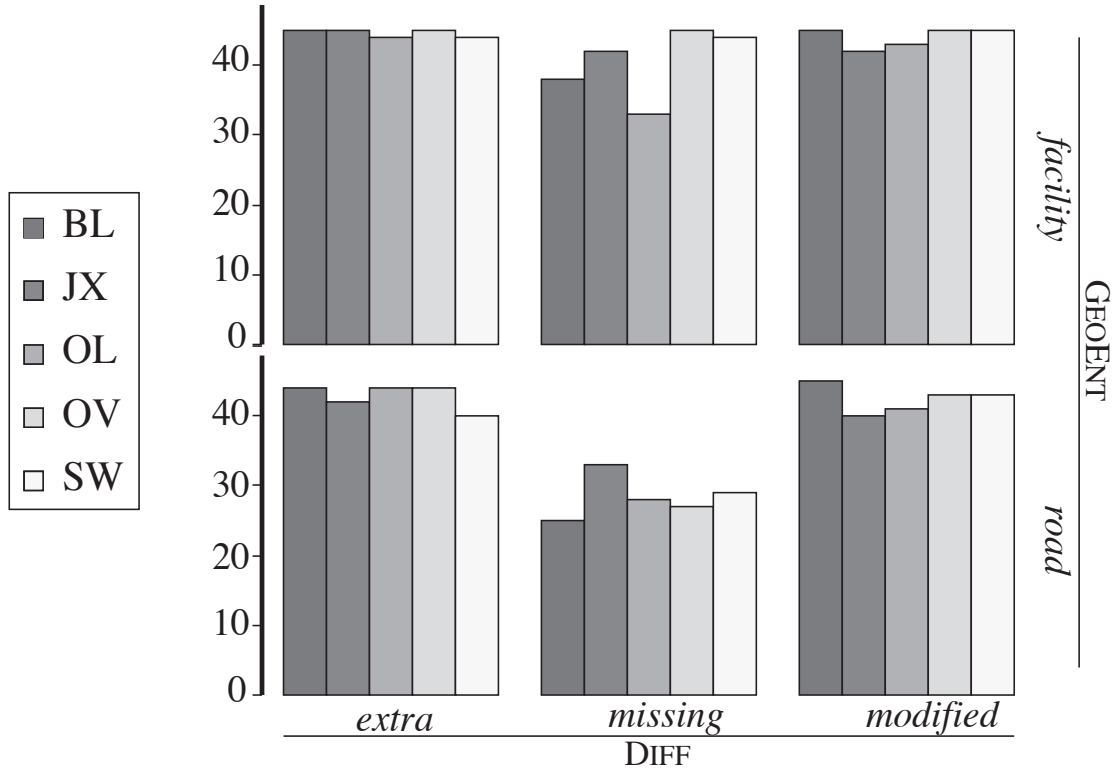


Figure 3.10: Number of successful trials per TECH \times GEOENT \times DIFF.

In 43 of the 642 trials in $D_{facility}$, participants made at least one error before finding the right difference (error rate = 6.5%). An ANOVA on TECH \times DIFF reveals that there is no significant effect on *Errors*, showing that no condition is more error-prone than others. On the contrary, both TECH and DIFF have a significant effect on *Time* ($F_{4,56} = 4$, $p = 0.005$, $\eta_G^2 = 0.08$ and $F_{2,28} = 39$, $p < 0.001$, $\eta_G^2 = 0.28$, respectively). First, finding a missing facility (avg=22.6s) takes significantly longer than finding a modified facility (avg=15s), which takes significantly longer than finding an extra facility (avg=10s). Second, OV (avg=12s) is the fastest technique, significantly different from SW (avg=16s) and OL (avg=20s), but not from JX (avg=14s) and BL (avg=14s). OL is significantly slower than all other techniques. Finally, an ANOVA reveals a significant TECH \times DIFF interaction effect on *Time*. Figure 3.11-a illustrates the results of post-hoc paired t-tests on the different pairs of conditions. For *missing* tasks, BL and OL perform much worse than the three other techniques, with no significant difference in-between those. However, for *modified* and *extra* tasks, the comparative performance between techniques changes. For *extra* tasks, BL and OL are the fastest techniques, although the difference is significant only for SW, which performs particularly poorly. For *modified* tasks, BL performs better than all other techniques. The difference is significant for all of them, except OV.

In 18 of the 426 trials in D_{road} , participants made at least one error before find-

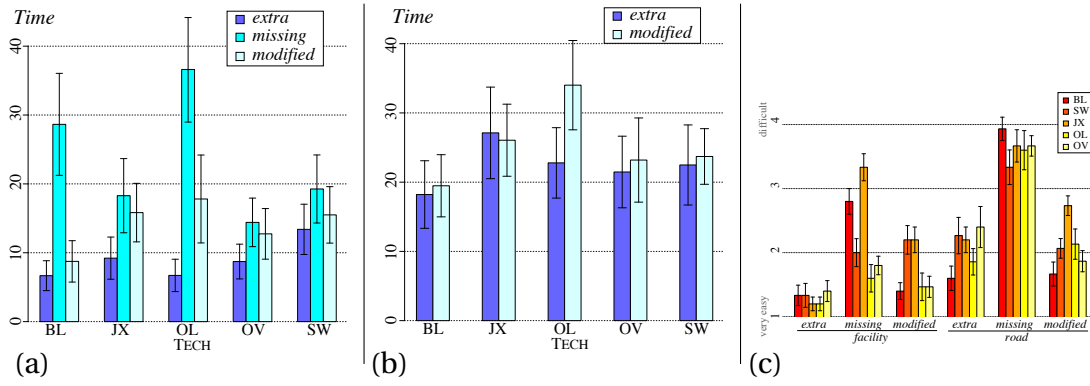


Figure 3.11: (a) *Time* per TECH \times DIFF for GEOENT = *facility* trials (error bars represent the 95% confidence intervals). (b) *Time* per TECH \times DIFF for GEOENT = *road* trials (error bars represent the 95% CIs). (c) Perceived *Difficulty* per TECH \times TASK (error bars represent the standard errors).

ing the right difference (error rate is 4.2%). As for $D_{facility}$, an ANOVA on TECH \times DIFF reveals that there is no significant effect on *Errors*. Regarding *Time*, only TECH has a significant effect ($F_{4,56} = 2.8$, $p = 0.03$, $\eta_G^2 = 0.06$). As shown in Figure 3.11-b, BL (avg=19s) is the fastest technique, the difference being significant with OL (avg=28s) and JX (avg=26s), but not with SW (avg=23s) and OV (avg=22s).

Additional insights can be gained by performing an ANOVA on TECH \times DIFF, without removing the *road* \times *missing* trials that were filtered out in the above analysis. Results indicate that TECH ($F_{4,57.43} = 5.01$, $p = 0.0015$), DIFF ($F_{2,26.65} = 71.57$, $p < 0.0001$), and TECH \times DIFF ($F_{8,108.3} = 6.31$, $p < 0.0001$) all have a significant effect on *Time*. OV is the fastest technique on average (avg=16s), followed by BL (avg=18s), JX (avg=20s), SW (21s) and OL (avg=25s). But BL is actually the best technique in the DIFF=*extra* and DIFF=*modified* conditions. However the difference with OV is significant only in the DIFF=*modified* condition.

Finally, Figure 3.11-c reports the perceived *Difficulty*, from 1 (very easy) to 5 (very difficult), of using a technique for completing each task (TASKs are the 6 GEOENT \times DIFF combinations). An ANOVA reveals effects consistent with the measured quantitative performance. TECH has a significant effect ($F_{4,56} = 4$, $p = 0.006$, $\eta_G^2 = 0.1461859$). OL, which performs the worst on average, was perceived as significantly more difficult to use than all other techniques, which do not significantly differ from each other. TASK also has a significant effect ($F_{5,70} = 69$, $p < 0.001$, $\eta_G^2 = 0.73$). Finding a missing road was rated as significantly more difficult than any other task. Finding a modified or an extra facility was significantly easier than other tasks (*extra* \times *facility* being the easiest). Using BL, all tasks were perceived as easy or very easy, except for tasks in which DIFF = *missing*. Using OV, all tasks were considered as easy or very easy, except finding a missing road, which was rated as difficult, no matter the technique.

3.3. DISCUSSION

A first general observation is that, as expected, the three types of differences (DIFF) have a strong impact on task completion time: finding *missing* features takes longer than finding *modified* ones, which takes longer than finding *extra* ones. This observation can be explained by the strategy employed to find the three types of differences. It is important to recall that orthoimagery (ORTHO) was used as the reference image, with errors introduced in the topographic map (SCAN25). In the *extra* and *modified* conditions, participants would simply look at the different features on the SCAN25 map and compare them against ORTHO. Eligible features were quite easy to identify visually on SCAN25. But in the *missing* conditions, participants had to do the opposite: look for features in the ORTHO image and check whether they existed on the SCAN25 map or not. In this case, eligible features were less easy to identify, which partially explains the strong task completion time difference between *extra* and *missing* conditions in Figure 3.11-a. Finding *modified* features took longer than finding *extra* features because it involved more elaborate geometrical comparisons. Still, participants were faster at this than at finding *missing* features, as eligible features could be identified on SCAN25 maps, as in the *extra* conditions.

Regarding our hypotheses, *H1* is only partially supported. When comparing *roads*, BL, OV and SW do perform better than JX and OL indeed, but only significantly so for BL. Also, OL performs significantly more poorly than all other techniques. When comparing *facilities*, OV and BL are the two best techniques, along with JX. SW performs poorly overall, as discussed later. Thus, superimposition (as opposed to juxtaposition) does seem to be the right strategy, when combined with other properties. But it is not a determinant factor.

H2 is also partially supported. When looking for *modified* roads or facilities, the three techniques based on superimposition (BL, OV and SW) indeed perform consistently better than those based on juxtaposition (JX and OL). However, the task-time difference is not systematically significant. As for *H1*, our results provide empirical evidence only for specific techniques (BL); not for the whole class of superimposition techniques. Thus, our results only suggest that superimposition works best when looking for subtle differences, such as the path of a road portion being slightly different or a building's position being offset. Subjective ratings are in accordance with these quantitative measures (Figure 3.11-c), reflecting the higher difficulty of performing such comparisons on juxtaposed representations as opposed to aligned ones.

OV features the best overall task completion time, significantly shorter than those

of SW and OL. *H3* is thus supported, though this should probably be attributed more to the low level of divided attention rather than to the vision-driven scanning strategy, since BL (which employs a more motor-driven strategy) performs as well – if not better – in many task conditions. This actually raises the question of whether a motor-driven scan could actually have positive side-effects. While it prevents users from performing fast visual searches, motor-driven scanning might actually help *structure* the exploration, making users less prone to performing random jumps between arbitrary locations, potentially reducing the number of revisits. Additional studies are required to confirm this.

H4 is rejected. By displaying the two representations closer together than JX does (thus limiting the amplitude of saccadic eye movements) while rendering both of them fully opaque (thus minimizing visual interference), we were expecting that OL would mitigate these two problems, but it does not.

As illustrated in Figure 3.5, techniques Translucent Overlay (OV) and Juxtapose (JX) can be seen as extreme points in the design space of composition techniques, with Blending Lens (BL), Offset Lens (OL) and Swipe (SW) trying to find compromises between those two extremes. OV is the most versatile technique, featuring good performance for all tasks. As expected, divided attention proved to be more detrimental than visual interference. While it looked promising, SW turned out to be surprisingly inefficient for most tasks. We tentatively attribute this poor performance to the tight coupling between motor actions and visual comparison, which forces users to adopt a very constrained scanning strategy if they want to avoid too many micro-swipes back and forth. OL was a DragMag-inspired [155] attempt at mitigating problems of both divided attention and visual interference, which failed for the most part, except for the identification of *extra* features. Somewhat surprisingly, BL turned out to perform quite well on both *extra* and *modified* tasks. Its bad performance in *missing* tasks is partly explained by the earlier-discussed higher difficulty of finding features in ORTHO images. Performance should improve drastically by allowing dynamic toggling of what layer is assigned to focus and context regions.

3.3.1. VALIDITY

To evaluate interactive maps, researchers in the domain of geovisualization use techniques from psychology, human computer interaction, geography and information visualization [130]. As researchers in human computer interaction, they use logging to track mouse movements and some times combine it with eye tracking data [106]. Different visualization techniques can be compared to better design dynamic maps,

for example, comparing dynamic to static maps. They also user centered design to create systems in collaboration with expert users, by incorporating user feedback from the beginning [133] or by asking for expert feedback once the system has been built [76].

User studies in HCI tend to rely on more abstract empirical tasks with more controlled conditions, yielding mostly quantitative measures (see, *e.g.*, [4, 69, 112, 116, 121, 125]) such as time to accomplish a task, or accuracy. For example, the study presented by Pietriga et al [112] compares the time needed in a search task using different multi-scale navigation techniques. Jakobsen et al [125] also compare multi-scale navigation techniques but they also take into account the display size.

HCI experiments that involve techniques for interacting with maps, are also in general quantitative, and are based on highly-abstracted experimental tasks [42, 75, 95, 112, 113, 121]. Even when seeking a higher-level of ecological validity, the base maps and overlaid features tend to be simplified, either in terms of rendering or layout, so as to minimize the influence of factors that would be otherwise difficult to control [14, 56, 69, 115, 116, 125, 163]. While this is a generally sound approach, our study warranted the use of real-world maps. Abstract representations, or even simplified maps, would have strongly altered the end-result of the different map composition strategies in terms of visual complexity, which would have significantly threatened the external validity of our experiment.

As a result, we had to address several potential threats to internal validity. First, orthoimagery and the corresponding topographic maps cannot easily be synthesized. We thus have to use actual maps, which entails that we cannot fully control the visual complexity of our base maps. As a consequence, we had to pay careful attention to our experiment design, so as to minimize this potential source of bias. In addition to the counterbalancing strategy we described earlier, we also tried to minimize differences between maps with respect to the particular geographical features considered. Some topographic maps were also customized by hand before the introduction of differences, changing the type of some roads so as to have a relatively equivalent length of eligible roads in all 20 map pairs.

Another threat to internal validity is the lack of control over participants stumbling upon the correct feature earlier rather than later. While some operationalizations address this issue [75, 112, 115], these elaborate strategies rely on synthetic data and are incompatible with the use of real maps. But with 3 replications per participant per TECH \times GEOENT \times DIFF condition, and a total of 1071 successful trials analyzed over our 15 participants, this threat should be minimal.

Our focus on external validity does not mean that our findings generalize to the comparison of arbitrary types of images. As mentioned earlier, these techniques are used in a wide range of application areas. This includes astronomy and medical imaging, *i.e.*, domains in which the two images are going to be visually much more similar than topographic maps and orthoimages are. Even in geovisualization, results may be different when comparing visually-similar maps. Indeed, the nature and impact of visual interference caused by the superimposition of such images is likely to be very different. Results might also differ in the case of topographic map & orthoimage comparison performed in highly-urbanized geographical regions, as we used maps of predominantly rural areas. Evaluating the influence of different map types would have required introducing additional factors such as object density. Doing so would have made the experiment's complexity unmanageable, and was left as future work.

3.4. SUMMARY

We characterized and evaluated five techniques for performing interactive map comparison, using real-world maps. The study's results suggest the following guidelines to UI designers: (1) Translucent Overlay (OV) is the best technique overall, which makes it a good choice when only one comparison technique should be provided to users. (2) Even if its overall performance is not as good as that of OV, Blending Lens (BL) performs as well as or better than OV when the task consists of identifying extra or modified entities. We tentatively attribute this to BL's mostly motor-driven scanning strategy, which helps structure inspection of candidate features in the upper layer. The combination of OV and BL should be favored by UI designers when possible, as the two are quite complementary. (3) Swipe (SW) and Offset Lens (OL) perform poorly. This is an interesting finding given that SW is a technique commonly encountered in Web mapping applications. Its use should probably be reconsidered.

4

MAPMOSAIC: DYNAMIC SPATIAL MULTIPLEXING

Main portions of this chapter were previously published in the journal IJGIS [92].

The evaluation presented in the previous chapter suggests that some techniques that superimpose geographical layers are more efficient than techniques that juxtapose them when comparing maps. *Translucent Overlay*, an interactive technique that superimposes one layer on top of the other with adjustable translucency was the most efficient overall. For some tasks, superimposing only in a subregion of the map (using the *Blending Lens* technique), was as effective or more than *Translucent Overlay*. Because a lens reveals another layer in a local area and can be freely moved over the map, users can structure their search, and make the lens follow their exploration path.

Drawing from these results, we aim at designing a novel *spatial multiplexing* interactive technique that allows users to perform spatial multiplexing in arbitrarily-shaped areas. The evaluated spatial multiplexing techniques in the previous chapter superimpose layers in areas of predefined shape: *translucent overlay* changes the transparency in the whole viewport and the *blending lens* technique reveals one layer in a fixed-size region and shape. They consider the layers as flat images and do not take into account the information contained in them. The existing GIS user interface front-ends, from both research projects [35, 55] and industrial-strength products such as ArcGIS (<http://www.esri.com/software/arcgis>) and QGIS (<http://www.qgis.org>) do not provide any more flexibility; the *creation and editing* of elaborate layer composites are cumbersome, involving many indirect manipulations.



Figure 4.1: A proof-of-concept implementation of the *MapMosaic* dynamic compositing model and interaction techniques. (a) Toolbar to navigate the map, and to create & select areas. (b) Map viewer. (c) Access to individual layers. (d) Compositing area inspector. (e) Query builder for compositing region filters.

In this chapter, we introduce a novel approach to spatial multiplexing. *MapMosaic* takes advantage of map’s semantics to enable dynamic visual compositing, featuring novel techniques to interactively create and manipulate local composites of multiple vector and raster map layers. We first report on interviews conducted with GIS experts. These interviews helped identify interactive layer compositing as a key aspect of users’ interaction with map content. We then introduce our dynamic map compositing model, and describe a prototype implementation (see Figure 4.1) for which we report graphics rendering performance figures. The two sections that follow provide first steps toward validating our model: 1) an evaluation against the “cognitive dimensions” framework [51], as well as a detailed analytical comparison of *MapMosaic* against QGIS, a widely-used desktop GIS, and MAPublisher, a professional cartographic editor (<http://www.avenza.com/mapublisher>); 2) a report on feedback gathered from workshops with GIS expert users.

4.1. BACKGROUND AND MOTIVATION

As mentioned before, we consider that geographical layers consist of either objects or fields. They can be superimposed in a single view, or composed using other strategies, making it possible for users to correlate entities from different sources, interpret them in a broader context, compare them, gain novel insight from this visual combination [41], and communicate them.

As mentioned in Chapter 2, beyond the compositing techniques commonly encountered in interfaces for comparing two maps or images [93], recent work in cartographic research has produced methods to overlay ortho-rectified imagery and to-

pographic maps or other vector data [66, 68, 120]. However, these techniques are designed to produce static maps and do not support effectively interactive exploration tasks. GIS and cartography tools such as MAPublisher offer more powerful features. They combine general-purpose layering and compositing techniques with geospatial data query and attribute-based filtering capabilities. However, these tools have been designed with the production of high-quality, static graphics output in mind. The cumbersomeness of the associated interaction model and its lack of support for dynamic compositing significantly impedes the exploratory processes users are often involved in. While some tools such as ArcGIS do feature support for more interactive forms of compositing, the capabilities are limited to coarse, data-agnostic techniques such as toggling and swiping between two layers [93], that only enable very basic types of compositing.

Exploratory tasks typically involve numerous navigation actions between layers: adding and discarding some of them, filtering objects within those layers, and making incremental adjustments to the visual representation, including adjustments to the graphical compositing settings, its geographical scope, the objects and fields to be considered based, *e.g.*, on their type or on individual attribute values. The approach taken by the tools mentioned above makes such iterative processes possible but quite cumbersome. As discussed in Section 4.5, users who want to make even small adjustments to their composite representation (adjust the scope of masks, filtering criteria, or graphical compositing rules) will often have to redo their selection and recreate the corresponding compositing masks from scratch, going through many interface dialog boxes. This interrupts users' train-of-thought and imposes much extraneous cognitive load on them [146] as they have to plan for these sequences of actions.

MapMosaic takes a different approach, proposing a new, *dynamic* model for layer compositing. *MapMosaic* exposes compositing operations as first-class objects, enabling users to manipulate them in a more direct manner [141]. Users are able to edit the properties of these objects at any time, and get immediate visual feedback that tightens the perception-action loop and facilitates the exploratory process from a cognitive perspective.

4.2. INTERVIEWS WITH GIS EXPERTS

Before designing *MapMosaic*, we conducted a series of interviews with a panel of GIS experts whose tasks are representative of the wide range of activities that require users to interact with multi-layer maps. Our pool of interviewees comprised four GI

Science researchers with diverse interests and expertise, and ten GIS operators from IGN, the French National Geographic Institute. These interviewees work in different departments, performing varied tasks that involve the combination of different datasets. We focus here on aspects of tasks and associated user interactions that involve some form of layer compositing.

4.2.1. GIS OPERATORS

4

In order to get a more realistic picture of the tasks performed by GIS operators, we interviewed them in their work environment, seated together in front of their workstation running the software applications they use daily. The following sections report on the tasks typically performed in the four departments we visited.

DIGITAL ELEVATION MODELS

The work of operators tasked with the production of Digital Elevation Models (DEM) basically consists of fixing a field (resolution: 5 meters) that results from the automatic processing of satellite data. This treatment also outputs a layer that contains masks delineating regions in which the algorithm has required several iterations to derive the altitude from the input data. Multiple iterations are indicative of uncertainties that should be checked manually. To perform such verifications, the operators compare the model to multiple layers: orthoimagery, masks derived from specific object types retrieved from a database (*e.g.*, water bodies, whose elevation is known and should be constant), data from a third-party, lower-resolution DEM. In the software application used for this task, the different layers are stacked on top of each other and are presented as a textual list with checkboxes that control their visibility. Beyond this basic, monolithic approach to layer management (where the content of a layer is either displayed in full or not at all), the software application also features a tool that enables operators to reveal another layer in a region of arbitrary shape by delineating that region and selecting which layer to reveal inside it.

AERONAUTICAL CHARTS

Aeronautical charts are very complex maps used when preparing flights and navigating aircrafts. They are made of many layers. The operators' task consists of updating both the civilian and military versions, including, among other things, lists of obstacles that have been identified by third parties. Based on a structured text-based representation of the data, operators must position these obstacles on both versions of the map, and input the version-dependent attributes that describe them. Their

user interface front-end features more than ten tabs, one per layer. As the tool only supports monolithic compositing, operators spend a significant amount of time toggling between tabs: two tabs show all layers superimposed, one for each version, while other tabs show a series of derived charts that provide a lighter representation. These latter charts focus on certain types of obstacles such as pylons or windmills, facilitating specific user tasks by simplifying the otherwise-very-complex visualization.

LAND COVER AND USE

The institute is also in charge of producing maps that describe the land in terms of *cover* and *use*. Using photo interpretation, operators have to make a partition of the land into contiguous polygons with attributes that describe the type of cover (*e.g.*, grass, trees, asphalt) and its destination use (*e.g.*, farmland, woods, urban area) at a scale of 1:2000. To cut or fuse polygons, they overlay a semi-transparent vector layer holding the polygons on top of orthoimagery. To help operators structure their work, the software application overlays a grid that they can follow sequentially. Depending on the regions considered and on their personal preferences, operators frequently switch between two types of orthoimagery layers, that differ in the data channels they encode (Red-Green-Blue or Red-Green-Infrared). Again, because only monolithic compositing is supported in this application, all layers are made available in separate tabs, requiring operators to toggle between them frequently.

DATABASE CURATION

Operators from this department are in charge of continuously updating the institute's main database, that contains the most detailed representation of all entities. Most of the products and services offered by the institute are derived from this database. The operators' task consists of identifying places that have to be updated, and collecting the necessary information to add, modify or remove objects. They perform one part of their job in the field, and the other part in the office. In the office, they access many sources of information, including numerous layers such as ortho-rectified aerial imagery (very high-resolution, updated every 5 years), satellite imagery (lower resolution, updated more frequently), and different thematic layers that hold specific types of entities (*e.g.*, roads, buildings and addresses, forests), as well a layer that highlights the areas that need to be checked. We observed the same pattern as in the above scenarios: operators frequently have to navigate between layers using tabs, thus changing the content of the entire viewport at once and often losing the contextual information that was available in the previous tab, imposing much back-

and-forth in terms of user interface window management.

4.2.2. GI SCIENCE RESEARCHERS

Among the four GI Science researchers, three investigate *cartographic processes* (e.g., generalization and symbol specification). The last one works in a team that studies *risk management* from a socio-environmental perspective.

CARTOGRAPHIC PROCESSES

The first three researchers focus on formalizing knowledge about cartographic processes (e.g., generalization, map styling and symbol specification). They investigate methods and tools to improve the design and automatic generation of maps. Because of the exploratory and bleeding-edge nature of their work, they rely mostly on programming tools to implement their data processing, geometry transformation and rendering methods. But they also use general-purpose GIS front-ends to debug and evaluate the results of these treatments. For instance, when developing a generalization algorithm [148], one researcher expressed the need to better understand and compare the different steps of his algorithm. He wanted to compare not only the input map with the generalized output map, but also intermediate steps for debugging purposes. Such comparisons require computing the intermediate output from the different steps, and rendering them on separate, superimposed layers using distinct graphical styling rules to make it easy for users to visually differentiate them. Furthermore, this complex compositing should be bounded to the region of interest only, leaving the surrounding area untouched so as to preserve the geographical context.

Another researcher, who is more interested in symbol specification and map styling, reported spending a lot of time and effort juxtaposing the multiple alternative renderings that she wanted to compare.

Here again, we observe that the associated layer compositing and comparison tasks are often performed manually, as current GIS user interface front-ends do not offer the flexibility these users need in terms of *region-based compositing* and *concurrent visualization of multiple layers*.

RISK MANAGEMENT

This fourth researcher works in a team that focuses specifically on the assessment of environmental risk, the modeling of its potential impact, and the creation of tools for supporting public policy decision makers and increasing public awareness. Activities

related to this type of risk management are split in three main phases: *before* the disaster, *during* the disaster, and *after* the disaster. Each phase involves different types of tasks.

Before the disaster, efforts focus on activities related to risk prevention, such as the creation of evacuation plans for coastal regions that face risks of tsunamis. The corresponding cartographic processes involve collecting knowledge from the field, which then gets correlated with different layers, both raster (orthoimagery) and vector (road networks, population grids, simulations).

After the disaster, situation awareness is of prime importance. Authorities need to know about the current status of the different geographical areas, road and communication networks and other critical infrastructure. They rely partly on *rapid mapping* [109] for this. These quickly-produced maps serve as a basis to coordinate emergency response actions, such as evacuating a school or routing injured people to hospitals. Rapid mapping essentially relies on photo interpretation performed by experts or crowdsourcers [109], who compare incoming up-to-date satellite or aerial imagery with photos taken before the disaster, in order to identify and mark damaged buildings and roads. As external elements, such as weather conditions, can impact the quality and coverage of post-catastrophe images (occluding clouds, snow-covered area), rapid mappers may have to consider multiple images of the same region (*e.g.*, at different times or seasons) to get a good visualization of the area they are working on. These mapping tasks not only require integrating information from *heterogeneous map layers* but also being able to *quickly identify specific types of entities* such as critical infrastructure.

4.2.3. DESIGN REQUIREMENTS

The above interviews show that experts work with a number of layers ranging from three to more than ten, both raster and vector. These experts frequently have to navigate between views that consist of different combinations of those layers. Which target layer(s) they aim at combining and visualizing not only depends on their task and work habits, but also in many cases on the nature of the region of interest. How these layers can be effectively combined then depends essentially on their respective content and level of visual complexity. For instance, a sparse, symbolic layer can be composited with a terrain base map using a simple overlay technique. But compositing a field with ortho-imagery will typically require the use of a more elaborate alpha blending operator to achieve semi-transparency [117]; and combining two dense and visually complex layers such as, *e.g.*, a topographic map and ortho-imagery, will re-

quire interactive navigation [93], as static alpha-blending would inevitably cause too much visual interference between the two layers.

As most software applications only support monolithic forms of layer management and compositing, users can only resort to a very limited set of strategies, which range from basic, individual layer visibility toggling to tab switching, each tab holding a specific layer compositing configuration. In all cases, users have to perform numerous *back-and-forth switches*. Our first design requirement is thus to *limit such switches*.

4

The second (related) issue is the lack of support for flexible, *local compositing*, i.e., layer compositing operations bounded to specific regions of the display. Monolithic compositing forces users to perform numerous switches between layer configurations. But it also prevents them from retaining contextual information visible on screen, knowing that context is often key to interpretation and navigation. Drawing a parallel with multi-scale navigation [29], users can be faced with a variant of desert fog when switching between layers or arrangements thereof. Desert fog is a term that was coined by [77] to refer to the “*condition wherein a view of an information world contains no information on which to base navigational decisions*”, or in other words, “*where the immediate environment is totally devoid of navigational cues*”, leaving users disoriented. This can happen in multi-scale worlds when, for instance, zooming in or out too much, as all relevant landmarks and meaningful visual cues disappear in the process. Transposed to the case of navigation between multiple layers, desert fog may occur when navigation is mainly driven by visual cues that belong to one specific layer. In an analysis task that involves multiple geographical areas, users might have to change the visibility or compositing settings of that layer to achieve the analysis task locally. Navigational cues may thus be invisible or difficult to see, in turn requiring more actions to resume navigation once the task has been completed in one location. Switching abruptly between two layers or layer composites can also cause disorientation, as some entities might be difficult to relate visually, despite the layers being spatially registered. Roads provide a simple illustration of this phenomenon: while they are easy to identify on topographic maps, they can be challenging to trace on satellite imagery depending on the terrain, vegetation, and lighting conditions. Our second design requirement – *providing support for local compositing* – is aimed at enabling users to keep landmarks and other navigational cues from the relevant layer visible as contextual information, while at the same time revealing layers relevant to the task-at-hand in the current region of interest. Users should be able to perform such local compositing operations either interactively, as part of the exploration process, or automatically, based on the value of individual ob-

ject attributes in the considered geographical area, which would provide cues about objects of potential interest, thus guiding users in their exploration.

We derive two additional, secondary requirements from our interviews with experts. 1) Some maps, such as the aeronautical charts mentioned previously, get quickly cluttered with many symbols. Users need to filter out part of the information to retain only what is meaningful to the task at hand. User interface front-ends should make the creation of simplified layers easy, by allowing users to *filter out objects based on their type and attributes*. 2) Updating a map often requires conducting a systematic exploration with either an entity-driven strategy (focusing on specific thematic layers) or a wall-to-wall strategy (scanning the full area at a given scale). Front-ends should support both strategies, allowing users to *keep track of their exploration* by automatically marking the regions or entities already visited.

The following section introduces the fundamental concepts of *MapMosaic*, explaining how our dynamic compositing model addresses those requirements, and detailing the core user interface design principles that we followed in order to provide GIS users with an interaction model that facilitates geovisualization-based exploratory processes.

4.3. MAPMOSAIC: CONCEPTS AND FEATURES

The *MapMosaic* model is designed to enable advanced graphical compositing operations, similar to those of cartographic tools, but more flexible thanks to 1) support for *dynamic* compositing, and 2) a very different user interaction model that provides immediate visual feedback and significantly simplifies the configuration and adjustment of layer compositing parameters, thus better supporting tasks such as those presented in the previous section [76].

MM's foundational concept is that of *compositing regions*. Compositing regions enable users to combine layers either globally or locally. They are the cornerstone of our approach to what we call *on-demand spatial multiplexing*, *i.e.*, the possibility to interactively combine multiple map layers into one dynamic map.¹ Compositing regions act as masks that can be moved using drag and drop, and whose parameters can be adjusted dynamically, according to the direct manipulation paradigm [141]: *MapMosaic* exposes those masks as first-class, interactive objects [81] that users can create and iteratively modify without resorting to complex tools or dialog boxes. Pa-

¹Drawing an analogy with the field of telecommunications, where the aim of multiplexing is to share a communication channel (a limited resource) by combining multiple signals into one: our signals are the individual map layers, and our main limited resource is screen real-estate.

rameters of those objects include typical image compositing settings such as alpha blending and boundary styling, as well as *filters* that can be used to further customize compositing within the region using simple queries. In this section, we describe *MapMosaic*'s interaction model and illustrate it on an example scenario in which an expert in risk management works on a tsunami evacuation plan for an island in the Caribbean. This scenario is directly derived from a concrete use-case that the crisis management expert's team worked on recently. The expert reported that building such an evacuation plan is a tedious process for which they relied mostly on QGIS and field observations.

4

4.3.1. LAYERS

As shown in Figure 4.1, users can load both raster and vector layers in the system. Each raster layer is made available directly in the Map viewer (Figure 4.1-b). As in most GIS user interface front-ends, users can navigate each map using pan-and-zoom. The COMPOSITE tab holds all layers. The opacity of each layer can be changed individually [41]. Users can also access a rendering of each layer in isolation by clicking on the corresponding tab (Figure 4.1-c). *MapMosaic* provides support for this classic layer-compositing interface because dynamic local compositing should not be seen as a replacement for monolithic compositing, but rather as a complement to it.

Typical raster layers include orthoimagery, pre-rendered maps, as well as raw fields such as digital elevation models or temperature maps, which can be rendered by mapping the field's range of values to a color gradient (the mapping's interpolation function and the color gradient's endpoints can be parameterized). Vector layers can also be rendered, but they serve other purposes as well. One of them is to partition the map into areas based on the objects they contain: buildings, roads, water bodies, *etc.* Because this partitioning is geographically meaningful, it can play a role in the identification, or more specifically in the *delineation*, of regions of the visual representation, to which specific compositing settings can then be applied. *MapMosaic* thus enables users to directly derive compositing regions from objects stored in vector layers.

Objects in vector layers can be selected by clicking on them with the default selection tool. As objects may overlap, *MapMosaic* highlights which one will be selected (changing its stroke and using a translucent fill color) whenever the cursor moves. The strategy consists in systematically picking the smallest object containing the cursor, so as to remove any ambiguity about which object is the current target. A click is

required to actually select the object. Additional clicks performed at the same location will pick the direct parent of the candidate object in the containment hierarchy.

In our tsunami evacuation map-making scenario (Figure 4.1), the expert has loaded five raster layers: ORTHO (aerial imagery), POPULATION (a population grid based on census data), ALTITUDE (a digital elevation model), SCAN (a topographic map) and a SIMULATION of forecast coastal flooding; as well as four vector layers that describe different types of objects: BUILDINGS, CRITICAL INFRASTRUCTURE, ROADS, and EVACUATION ZONES.

4.3.2. COMPOSITING REGIONS

Once a set of layers has been loaded, users can derive a compositing region from the currently-selected area in the COMPOSITE tab. Compositing regions can be seen as geo-referenced clipping masks that composite multiple layers. A key element of our model is the reification [9] of compositing regions, meaning that compositing regions are considered as first-class objects. Those objects have many attributes that can be edited at any moment using an inspector window similar to those found in integrated development environments or vector graphics applications, as detailed below.

Compositing regions always delineate a particular geographical region, but are not necessarily derived from objects stored in vector layers. Following the principle of polymorphism in user interface design, which “*permits commands to be applied to objects of different types*” [9], compositing regions can also be derived from any user-drawn selection made with one of the tools featured by *MapMosaic*: lasso, circle, rectangle and magic wand (Figure 4.1-a), inspired by similar selection tools found in image editors. In the absence of any selection, the entire map acts as an implicit compositing region, enabling the specification of compositing operations that affect the whole workspace, which can be useful when combined with filters, as we will see later.

Once a selection has been turned into a compositing region in the inspector (Figure 4.1-d), users can select the target layer from the list of all layers, or they can choose to composite with a custom solid color. Additional parameters include: opacity settings, outline stroke to artificially delineate a region, and control over the smoothness of the transition with the surrounding region (see Figure 4.3), so as to achieve an effect similar to that enabled by blending lenses [114]. In our example scenario, the user selects one of the evacuation zones (stored in the EVACUATION ZONES vector layer) and turns it into a compositing region (Figure 4.2-1), choosing SIMULA-

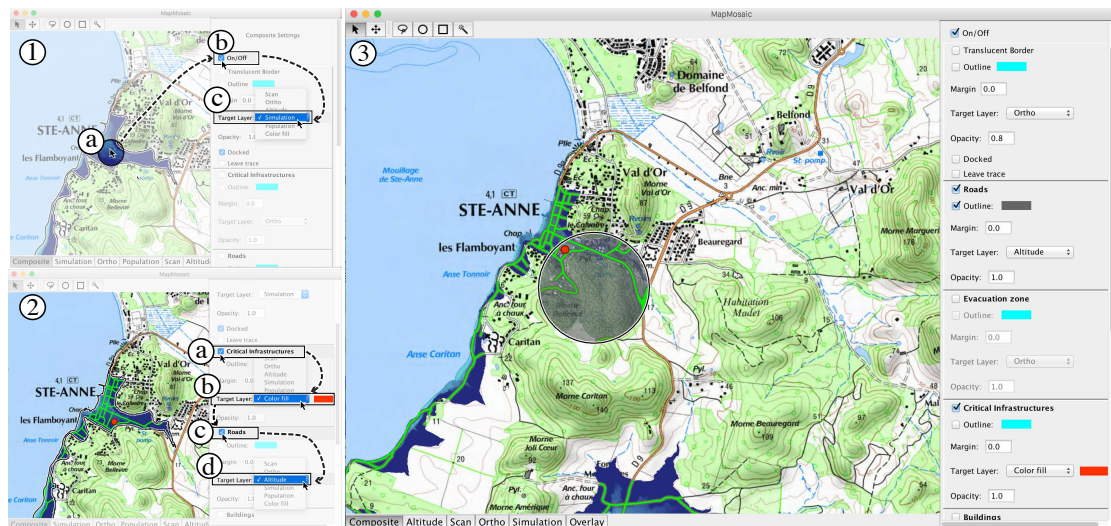


Figure 4.2: (1) Creating a compositing region from an existing vector object: (a) the EVACUATION ZONE is selected by clicking on it, (b) the selection is turned into a compositing region, and (c) SIMULATION gets chosen as the target layer. (2) Activating filters in the region: (a) the CRITICAL INFRASTRUCTURE filter is activated, (b) red is chosen as a fill color for the target layer, (c) the ROADS filter is activated, and (d) the ALTITUDE layer gets selected for compositing. (3) User-created free compositing region: showing ORTHO imagery to find a refuge zone, rendering ROADS by compositing them with the ALTITUDE layer, and showing CRITICAL INFRASTRUCTURES as red spots.

TION as the target layer in order to identify the zones that would be flooded in case of a tsunami.

Other operations borrowed from graphics editors include dilation and erosion, which respectively expand and contract a region while preserving its general shape. This can be particularly useful when selecting objects such as, *e.g.*, roads. A road is typically represented as a simple polyline in vector layers, which only becomes a renderable surface through a symbolization process. In *MapMosaic*, such objects can be turned into compositing regions by clicking the corresponding polyline, and dilating it to encompass not only the road itself but also the roadside. Users no longer have to manually delineate the region around the polyline, an interaction which is often tedious to perform. If only raster layers are available, the magic wand can be used as a tool to perform a selection of all contiguous pixels of the same color, which can then also be dilated.

The reification of compositing regions allows multiple regions to coexist and remain fully manipulable throughout their lifecycle. Any of their attributes can be dynamically modified using the inspector (Figure 4.1-d). The changes are reflected on-the-fly in the map view, as parameters get adjusted [142], providing users with immediate visual feedback that tightens the perception-action loop.

When moved according to the principle of direct manipulation, using drag-and-drop, compositing regions will behave in one of two ways:

- Regions can be *free*, in which case their geographical scope gets updated in real-time as users drag them over the map, behaving as magic lenses [10]. Free regions stay where users drop them.
- On the contrary, *docked* regions follow the cursor when dragged, but revert to their original position when dropped. Their geographical scope is not updated while dragging, as the rationale behind this behavior is to provide a quick and easy means to temporarily move a compositing region away, so as to compare the same geographical area in two or more layers using simple juxtaposition, without having to modify the settings of the layer composite.

Compositing regions derived from vector objects are docked by default, as they are associated with a meaningful geographical region, while compositing regions derived from hand-drawn selections are left free, as they are more likely to be used as scanning tools.

The above two types of compositing regions can also provide support to users who need to adopt a systematic scanning strategy as part of their task. *Wall-to-wall scanning* is best supported by free compositing regions. These regions can be set to leave a translucent trace wherever they have been on the map, thus helping users keep track of their exploration path. *Entity-driven scanning* is also supported, as users can apply a filter (see Section 4.3.3) on entities of interest to highlight them, and to derive docked regions from them. Users can then inspect these regions one after the other, and mark the visited ones by, e.g., compositing them with a translucent solid color.

4.3.3. FILTERS

Users can declare specific styling and compositing rules, that will apply to all objects that fall inside a given compositing region, based on their type and attributes. Simple rules that apply to all objects of a given type are created using the corresponding sub-panel in the region's inspector (see, e.g., Critical infrastructure or Roads in Figure 4.2-2). More complex rules that involve elaborate selection criteria (for instance, "all buildings at least 10 meters high", Figure 4.4-b) are created using the query builder (Figure 4.1-e). This UI component contains one tab per type of object, each tab featuring a widget for each attribute associated with that type: spinners for numerical ranges, checkbox lists for categorical attributes. An additional text field enables skilled users to input compound queries using numerical and boolean operators. The content of this user interface component always reflects what is stored in the layers currently

loaded in the application. For instance, in Figure 4.1-e, all widgets (attribute names, possible values for categorical variables, *etc.*) have been extracted automatically from the shapefiles that correspond to the vector layers.

Following the principles of reification, polymorphism and reuse further [9], filter settings can be copied and pasted from one compositing region to another using a contextual menu, thereby making it easy to create a new region with a different scope but featuring the same styling and compositing rules.

4

Coming back to our tsunami scenario, the user now wants to find the buildings that are part of the critical infrastructure to be evacuated. To do so, he activates the CRITICAL INFRASTRUCTURE filter and selects a target layer that consists of a simple red solid color fill to emphasize those buildings (Figure 4.2-2-(a,b)). This reveals one such building on the map (a school). The next step is to find a suitable road for evacuation. He applies a filter on ROADS in the evacuation region, selecting ALTITUDE as the compositing target layer to avoid choosing a road that is too steep (Figure 4.2-2-(c,d)). In this area, all roads are flat and at a low altitude (*i.e.*, they are all rendered with shades of green in the revealed ALTITUDE raster layer), so he can choose any of them to evacuate the school. The user then wants to explore the vicinity of the candidate evacuation roads, in order to identify an area that can serve as a refuge zone. Satellite imagery would help him get oriented and recognize areas that he explored during his field survey. He draws a circular selection, copies and pastes the filter settings from the evacuation zone to keep the roads and the critical infrastructure visible, but chooses ORTHO as the target layer.

The user can then scan the area by dragging this compositing region. Once a suitable refuge zone has been found, he uses the lasso tool to delineate it on ORTHO, and derives an additional compositing region (Figure 4.3-a) from this selection. Using a second lasso selection, he selects the part of the road that links this zone to the school, restricting compositing to ROADS using a white solid color (Figure 4.3-b). Finally, he creates one last compositing region using the POPULATION grid as a target layer (Figure 4.3-c) to check that the refuge zone is large-enough to accommodate all people in the area.

4.4. IMPLEMENTATION AND PERFORMANCE EVALUATION

Roth observed that when users interact with a map view, the display should be refreshed with a minimum frequency of 10Hz “*for the user to feel as though the system is responding immediately*” and 1Hz “*to avoid interrupting [his] thinking process*” [129].

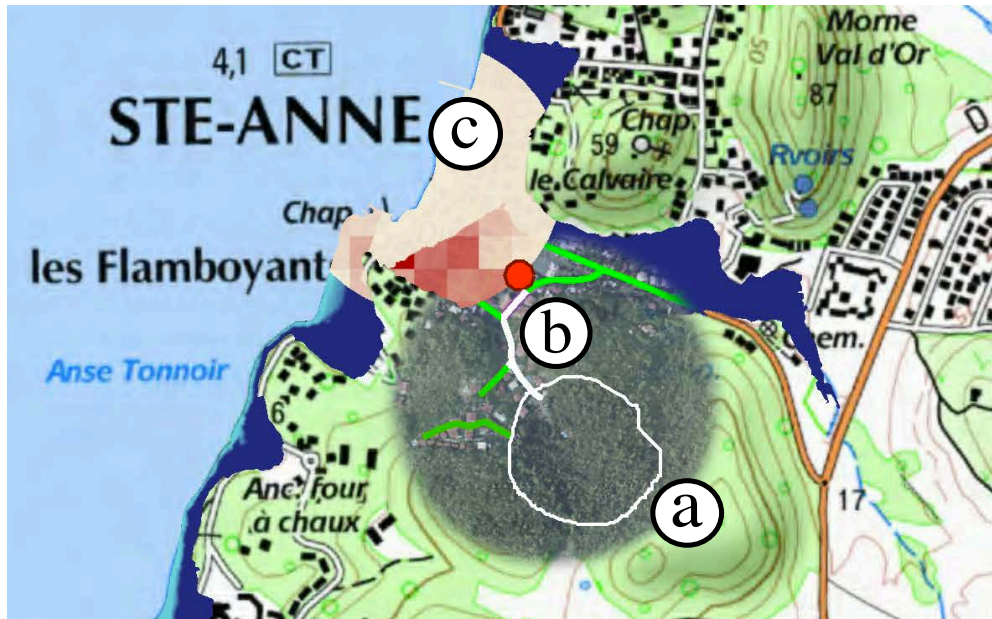


Figure 4.3: (a) Refuge zone found by examining aerial imagery. (b) Road connecting the critical infrastructure (symbolized by the red spot) to the refuge zone. (c) Compositing region showing population grid in the evacuation zone.

This is especially true for interactive exploratory tasks, that will benefit from high frame rates. Multi-layer maps are complex graphical objects that are resource intensive in terms of graphics rendering. *MapMosaic*'s compositing techniques add even more complexity to the rendering pipeline. We thus implemented our prototype in Java and OpenGL using JOGL (Java Bindings for the OpenGL API) to ensure interactive frame rates: Java's event dispatching thread handles input events, feeding a FIFO (first-in/first-out) queue processed by the JOGL rendering loop at each iteration.

Our prototype implementation is not a full-fledged GIS front-end, but rather a proof-of-concept implementation of our dynamic compositing model, and demonstrator of novel techniques that could be integrated in existing GIS software. We thus only support a subset of input data formats, including Esri shapefiles for vector data and GeoTIFF for raster images. Vector data are imported using GeoTools (<http://www.geotools.org>) and converted to OpenGL-renderable shapes using the *poly2tri* algorithm [85].

Compositing regions are drawn using shader programs written in GLSL (the OpenGL Shading Language). Different shaders are used depending on the shape's content (texture or solid color) and border style (featuring smooth blending or not). Data such as model-view and orthographic projection matrices, textures, and colors, are passed to the shaders using global variables.

The rendering of compositing regions that feature filters can be resource-

intensive, as it involves many operations on a potentially large number of shapes, including the computation of intersections. We adopt the following strategy to ensure interactive frame rates when moving those regions. We render the compositing region's main target layer into a frame buffer object (FBO) whose dimensions match those of the map view. For each filter activated, we then render the corresponding layer clipped to the shape of the vector objects that match this filter, along with the associated styling instructions. The resulting texture is then drawn on screen, clipped to the shape of the compositing region. Thus, the texture held in the FBO only needs to be recomputed when users pan and zoom the map view, or when one of the filters is modified.

Additional frame buffer objects are necessary when rendering translucent traces that help users keep track of their exploration path. More frame buffer objects are also necessary when rendering smooth blending effects between a compositing region and the surrounding area, as this requires applying multiple gaussian blur effects to the region's silhouette and using the resulting texture as alpha channel data for the fragment shader that draws the compositing region on screen.

In order to better evaluate the performance of our prototype implementation of *MapMosaic's* dynamic compositing model, we ran five rendering tests on a MacBook Pro Retina equipped with an NVIDIA GeForce GT 750M 2048 MB graphics card. The dimensions of the OpenGL panel were set to 2260×1330 pixels on a 2560×1600 Apple Thunderbolt display (the remaining space accommodating Java Swing widgets such as toolbars, menus and inspectors). Tables 4.1 and 4.2 summarize our results, when compositing all polygons and all lines of a scene, respectively. These results show how the number of entities impacts the time required to create a compositing region and, to a lesser extent, to edit it. They also show that our approach, which as described above consists of pre-rendering in textures, ensures very high frame rates no matter the scene's characteristics. Performance could be further improved by grouping objects in the same Vertex Buffer Object (VBO), instead of using a different VBO for each object.

4.5. COMPARISON WITH MAPPUBLISHER AND QGIS

As mentioned earlier, some industrial-strength GIS front-ends also feature advanced graphics editing and compositing capabilities. So does MAPublisher™, which works as a plugin for Adobe Illustrator™, a powerful, general-purpose vector graphics editor. MAPublisher enables users to import and filter geospatial data directly in the Illustrator workspace, thereby making it possible to use all of the underlying graph-

Table 4.1: *MapMosaic* performance with a compositing region (*CR*) whose scope is all *polygons* in a 2260×1330 pixels scene.

	Scene 1	Scene 2	Scene 3
SCENE CHARACTERISTICS			
Number of entities	28	1153	3714
Number of points			
Mean (Standard deviation)	597.8 (2027.0)	8.6 (6.8)	6.9 (4.6)
Min / Max	11 / 9713	5 / 89	5 / 77
Pixel area			
Mean (Standard deviation)	9526 (32265.8)	16.8 (30.2)	3.6 (4.0)
Min / Max	1.8 / 180567.3	0.32 / 369.81	0.2 / 72.3
PERFORMANCE			
Creating <i>CR</i> (in ms)	16	230	915
Editing <i>CR</i> (in ms)	2	23	73
Moving <i>CR</i> (in fps)	170 ^(*)	170 ^(*)	170 ^(*)

^(*) Constant, depends on the GPU's characteristics.

Table 4.2: *MapMosaic* performance with a compositing region (*CR*) whose scope is all *lines* in a 2260×1330 pixels scene.

	Scene 1	Scene 2
SCENE CHARACTERISTICS		
Number of entities	506	1980
Number of points		
Mean (Standard deviation)	7.4 (7.2)	10.7 (11.6)
Min / Max	2 / 52	2 / 154
Pixel length		
Mean (Standard deviation)	16.2 (23.9)	20.0 (22.6)
Min / Max	0.45 / 249.16	0.42 / 295.18
PERFORMANCE		
Creating <i>CR</i> (in ms)	69	620
Editing <i>CR</i> (in ms)	15	37
Moving <i>CR</i> (in fps)	170 ^(*)	170 ^(*)

^(*) Constant, depends on the GPU's characteristics.

[Task: CREATING A COMPOSITING REGION AND MOVING IT]

Scenario: The user reveals orthoimagery through a topographic map in an arbitrarily-shaped region, and scans the map by moving that region.

QGIS	MAPublisher	MapMosaic
Tool: QGIS' raster clipping tool	Tool: Adobe Illustrator's clipping mask	Tool: MapMosaic's inspector
CREATE A COMPOSITING REGION 1. Create a new vector layer REGIONS and draw a new shape S in it. 2. Select the <i>raster clipping tool</i> in the raster menu. 3. Select the ORTHO raster layer and the newly-created vector layer REGION. 4. Specify an output file where to store the resulting raster layer.	CREATE A COMPOSITING REGION 1. Draw a shape S in the ORTHO raster layer. 2. Select both S and the ORTHO layer. 3. Create a clipping mask from the contextual menu.	CREATE A COMPOSITING REGION 1. Draw a shape S. 2. Toggle compositing ON in the inspector. 3. Select the satellite imagery layer in the inspector's dropdown list.
<i>The new layer is georeferenced but has no relationship with the source layer. Changing the coordinates of the layers will not update the region's geographical scope (neither the source layer nor the new layer).</i>	<i>Compositing regions are not georeferenced. Dragging the region will change its position on the canvas, but will not update its geographical scope.</i>	<i>Compositing regions are georeferenced. They can be dragged to update their geographical scope.</i>
MOVE IT 5. Move S to a new position. 6. Redo 2, 3, 4 each time S is to be moved.	MOVE IT 4. Release the clipping mask. 5. Move S to a new position. 6. Redo 2, 3, 4 each time S is to be moved.	MOVE IT 4. Drag-and-drop S (direct manipulation).

[Task: CREATING A FILTERED COMPOSITING REGION AND EDITING THE FILTER'S PARAMETERS]

Scenario: The user reveals orthoimagery only inside the shapes that correspond to *building* objects more than 10-meter high and that fall within a given region. He then edits the filter to relax the criterion to 20-meter high.

QGIS	MAPublisher	MapMosaic
Tools: QGIS' query editor + vector clipping tool + raster clipping tool	Tool: MAPublisher's Map Selections + Illustrator's clipping mask	Tool: MapMosaic's inspector and query builder
CREATE A COMPOSITING REGION 1. Create a new vector layer REGION and delineate the region of interest S. 2. Invoke a contextual menu on the BUILDINGS vector layer to open the query editor and filter the layer to keep only buildings higher than 10 meters. 3. Select the <i>vector clipping tool</i> in the vector menu. 4. Select the BUILDINGS and REGION layers 5. Specify an output vector layer LOCALFILTER. 6. Select the <i>raster clipping tool</i> in the raster menu. 7. Select the ORTHO raster layer and the newly-created vector layer LOCALFILTER 8. Specify an output file where to store the new raster containing the mask.	CREATE A COMPOSITING REGION 1. Delineate the region of interest S. 2. Make it into a MAP Selection (MS _{REGIONS}) using <i>Art Selection</i> in the Map Selection dialog box. 3. Create a second MAP Selection (MS _{HIGHBUILDINGS}) using an attribute filter for buildings higher than 10 meters. 4. Combine MS _{REGIONS} and MS _{HIGHBUILDINGS} by selecting one of them, turning it into an Illustrator Selection (using <i>Apply as New Selection</i>), before selecting the other to refine it (using <i>Get Subset of Current Selection</i>). 5. Put the objects of interest in the ORTHO, raster layer, group them into a compound path and turn them into a clipping mask.	CREATE A COMPOSITING REGION 1. Delineate the region of interest S. 2. Filter for buildings only in the region by ticking the <i>buildings</i> checkbox in the inspector. 3. Change the minimal height value in the query builder.
<i>There is no relationship between the created layers and the ones used for the clipping operation. As a consequence, any changes made to the source layers will not be reflected in the new layer.</i>	<i>All MAP Selections are listed in the MAP Selections panel and can be edited, but MAPublisher does not maintain any dependency relationship between compound MAP Selections and the selections they are derived from. As a consequence, any changes to the latter will not be reflected in the former.</i>	<i>Compositing regions are interactive objects whose filters remain editable in the inspector.</i>
EDIT ITS FILTER'S PARAMETERS 9. Redo 2 (setting building height to 20-meter criterion), 3, 4, 5, 6, 7, 8 and 9.	EDIT ITS FILTER'S PARAMETERS 6. Edit MS _{HIGHBUILDINGS} , updating the criterion to 20-meter. 7. Redo 4 and 5.	EDIT ITS FILTER'S PARAMETERS 4. Using the query builder, adjust the criterion to 20 meters.

Table 4.3: Comparison between QGIS, MAPublisher and MapMosaic for creating, moving and filtering compositing regions.

ics layering, masking and compositing techniques of the editor. What differentiates *MapMosaic* from these pieces of software is its simpler and more flexible interaction model, backed by its dynamic visual compositing capabilities. In this section, we compare this interaction model with that of QGIS and MAPublisher:

- QGIS is representative of what GIS applications offer in terms of data editing, publishing and analysis features. On the graphics side, QGIS provides an editor to build attribute queries that select subsets of objects from vector layers. Some visual compositing operations can be achieved using clipping tools, that enable users to set a vector layer as a mask applied to another layer, either raster or vector.
- MAPublisher allows users to produce high-quality static cartographic renderings, providing them with all the power of Illustrator's vector graphics editing features. MAPublisher augments the toolset with tools typically found in raster-image-editing software. Beyond support for geospatial data importing, MAPublisher extends Illustrator with MAP Selections, to select objects on the canvas based on both spatial and styling attributes.

Our evaluation is based on an analytical comparison of the three interaction models above, when performing the generic actions involved in dynamic compositing detailed in Table 4.3: creating compositing regions, moving them, and editing their parameters. This analytical comparison serves as a basis for a more general comparative discussion based on the "*cognitive dimensions*" framework from [51]. Originally designed as an evaluation technique for visual programming languages, many of the dimensions defined in the framework actually bear strong relevance beyond this particular type of interactive environment. They have been used to evaluate a much wider range of user interfaces. The following subsections discuss subsets of related cognitive dimensions. Some dimensions of the framework are omitted, such as *Diffuseness* and *Secondary notation*, as they do not directly apply to our specific case.

4.5.1. VISCOSITY, PREMATURE COMMITMENT AND PROGRESSIVE EVALUATION

These three dimensions relate to the flexibility of the system; in our case, how flexible is the creation and editing of layer composites. *Viscosity* captures resistance to change, *i.e.*, how much effort has to be put into making small changes (number and

complexity of the associated actions). *Premature Commitment* arises when a specific order is imposed on the sequence of actions to accomplish a goal. *Progressive evaluation* represents the capability to evaluate an action in its intermediary steps, “allowing users to stop in the middle to check work so far” [11].

As detailed in Figure 4.3, both MAPublisher and QGIS enable users to achieve elaborate graphical renderings based on layer compositing. MAPublisher, through Illustrator’s capabilities, has even more expressive power than *MapMosaic*. But achieving such elaborate renderings involves interacting with numerous interface menus and dialog boxes. Most importantly, the equivalent of *MapMosaic*’s compositing regions cannot be moved or edited: a new clipping mask has to be created from scratch whenever the position or filter settings of a region change, resulting in a very high level of viscosity, and a very high level of premature commitment.

MapMosaic’s compositing regions aim at making spatial compositing more flexible. Compositing regions can be moved by simple drag & drop. Filter settings and compositing parameters can be adjusted using a simple inspector. This significantly lowers viscosity, as illustrated on the tasks detailed in Figure 4.3, and enables progressive evaluation, as all changes are immediately reflected in the visual representation. The use of an inspector to edit the settings of compositing regions also helps address the issue of premature commitment encountered in the interaction models of QGIS and MAPublisher. The latter two impose a strong order on the specification of the different settings through dialog boxes, that do not provide much feedback until the specification is complete and the user clicks the OK button.

The second task in Figure 4.3 illustrates the difference on a concrete example: restricting compositing to buildings higher than 10m only. With *MapMosaic*, the user first enables the BUILDINGS filter, and then specifies a constraint on their height. Adjusting the minimum height to, e.g., 20m, only requires editing this value in the query builder. Creating the same filter in QGIS takes more steps. The user first has to filter the BUILDINGS layer, then use the vector clipping tool, and finally the raster clipping tool. A detailed breakdown of all these steps is available in Figure 4.3. Adjusting the minimum height to 20m requires performing all those steps again.

The interaction model of both QGIS and MAPublisher assumes that users have a well-defined end-result in mind, and that they are able to specify all compositing rules when creating the clipping mask, before actually applying it. This puts a heavy cognitive load on users, and requires much *premature commitment*. On the opposite, *MapMosaic*’s dynamic model features compositing regions that remain manipulable throughout their lifecycle, enabling better *progressive evaluation*.

4.5.2. ABSTRACTION, ROLE EXPRESSIVENESS

Abstractions are groupings of objects that should be treated as a single entity for a particular purpose. *Role expressiveness* indicates to what extent the purpose of an entity is readily inferred by users. All three systems considered here make use of abstractions to enable users to specify compositing operations, but the role of those abstractions is not always obvious.

The main abstraction in *MapMosaic* is the compositing region. All objects that fall in the region and meet the filtering criteria are treated (composited) in the same way. The primary role of the compositing region, as an abstraction, is to define the scope of the compositing, and its settings. Filters represent a secondary abstraction, tightly bound to the compositing regions, the latter remaining the main entities exposed to, and manipulated by, users.

MAPublisher relies on two rather separate abstractions. The first is a clipping mask, associated with two layers, that defines the scope of the compositing at the level of pixels. Abstracting objects rather than pixels requires first creating MAP Selections (the second abstraction), and combining them to create the aforementioned clipping masks, as detailed in Figure 4.3. These clipping masks are neither associated with the layers they were applied to, nor with the geographical position of objects they were derived from. They are generic vector objects that delineate regions. Clipping masks could thus be used for a variety of purposes, but whose specific role in the context of compositing operations is not straightforward to understand, especially considering the lack of immediate visual feedback when relocating them on the map (discussed in the previous section).

Finally, QGIS relies on one essential abstraction: the layers themselves. Achieving the same operation as in the previous two cases requires creating a temporary layer holding the delineation of the region of interest, loading the layer containing the objects involved in the filter (e.g., “all buildings higher than 10m”), using the vector clipping tool to combine them, and then using the raster or vector clipping tool to achieve the actual layer compositing with the target layer. Compared to MAPublisher, relationships between abstractions are preserved in terms of geographical location. But, as a consequence, the region of interest defined in the temporary layer cannot be moved easily, and the dependency relationships between layers are lost. Again, the lack of immediate visual feedback and the lack of support for direct manipulation makes the role of the different layers involved in the compositing process hard to understand.

4.5.3. HARD MENTAL OPERATIONS, ERROR PRONENESS

Hard Mental Operations imply that the system is putting a high demand on the user's cognitive resources. *Error proneness* is about the extent to which the interaction model invites mistakes by design, and to what extent it provides protection against certain types of mistakes.

In *MapMosaic*, compositing regions are treated as first-class objects whose properties can be edited using an inspector. The incremental and reversible nature of the creation process, coupled with the immediate visual feedback provided during construction, helps decrease errors. Users immediately see the consequences of their actions, ranging from selecting specific layers for compositing to adjusting filter settings. As illustrated in Figure 4.3, the much smaller number of actions required to create a compositing region or modify it also helps reduce the difficulty of the associated mental operations, as the degree of premature commitment is lower.

Both MAPublisher and QGIS expose lower-level, more generic abstractions to users. This means, at least in the case of MAPublisher, that users are given more expressive power. But it also means that more actions are required to achieve the same result, mechanically making them more error prone, and that planning for those actions involves more complex mental operations. For instance, in QGIS, layers represent the main abstraction. Their number grows with the complexity of the compositing configuration. Additional layers are created when adding filters within the region. Users consequently have to label those layers and remember their purpose, so as to select the correct one for each step. This also creates potential confusion between layers that actually hold content, and layers that exist solely for the purpose of compositing. In MAPublisher, compositing essentially relies on the underlying generic mechanism offered by Illustrator. The vector shapes that will be used for clipping must thus be instantiated in the layer that contains the raster or vector objects to be cropped, which can lead to mistakes. Similarly, applying a filter relies on the generic set operations performed on shapes provided by Illustrator, requiring users to create two MAP Selections, and to select them in the correct order so as to subtract one from the other as intended.

4.5.4. VISIBILITY AND JUXTAPOSABILITY

Visibility refers to the system's capacity to readily make components-of-interest visible in the workspace. Juxtaposability refers more specifically to the capability of putting components side-by-side, which eases tasks related to their comparison.

The settings of a compositing region in *MapMosaic* are made visible in one single place, the inspector. Compositing regions can be freely relocated, making it possible to juxtapose them. However, the content of *free* compositing regions updates automatically as the region is dragged by the user, thus preventing side-by-side comparison of distant regions that do not fit in the same viewport. *Docked* regions help alleviate this problem, but in the current interaction model those regions revert back to their original location as soon as they are released. The model could be extended to enable the temporary pinning of docked regions, which would enable the juxtaposition of more than two of them.

In QGIS and MAPublisher, compositing settings are made visible through other means, since the abstractions used to achieve compositing are different, as discussed above. In QGIS, users can toggle layer visibility to display the intermediate layers involved in the compositing process. MAP Selections, associated with clipping masks in MAPublisher, can be displayed by making them the active selection in the workspace. Modifying their parameters is typically achieved using pop-up dialog boxes. The geometric shapes involved in compositing (masks and selections) can be moved at will, enabling the side-by-side comparison of multiple – possibly distant – regions. Indeed, the content of a mask gets rendered at the time it is created. It never gets updated afterwards, meaning that moving the shape that delineates it to other geographical coordinates will not change its content. As discussed earlier, this behaviour negatively affects progressive evaluation, and increases viscosity. But it also favors juxtaposability. A side-effect of this behavior, however, is that changes to the content that are performed after the mask has been created will not be visible in the region that was composited in the first place.

4.6. EXPERTS' FEEDBACK

Different articles have been promoting empirical research in the visualization community starting approximately ten years ago – see, e.g., [22] and [138] – and an increasing number of papers are now reporting on user experience or user performance evaluations [83]. Formal testing sessions that involve basic tasks, while required to evaluate *user performance*, seldom yield meaningful results [46] when evaluating the usefulness of systems such as MapMosaic for performing complex tasks, both because of the open-ended nature of cartographic and geovisualization activities [22], and because of the need for users to feel engaged with the data [88]. *User experience* evaluation, “*eliciting subjective feedback and opinions [about a visualization system]*” [71], is more relevant in this context. Some studies evaluate usability

by asking users to perform specific tasks [131], possibly using tailor-made applications [76]. However, we were interested not only in evaluating *MapMosaic* for certain tasks or usage scenarios [138], but also in finding specific contexts in which it would be useful. We thus chose another method: organizing workshop sessions to collect feedback from experts, with the aim of “*maximiz[ing] the realism of the findings*” [71]. Participants were encouraged to talk about their specific activities and then engaged into discussions about how *MapMosaic* could be helpful in such contexts.

4

In the first session, we invited six operators from the *Database Curation* and *Land Cover & Use* departments at IGN to give feedback on *MapMosaic*. None of them had participated in the interviews reported in Section 4.2. In the second session, we met the researcher who works on environmental risk management, that we had already talked to during an interview session. Both workshop sessions were conducted in a meeting room at IGN. They started with the participants describing their daily tasks or research interests informally in about 10 minutes, followed by a 20-minute presentation of *MapMosaic*’s features by the authors. Participants were then invited to comment on *MapMosaic* and to describe in which scenarios it would be helpful. They all commented very positively on the directness of the interaction model. The capacity to provide immediate visual feedback when moving compositing regions and when adjusting their parameters was also well received, suggesting that the principles that drove the design of *MapMosaic*, such as direct manipulation, reification, polymorphism and reuse, are indeed relevant in this context. In both sessions, we had rich discussions (~2 hours each) about scenarios in which *MapMosaic* would be useful. This section summarizes these discussions.

4.6.1. CARTOGRAPHIC DATA MAINTENANCE

Operators found it particularly useful to be able to quickly switch back-and-forth between layers by selecting a region on the map, compositing it with a different layer, and temporarily moving it to collect and correlate information from the different sources. For instance, *Land Cover & Use* operators have to categorize forest areas as coniferous or deciduous based on 4-band (RGB+infrared) orthoimagery. Some operators rely on the RGB rendering only, others on the RG-Infrared rendering only, and yet others switch between both. As discussed earlier, the best representation not only depends on an area’s actual content, but also on the work habits of individual users. Operators mentioned that *MapMosaic*’s local compositing capabilities make the customization of the visual representation easy and flexible.

Operators also saw great value in the display and dynamic update of query re-



Figure 4.4: (a) Highlighting all buildings in a given geographical area using color compositing. (b) Dynamically adjusting query parameters to highlight buildings taller than 10m only.

sults in the map view. The user interface front-end they currently use (QGIS) displays query result-sets as textual lists, making it difficult to spatially relate items and to get feedback about the impact of query parameter adjustments. One of the operators described a case he was currently working on, that involves fixing 3D building models. The process is semi-automated. It relies on an algorithm that compares the current 3D model with elevation models, applying corrections to it when finding inconsistencies. Operators then perform a manual (one by one) check of all corrections considered too large, according to a given threshold (for instance: “altitude modified by more than 5 meters”). Getting a clear picture of the spatial relationships between similar corrections (according to such a threshold) would help operators diagnose possible common sources of errors. A typical common source mentioned by one of the operators was the use of a lower-precision distance sensor in one particular area, which would have been diagnosed more quickly, had he been able to visualize the candidate corrections spatially and interactively, as *MapMosaic* enables him to.

Operators also appreciated the persistence of spatial queries and the possibility to adjust attribute filters at any moment using the inspector. An operator from the *Database Curation* department mentioned that people in his group have to set thresholds on several attribute values when starting to work in a new area. This enables them to better anticipate the amount of work, and set a reasonable level of precision for edits that is compatible with the time resources allocated to the project. Going back to the above example about correcting building data: operators might have initially set 1m as the minimum error length, and then increased it to 5m when realizing that the initial value would have yielded too many buildings to check. Finding the right threshold is tedious with current tools, as operators have to create a new

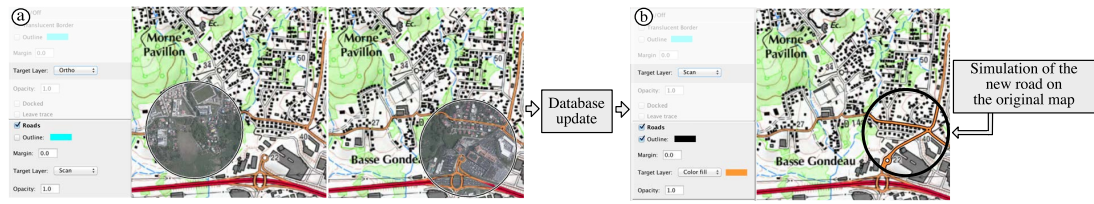


Figure 4.5: (a) Scanning ortho-imagery for new roads. (b) A compositing region with styling rules to preview the new road on the map.

4

query each time they want to test a new value. Using *MapMosaic*'s query builder, they could easily edit the filters dynamically and immediately see the result in the map view (Figure 4.4). Another operator suggested creating multiple compositing regions configured with similar queries, differing only in their filter settings, and that would coexist in the same map view.

The *Database Curation* department is in charge of ensuring the completeness and timeliness of the main database. Operators work at the highest level of detail, and only get updates of the maps derived from their data every three months. They thus have to wait a long time before seeing the results of their modifications in the products made by the institute. An operator came up with the idea of using *MapMosaic* for “rapid symbolization”: following modifications made to the database, he would create a compositing region that approximately simulates the symbolization used for the considered type of object, and put this region over the corresponding area on the map to get a preview of the newly-added object. In Figure 4.5, an operator identifies a missing road, adds it to the database, and creates a compositing region on vector road elements, styling them to roughly match their actual map symbolization: black outline, solid orange fill as target layer, dilation by a few pixels.

4.6.2. CRISIS MANAGEMENT

The presentation of *MapMosaic* led to a rich discussion with the researcher in risk management. He quickly came back to the usefulness of *MapMosaic*'s techniques for one of the activities discussed during the interview (Section 4.2): rapid mapping, which consists of creating maps in very short timeframes (between 6 hours and a day) to provide support to crisis management authorities. Such activities are performed by experts, as well as volunteers [109] through Web sites such as Tomnod (<http://www.tomnod.com>).

Figure 4.6 illustrates our interviewee's idea about how *MapMosaic* can help perform tasks such as rapid damage assessment. These tasks typically involve comparing satellite imagery captured before and after the event. While they provide valuable



Figure 4.6: Nepal earthquake, 2015. Compositing vector objects to quickly identify buildings of interest and make a preliminary damage assessment. (a) Close-up on a building, showing imagery captured before the disaster. (b) Compositing post-event imagery inside these buildings only.

information, it can be hard to identify what areas have suffered significant damage. It can also take time to identify specific points of interest such as critical infrastructure just by looking at those images. *MapMosaic* can help focus users' attention on specific areas, first by compositing vector data with the satellite imagery to emphasize buildings of interest, and by restricting the compositing of before-and-after images to the interior of those vector objects. Applying some dilation to these compositing regions may help better assess the amplitude of the damage. Once categorized as, *e.g.*, *inaccessible* or *destroyed*, buildings, roads or even entire areas, which have already been transformed into compositing regions, can be composited with the corresponding solid color to make them easily identifiable.

Similarly, *MapMosaic* could help users inspect the results of algorithmic techniques that perform automatic classification on remote sensing imagery. Indeed, such inspection tasks often require moving back and forth between the layer holding the classification results, the layer holding the imagery on which the classification was performed, and possibly the layer holding imagery of higher spatial resolution for the considered area, that provides the ground truth.

The researcher in risk management also brought up a related use case. Images of the same area acquired from different satellites often suffer from distortion and parallax effects, which can be partially corrected through image registration. GIS front-ends let users register two images by identifying multiple locations in both rasters and then deriving the transformation from this set of matching pairs of points. The simplest interface juxtaposes both images and waits until users have input all matching locations before computing the transformation. But as the number of pairs re-

quired can be fairly high, depending on how complex the transformation is, some front-ends such as ArcGIS enable a more iterative process. They overlay the two images and dynamically adjust the transformation whenever the user inputs a new pair of points. This can be very useful, but it also makes interaction more cumbersome, as users have to frequently toggle between both image layers. In such situations, a compositing region could be displayed after the first point of a pair has been input to reveal the other layer in the vicinity of the cursor. It would let users scan for and select the matching point, and then disappear.

4

4.7. SUMMARY

MapMosaic defines a user interaction model based on the concept of dynamic *compositing regions*, that facilitate the spatial multiplexing of multiple map layers. Compositing regions can be moved freely on the map, and their parameters, including attribute filters, can be changed on the fly. Updates happen in real-time at interactive frame rates, providing users with immediate visual feedback and enabling them to explore the data using dynamic queries. This new model was well-received by a panel of GIS experts, who identified several concrete scenarios in which it would be useful.

Compared to existing editors, *MapMosaic*'s dynamic, local, and more interactive approach to visual compositing aims at better supporting users' train of thought, and should be less cognitively demanding [51]. Further user studies need to be conducted to verify this empirically. Beyond expert users, we believe that *MapMosaic* can be useful to a wider audience, including lay users who want to design custom maps: marking an itinerary or making some entities or areas more salient using effects such as *selective brightening* or *spotlight highlighting* [105], which can all be achieved effortlessly with compositing regions. *MapMosaic* could also be useful for collaborative mapping [44] to, e.g., inspect the history of objects, or the activity of collaborators by organizing changes into layers, per revision or per user.

5

ANIMATION PLANS FOR BEFORE/AFTER SATELLITE IMAGES

Main portions of this chapter are under review as a journal submission.

In this manuscript's introduction, we present the four types of activities that users perform with maps: *exploration*, *confirmation*, *synthesis* and *presentation* to an audience [97]. While spatial multiplexing techniques can provide a good support for comparing and exploring multi-layer representations, they might not be the best option for designing presentation maps. Using spatial multiplexing only leads to static presentation maps that may not be effective for some type of representation, and type of phenomenon to illustrate. For example, superimposing layers that have similar representations (*e.g.*, two topographic maps or two satellite images) might lead to maps that are confusing, because users can have problems determining which layer the objects belong to. More advanced techniques that automatically detect the differences between images, such as in VAICo [137], can be effective at showing differences but support more analytical forms of comparisons aimed at expert audiences. More importantly, static maps might not be effective when the phenomenon of interest involves changes over time.

Going back to the “after crisis” scenario that we mention in the last chapter, experts and crowd-source participants need to compare before-and-after images to identify damaged areas. In this case, they work with pairs or longer sequences of images of the same geographical area (aligned using image registration techniques [41]), taken at different points in time, that show how that region has been impacted by natural phenomena such as earthquakes and tsunamis [102]. Before-

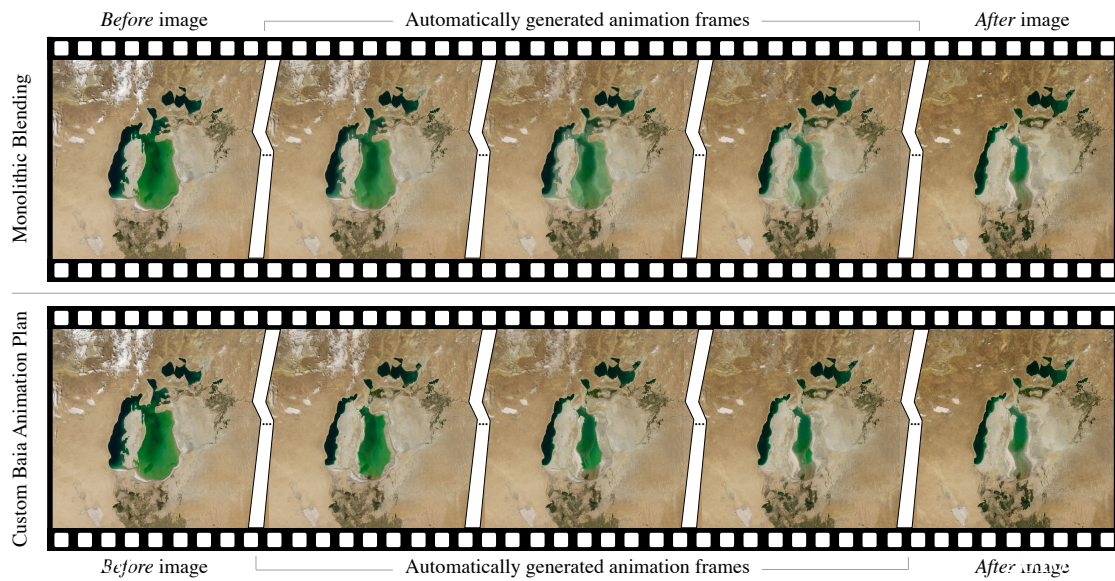


Figure 5.1: Keyframes from the Aral sea animation generated using a *before* image from 2000 and an *after* image from 2010. Top row: keyframes of an animation generated using classic monolithic blending. In the intermediate keyframes, both states of the sea in 2000 and 2010 co-exist, visually interfering with one another as 2010 fades in and 2000 fades out. Bottom row: keyframes of a Baia animation plan (Figure 5.6). There is no such visual interference, and the sea actually gives the impression of shrinking. Source: NASA.

and-after images can also be used to study other types of natural changes such as landscape evolution due to climate change. As mentioned in the previous chapter, spatial multiplexing can provide a good support for identifying changes. However, when it comes to presenting those changes to an audience, there is a need for conveying the phenomenon dynamism in order to better explain the transition between the before and after states. In such cases, temporal multiplexing has been used but mostly through simple animations to transition between the two images, such as what we call monolithic blending, *i.e.*, the classic, simple technique that consists in smoothly transitioning between two images by gradually increasing the transparency of the source image drawn on top of the destination image, considering the entire image as a single, uniform block. Even if this simple strategy can provide good support to analysts for change detection and comparison tasks, they often fail to draw attention on, and effectively illustrate, the observed phenomena to wider audiences.

In this chapter, we explore time multiplexing to enable presentation of the geographical evolution between pairs of before-and-after satellite images. We introduce *Baia*, a framework to create more meaningful animated transitions between a pair of images (Figure 5.1): animations that better illustrate specific changes, and that help focus the audience's attention on those changes. The approach relies on *animation plans*, a pixel-based transition model that gives authors much flexibility in

the specification of transitions. We describe this model, as well as the accompanying set of predefined animation primitives that capture common types of changes. Next, we report on a user study in which elaborate transitions enabled by Baia were perceived as more realistic and better at focusing viewer’s attention on specific changes than monolithic blending. Finally, we describe a prototype animation plan authoring tool inspired by timeline-based video editing applications, and report on a small user study to gather feedback about its usability.

5.1. BACKGROUND AND MOTIVATION

5.1.1. SLIDESHOW-STYLE ANIMATIONS

Animations have been used in visualization to convey different kinds of changes: changes to the data themselves, such as the addition, modification or removal of data elements in the representation [5], changes of viewpoint on the representation [123], and even changes of representation type [65]. Animations provide strong support for object constancy [123] and can also suggest chains of causality. They thus play an important role in visually-illustrated narratives [65, 139].

Animations have also proven useful in geographical applications, as they are natural candidates for representing spatio-temporal data and processes [60]. However, most animations focus on abstract vector data. Authoring animations for remote sensing imagery is more challenging, as the content of raster images cannot be manipulated as easily as vector graphics.

When authoring presentations that involve sequences of raster images, authors have to choose from a limited set of options in terms of transitions. One image will be replaced by the next, either abruptly swapping or gradually blending between them (what we called *monolithic blending* earlier, illustrated in the top row of Figure 5.1). Such temporal composition can work better than static representations [110], and online tools such as MODIS-VIIRS GIF Creation¹ make their construction easy.

This type of slideshow-style animation has been used to show, *e.g.*, seasonal changes [58], to illustrate the effect of climate change [23] and major construction projects [156], or to show predictions of future conditions such as sea-level rise [152]. But according to Lobben [89], these slideshows are not real animations, as the audience can clearly detect the moment when a new slide (or frame) replaces the previous one. Animations are supposed to generate smoother transitions, in which the audi-

¹https://remotepixel.ca/projects/viirsmodis_gif.html



Figure 5.2: Images showing Lake Powell's evolution between 2002 and 2004. The only area that varies significantly across the three images is the lake itself, making it relatively easy to track changes. Source: NASA.

5

ence does not detect such discrete transitions that break the temporal continuum.

5.1.2. SUPPORTING PROCESS ANIMATIONS

Simple transitions can be effective when the different snapshots are very homogeneous, only differing in the actual element of interest (Figure 5.2), and when the evolution of that element is straightforward to understand. However, if the evolution is complex, as is often the case when dealing with what Lobben calls *process animations* [89], simple transitions might not convey the nature of the corresponding changes in the most meaningful manner. For instance, the before-and-after images might differ too much, or miss key intermediate steps, making it difficult for the audience to understand the transformation.

To our knowledge, process animations have not received much attention in the context of remote sensing imagery, except for some image-morphing-based techniques designed for highly-specific types of data [158]. Generally-speaking, morphing techniques are used to transition between two raster images, one image being smoothly warped and blended into the other based on user-specified keypoints or subareas [154]. Some algorithms use morphological image operations, entirely removing the need for keypoint specification [72, 118]. Morphing has been used to transition between maps featuring incongruent projections [?], as is typically the case when historical maps are involved. Morphing does not play a central role in our approach, but is rather used as a means to compute some of the more elaborate pixel-based animations generated by Baia's deformation primitives.

Another important issue to consider is that before-and-after image pairs may also



Figure 5.3: Before & after the 2015 Nepal earthquake. When swapping or blending between the two images, changes in color histograms, shadows, and the shelters set up on the Tundikhel field (located on the right) make it difficult to focus attention on damaged areas such as the one highlighted in the red circle. Source: Airbus/Pléiades.

feature changes unrelated to the elements of interest, such as, *e.g.*, varying cloud cover, large differences in color histograms, other elements appearing, disappearing or moving. Simple transitions might then make it difficult for the audience to focus their attention on the evolution of the elements of interest, as multiple areas will be changing concurrently, interfering with them (Figure 5.3). As Tversky *et al.* state, transitions should be kept as simple as possible, creating animations that are “[...] *slow and clear enough for observers to perceive movements, changes, and their timing, and to understand the changes in relations between the parts and the sequence of events*” [151]. But they should also be effective at focusing the audience’s attention, and at conveying the nature of changes following the *principle of congruence* also from Tversky *et al.* [151], which in this case advocates for establishing “*a natural correspondence between change over time [...] and the essential conceptual information to be conveyed*” (here, the phenomenon to be illustrated).

5.2. ANIMATION PLANS

Chevalier *et al.* define an animation as “*a (usually perceptually continuous) sequence of intermediary images that give the illusion of a smooth progression from a transition’s initial visual state to its final visual state*” [25], where a transition is a “*pair of visual states - an initial one and a final one.*” The animation model that provides the foundation of our framework, *Baia*, automatically generates sequences of intermediate images (Figure 5.1), enabling smooth transitions between before images (initial state) and after images (final state).

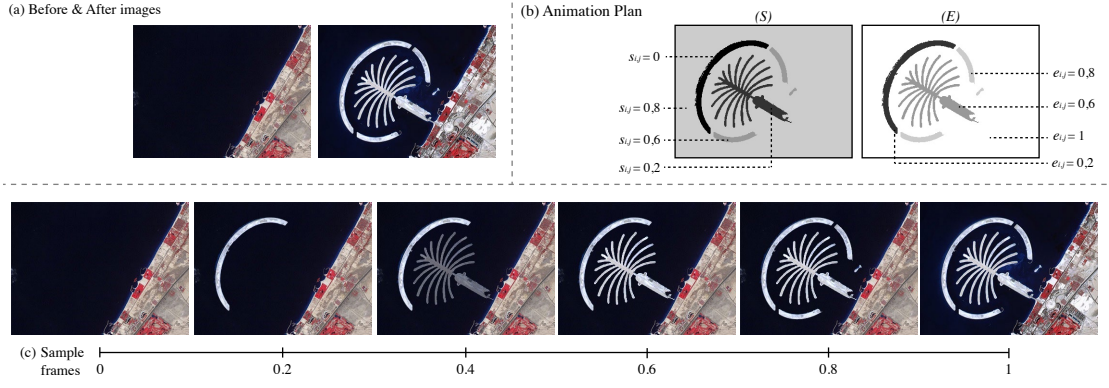


Figure 5.4: Example illustrating how animation plans work: (a) before-and-after image pair showing land reclamation in Dubai (construction of the Palm Islands); (b) animation plan matrices S and E specify that: the pixels from the *before* image that belong to the outer island arc start blending at the very beginning of the animation ($s_{i,j} = 0$) and have been fully replaced by the corresponding pixels in the *after* image at 20% of the animation's course ($e_{i,j} = 0.2$); pixels in the palm-tree itself start blending when the animation reaches 20% of its course ($s_{i,j} = 0.2$), and have been fully replaced when it reaches 60% ($e_{i,j} = 0.6$); the two lower arcs are animated next, between 60% ($s_{i,j} = 0.6$) and 80% ($e_{i,j} = 0.8$); finally the rest of the image starts blending at 80% ($s_{i,j} = 0.8$) and gets fully replaced at the very end of the animation ($e_{i,j} = 1$). (c) Keyframes from the animation.

5

Baia's animation model is based on what we term *animation plans*. An animation plan is a pair of matrices S (for *start*) and E (for *end*) whose dimensions match that of the before-and-after images. The two matrices respectively specify, for each pixel in the before image: (S) at which moment in the course of the whole animated transition does that pixel start to be gradually composited with the corresponding pixel in the after image, and (E) at which moment does this gradual compositing stop, the pixel from the after image having fully replaced the pixel from the before image. Values in the matrices can thus be seen as time stamps that mark, independently for each individual pixel at coordinates i, j , the beginning ($s_{i,j}$) and end ($e_{i,j}$) of their transition. These time stamps are encoded as normalized values, and the end time stamp can be equal to, but not less than, the start time stamp ($\forall i, j : s_{i,j} \in [0, 1], e_{i,j} \in [0, 1], s_{i,j} \leq e_{i,j}$). If $s_{i,j} = e_{i,j}$, then the *before* pixel gets swapped with the *after* pixel at the specified moment, instantaneously. If $s_{i,j} < e_{i,j}$, then the before pixel gets smoothly blended with the after pixel using OVER alpha compositing [?], the value of α being linearly interpolated from 0 to 1 between times $s_{i,j}$ and $e_{i,j}$. As interpreting the visual rendering of animation plans is not straightforward, we illustrate in Figure 5.4 how they work on a schematic example. We then discuss more realistic examples.

Values in S and E close to 0 mean that corresponding events happen early in the animation. Conversely, values closer to 1 correspond to events that occur late. When

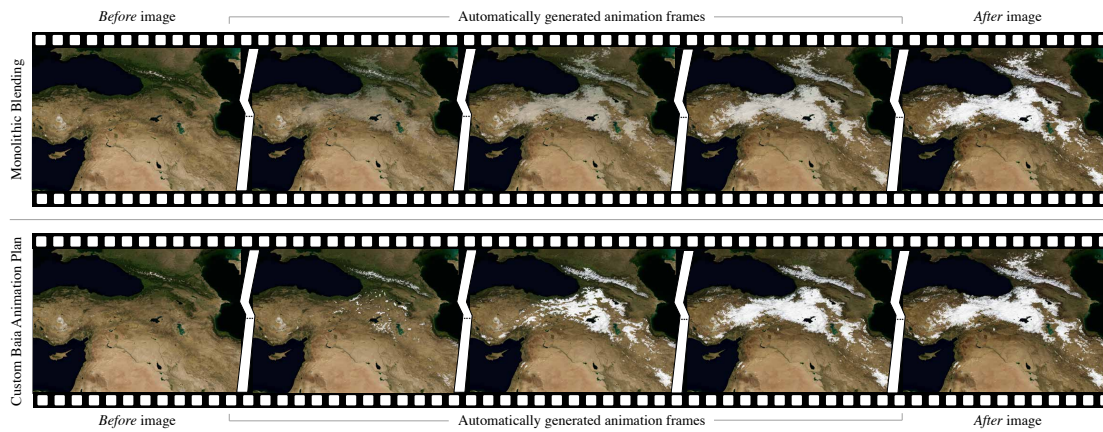


Figure 5.5: Animated transitions based on one single before-and-after image pair (NASA Blue Marble Next Generation – August & December 2004) showing seasonal snow cover over northern Middle East. The top row shows keyframes generated using basic monolithic blending. Snow fades in gradually but uniformly, regardless of altitude. The bottom row shows keyframes generated using a Baia animation plan (Figure 5.7) derived from a Digital Elevation Model. Snow fades in gradually, but this time spreading from high-altitude to low-altitude areas. Source: NASA.

5

mapped to a grayscale gradient: dark pixels in S start blending early, while bright pixels start blending late; dark pixels in E finish blending early, while bright pixels finish blending late. See Figure 5.4-b for a simple example.

This pixel-based model is flexible. It makes it possible to create more meaningful transitions than what can be achieved with a monolithic blending function between the before and after images. In particular, it enables the creation of animations that can focus users' attention on a specific region of interest, and that can convey the changes' dynamics in a manner that better reflects the temporal evolution of the actual phenomenon that impacted the region, as we discuss in the next section.

Focusing on a region of interest. As mentioned earlier, two images of the same geographical area can feature many differences (Figure 5.3), making it difficult to focus on changes of actual interest. Staged animations can guide viewers' attention to a particular area by first animating changes of interest and then all other areas; or conversely, starting with the other areas first. Both options can make sense, depending on the context in which the animation author wants to reveal those changes. As mentioned by Heer and Robertson [65], staging a complex transition to break it up into a series of simple transitions can help support Tversky's apprehension principle [151] by drawing the audience's attention on one single change at a time. Animation plans make it easy to define such stages. Coming back to the example in Figure 5.4, the animation is broken down in four stages: first the upper island arc appears, then the palm-tree itself, then the lower arcs, and finally the coast gets updated.

Conveying changes' dynamics. By controlling the order in which *before* pixels

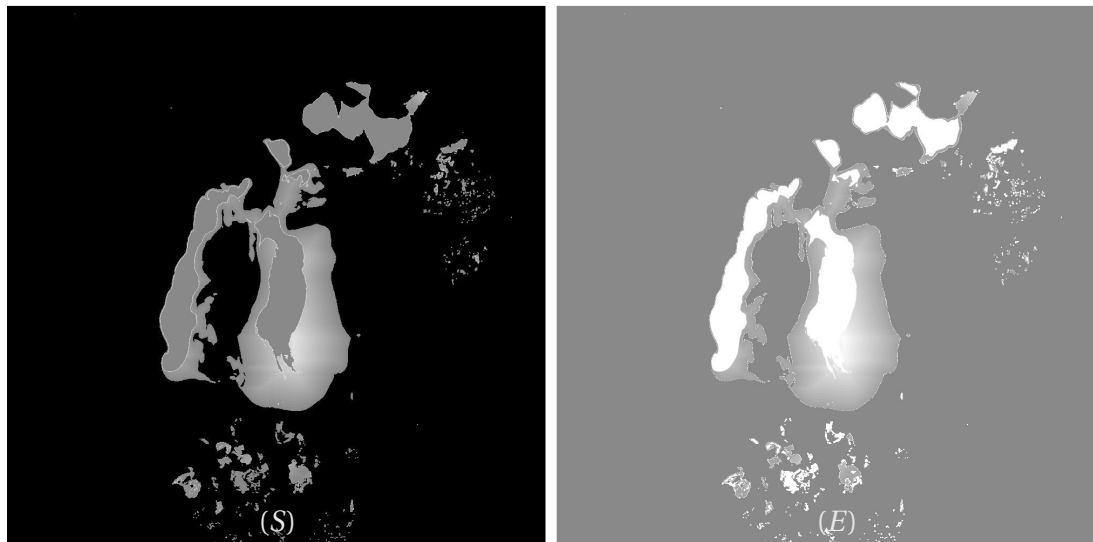


Figure 5.6: Matrices S and E of the animation plan created using *Baia*'s DEFORMATION primitive to generate the transition in Figure 5.1. The land surrounding the sea gets blended first, then the sea gradually shrinks.

get swapped or blended with *after* pixels, an animation plan can approximate the dynamics of some changes, better illustrating how the corresponding entity has evolved over time. Figure 5.1 shows some keyframes from an animation that gradually blends (inward) pixels that belonged to the Aral sea in 2000 but that were no longer part of it in 2010. Figure 5.6 shows the matrices that define the corresponding animation plan, which approximates the phenomenon and was created using the DEFORMATION animation primitive (discussed later). The grayscale gradient specifies that the closer the pixel to the 2010 contour, the later its animation will start. Figure 5.5 illustrates another way of conveying dynamics by approximating the actual phenomenon in the animation plan, comparing it to monolithic blending. Figure 5.7 shows the corresponding animation plan. In this case, *before* pixels get swapped with *after* pixels instantaneously, but at different times, determined by a digital elevation model (DEM) of the geographical region.

In our prototype implementation, animation plans are encoded as TIFF files, matrices S and E being stored in bands R and G (band B remaining unused for now). This way, animation plans can be displayed using any general-purpose image viewer that supports selecting and rendering RGB bands separately. Raw animation plans can actually be modified and even created from scratch directly with an image editor. However, this is tedious in most cases. We have thus designed and implemented a prototype animation plan authoring tool as part of the *Baia* framework, whose user interface is described later in the paper. The tool features the predefined animation primitives described next, that build on top of the core pixel-based animation model

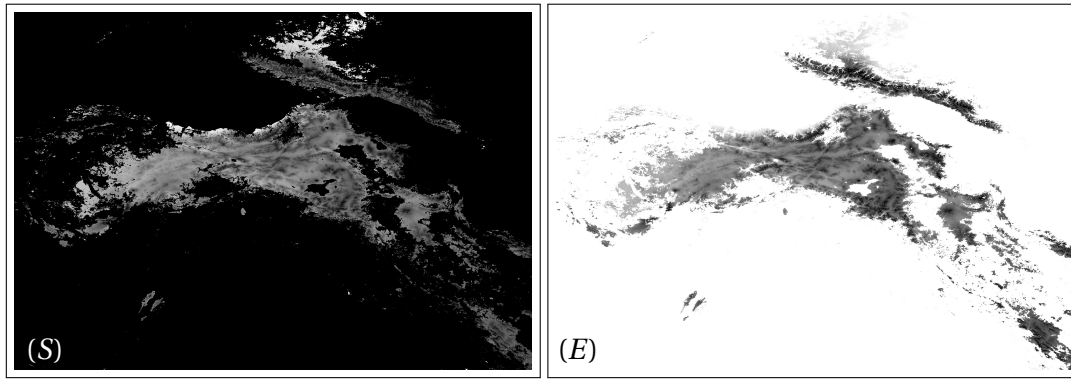


Figure 5.7: Matrices S and E of the animation plan derived from a digital elevation model (METI/NASA ASTER GDEM V2, 2011), that was used to generate the transition in Figure 5.5. Values in the two matrices corresponding to snow-covered areas are identical: pixels get swapped instantaneously, but following a spatial pattern based on terrain elevation.

to make animation plans easy-to-author for a wide range of users.

5

5.2.1. PREDEFINED ANIMATION PRIMITIVES

In order to keep transitions for common types of changes easy to create, *Baia* features a set of predefined animation primitives that can generate the corresponding animation plans automatically. This set is based on the following primitive changes from Claramunt and Thériault's modelling of temporal GIS [28]: APPEARANCE, DISAPPEARANCE, CONTRACTION, EXPANSION, DEFORMATION. Several of these primitives, such as APPEARANCE and DISAPPEARANCE, are straightforward to map to locally-bounded blending operations. Other primitives require more explanation.

- **CONTRACTION** corresponds to shapes that become smaller, such as a glacier retreating. We approximate contraction phenomena by generating animation plans that progressively blend pixels that fall in the shape's contour in the *before* image. Start values $s_{i,j}$ and end values $e_{i,j}$ are set proportionally to the pixel's distance to the new contour in the *after* image. To achieve this, we define two binary masks for the shape, one in the *before* image (M^s) and the other in the *after* image (M^e). We apply an erosion filter to M^s with a structuring element of 3×3 until it matches M^e . In the absence of an end mask M^e , the shape contracts until it disappears. E^k is the matrix obtained after applying the erosion filter at iteration k . For each iteration, we add the matrix to our animation plan, resulting in:

$$s_{i,j} = e_{i,j} = \sum_{k=1}^n (E^k_{i,j})$$

where n is the total number of iterations. Finally, we normalize the matrices' values. Pixels that belong to the intersection of both masks or that belong only

to end mask M^e are set to: $s_{i,j} = 0$ and $e_{i,j} = 1$.

- **EXPANSION** corresponds to shapes that become larger, such as a river flooding. We approximate expansion phenomena in a similar manner: we generate animation plans that progressively blend pixels that fall in the shape's contour in the *after* image. Start values $s_{i,j}$ and end values $e_{i,j}$ are set proportionally to the pixel's distance to the new contour in the *after* image. To achieve this, we define the same two binary masks as above. We apply a dilation filter to M^s with a structuring element of 3×3 until it matches M^e , and call D^k the matrix obtained after applying the erosion filter at iteration k . For each dilate iteration, we add the matrix to our animation plan, resulting in:

$$s_{i,j} = e_{i,j} = \sum_{k=1}^n (D_{i,j}^k)$$

where n is the total number of iterations. Finally, we invert the gradient in the animation plan (simply reversing its direction) and we normalize the matrices' values. Pixels that belong to the intersection of both masks, that belong to M^s only, or that belong to M^e only and are not connected to M^s , are set as follows: $s_{i,j} = 0$ and $e_{i,j} = 1$.

- **DEFORMATION** corresponds to shapes that present changes that are not a simple contraction or expansion. We propose two approaches to generate animation plans for this kind of transition. The first one superimposes the shape's contour in the *before* image on the shape's contour in the *after* image. It leaves the intersection between both contours untouched, contracts the parts that disappear, and expands the parts that appear using the filters described above for **CONTRACTION** and **EXPANSION**. However, this approach does not always provide convincing results, especially when non-convex shapes are involved, as these filters do not necessarily preserve the topology. Our second approach, **VECTOR MORPH**, addresses such cases. It is based on shape morphing [87] and enables more elaborate deformations. Using OpenCV and the approach described in [87], we compute vertex correspondences between the two contours, and generate linear vertex paths for each pair of matched vertices between the start (P^0) and end (P^1) polygons. The matrix H^k corresponding to each iteration k holds polygon P^k , generated according to the previous linear paths: $P^k = (1 - k)P^0 + kP^1$. For each iteration k , we add the matrix H^k to our animation plan, resulting in:

$$s_{i,j} = e_{i,j} = \sum_{k=1}^n (H_{i,j}^k)$$

where n is the total number of iterations. Finally, as in the case of EXPANSION, we invert the gradient (where the shape has expanded), and we normalize the matrices' values.

Baia features additional generic primitives for radial and directional progression, that visually translate to radial and linear gradients in animation plan matrices. As hinted at earlier, animation plans can be generated by any means, as long as the result is normalized and encoded in the first two bands of a TIFF file. For instance, Figure 5.7 shows the two matrices S and E of an animation plan derived from terrain elevation data fetched from the Web and pre-processed before import.

5.3. USER STUDY

Animation plans make it possible to control a transition between two images at the pixel level, where each pixel can blend independently from one another. We hypothesize that this fine-grained control enables the design of custom animations that can convey a more elaborate semantics than monolithic animations can. Because animations can serve different purposes, a measure of their efficiency is dependent on what the animation's author wants to communicate.

We ran a study in the scenario where the goal of the animation's author is to illustrate a geographical phenomenon in a realistic manner. In this specific context, an animation can be considered as efficient when it puts the emphasis on the region that is affected by the phenomenon, and when it illustrates how the region got affected over time. We hypothesize that, in this context, animation plans can be used to design custom transitions that should be effective because of the following two properties.

- Custom animations can delineate the region of interest (RoI) from the background (BG)² by putting the animation of the RoI and that of the BG in sequence. This should help draw the audience's attention to the region of interest in comparison with transitions that animate both regions concurrently (H_{focus}).
- Custom animations can gradually reveal pixels in the RoI, according to the progression of the actual phenomenon to illustrate. This should provide a more accurate and thus more realistic illustration of how the RoI got modified over

² *Background* refers to all areas in the image not affected by the geographic phenomenon of interest that the audience should focus on.

time in comparison with animations that consider the RoI as a uniform area ($H_{realism}$).

In order to operationalize the above hypotheses, our experiment involves animations that differ according to two main factors, $RoI_{anim} \in \{BG_CONCURRENT, BG_SEPARATE\}$, and $RoI_{reveal} \in \{UNIFORM, PROGRESSIVE\}$, and the experimental task requires participant to rate the different animations according to their capacity to **Focus** attention on the RoI, and according to their **Realism**.

5.3.1. PARTICIPANTS AND APPARATUS

Sixteen unpaid volunteers (seven females), daily computer users, age 24 to 41 (average: 29.6, median 28), served in the experiment. All had normal or corrected-to-normal vision and did not suffer from color blindness. The experiment was implemented using OpenGL in Java (JOGL), and ran on a MacBook Pro Retina equipped with an NVIDIA GeForce GT 750M 2048 MB graphics card connected to an external 2560×1440 pixels Apple Thunderbolt display.

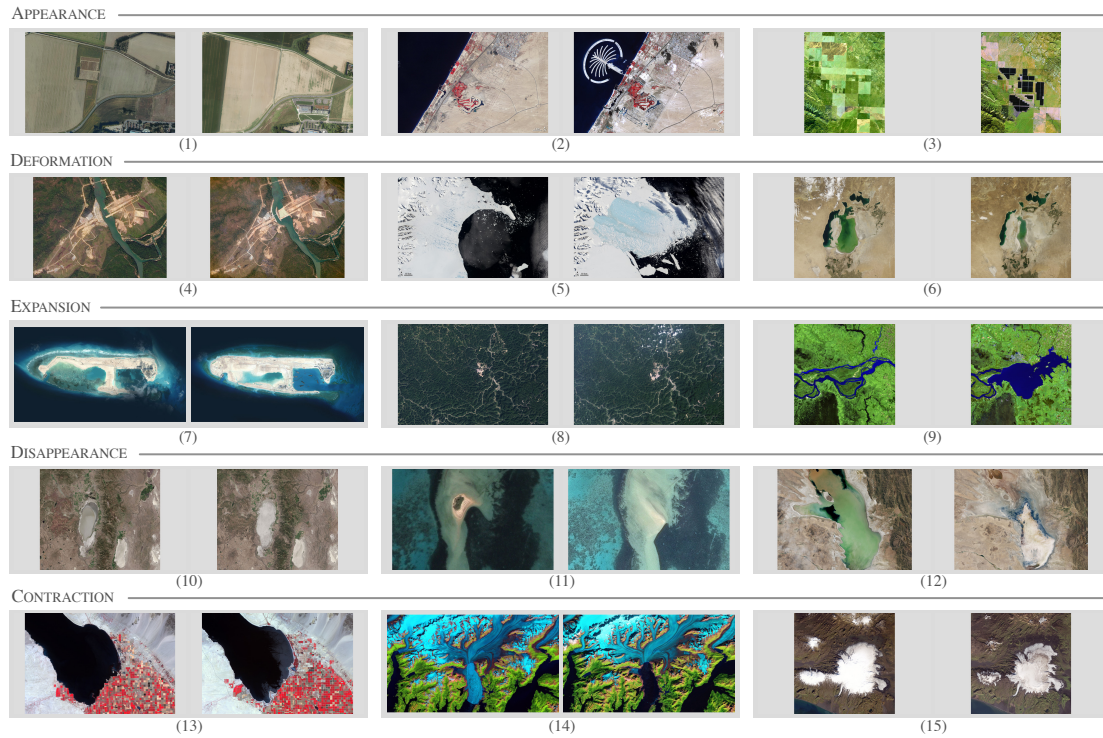
5.3.2. PROCEDURE

We followed a $5 \times 2 \times 2$ within-subject design with 3 factors: animation *Strategy* We followed a $2 \times 2 \times 5$ within-subject design with 3 factors: RoI_{anim} , RoI_{reveal} and geographic *Change*. Geographic *Change* corresponds to one of the five types of entity evolution detailed earlier, taken from [28]: [APPEARANCE, DISAPPEARANCE, CONTRACTION, EXPANSION, DEFORMATION]. In order to collect enough measures for statistical analysis, we sought three pairs of *before* and *after* images for each *Change* using online sources. Figure 5.8 lists all image pairs, along with their source. All fifteen pairs were shown to each participant. The presentation order was randomized.

For each before-and-after image pair, participants completed a 3-step trial. Figure 5.9 illustrates the trial corresponding to image pair (11) in Figure 5.8. In each step, the interface features: a textual description of the phenomenon (top),³ two animations presented side by side (center)⁴, and a set of input widgets (bottom). Each animation is displayed in a 1270×1138 pixels panel. Participants are free to play each animation as many times as they want, and can navigate through the animations' frames using a slider. For each of the three steps, they indicate, between the two

³All fifteen descriptions are available on the Web page provided as supplemental material. See details in the note below Figure 5.8.

⁴We counterbalance the on-screen position of the two animations.



- (1) <http://professionnels.ign.fr/orthoimages> (48°42'46.1"N 2°09'57.3"E),
 (2) <http://earthobservatory.nasa.gov/Features/WorldOfChange/dubai.php>, (3) <http://climate.nasa.gov/images-of-change?id=541#541>,
 (4) <https://www.planet.com/gallery/> Lower Se San 2 dam, Cambodia, (5) <http://earthobservatory.nasa.gov/Features/WorldOfChange/larsenb.php>,
 (6) <http://earthobservatory.nasa.gov/Features/WorldOfChange/araksea.php>, (7) http://www.nytimes.com/interactive/2015/07/30/world/asia/what-china-has-been-building-in-the-south-china-sea-2016.html?_r=0,
 (8) <http://earthobservatory.nasa.gov/Features/WorldOfChange/hobet.php>, (9) <http://climate.nasa.gov/images-of-change?id=335#335>,
 (10) <http://climate.nasa.gov/images-of-change?id=555#555>,
 (11) <https://www.google.fr/intl/en/earth/> (21°06'17"S 175°08'26"W),
 (12) <http://climate.nasa.gov/images-of-change?id=565#565>, (13) <http://earthobservatory.nasa.gov/IOTD/view.php?id=86746>,
 (14) http://earthobservatory.nasa.gov/Features/WorldOfChange/columbia_glacier.php, (15) <http://climate.nasa.gov/images-of-change?id=535#535>

Figure 5.8: The 15 before-and-after image pairs used in the experiment.

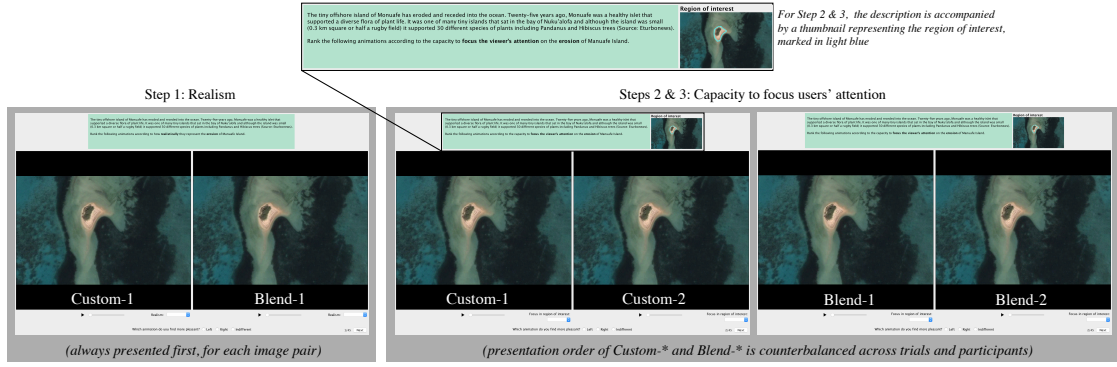


Figure 5.9: User interface shown two participants in each of the three steps of trial. Users were first asked to rate two animations that differ in their way of revealing the region of interest (RoI_{reveal}) according to their **Realism**, and then rate two animations that act on the RoI and the background either concurrently or in sequence (RoI_{anim}) according to their capacity to **Focus** users' attention. Custom-* and Blend-* labels were not shown to participants, and have been added here only to facilitate understanding of the experiment's design and interface.

5

animations, which one they find most pleasant. Then, they rate both animations according to a criterion that depends on the step they are completing, as described below.

Step 1 aims at testing $H_{realism}$ by comparing animations that reveal the RoI using a UNIFORM or PROGRESSIVE blending. In this step, participants are asked to read the textual description of the phenomenon to be illustrated in the animation, and to rate each animation according to how realistically they illustrate the phenomenon using five-point Likert scales (from 1: very poor, to 5: very good). To isolate the effect of animation factor RoI_{reveal} from RoI_{anim} , step 1 considers only single-stage animations ($RoI_{anim} = BG_CONCURRENT$).

The next two steps of each trial then test H_{focus} by comparing, for each RoI_{reveal} strategy, the case where the animation of the region of interest is separate from the animation of the background (BG_SEPARATE) with the case where the two animations are played concurrently (BG_CONCURRENT). In steps 2 and 3, the textual description is complemented by a thumbnail of the *before* image featuring a light-blue stroke that highlights the region of interest, so as to ensure that participants consistently identify this region. Participants are asked to carefully look at the region before rating each animation according to how well it focuses their attention on it, using a five-point Likert scale.

5.3.3. RESULTS

We used the Aligned Rank Transform procedure for non-parametric data [159] to analyze participants' ratings for both **Realism** and capacity to **Focus** attention. As Fig-

ure 5.10-a illustrates, the collected data support hypothesis $H_{realism}$. Participants found PROGRESSIVE animations, which gradually reveal the changes, more realistic than UNIFORM animations. The ANOVA test actually detects a significant effect of RoI_{reveal} on **Realism** ($F_{1,455} = 121.9$, $p < 0.0001$). The effect size between UNIFORM and PROGRESSIVE conditions is estimated as large (-1.007) using Cohen's d . *Change* does not have a significant effect on **Realism** ($p = 0.07$). Neither does the interaction between *Change* and RoI_{reveal} ($p = 0.2$).

Over 240 trials, participants preferred the PROGRESSIVE revealing strategy 172 times, and the UNIFORM strategy 47 times. In order to test whether PROGRESSIVE animations were preferred for specific types of geographic *Change*, we ran an ANOVA (using the Aligned Rank Transform procedure) to analyze the effect of *Change* on the **Progressive_Preferred** binary measure, which is computed as follows: 1 if PROGRESSIVE is more pleasant, 0 otherwise (*i.e.*, UNIFORM is the most pleasant or participant found them indifferently pleasant). We find that *Change* does not have a significant effect on **Progressive_Preferred** ($p = 0.33$), suggesting that participants tend to prefer PROGRESSIVE animations regardless of the type of *Change*.

In order to test H_{focus} , we analyzed the effect of RoI_{anim} on the animation's capacity to **Focus** participants' attention. In this case, the results do not support our hypothesis. RoI_{anim} has a significant effect on the rating ($F_{1,925} = 4.9$, $p = 0.025$). However, the analysis also reveals an interaction effect of $RoI_{anim} \times Change$ on **Focus** ($F_{4,925} = 4.13$, $p = 0.0025$). Looking at the details, we observe that image pair (6) is the only one where RoI_{anim} has a significant effect ($F_{1,47} = 40.4$, $p < 0.0001$), with BG_CONCURRENT being rated higher than BG_SEPARATE. It may be because, for this particular image pair (as well as for image pair (11)), we animate pixels in the background first; as opposed to all other image pairs, for which we animate pixels in the RoI first. We made this choice because the old contour of the modified entity (the lake from the *before* image of pair (6)) would have remained visible during the contraction process. However, participants commented that it had the effect of drawing their attention to areas that were not part of the RoI. For all other changes, as illustrated in Figure 5.11 BG_SEPARATE animations were rated slightly better than BG_CONCURRENT animations, but the difference is not significant.

Surprisingly, as Figure 5.10-b illustrates, RoI_{reveal} also has a significant effect on the capacity to **Focus** participants' attention ($F_{1,925} = 187.5$, $p < 0.0001$). The effect size between UNIFORM and PROGRESSIVE conditions is estimated as large (-0.88) using Cohen's d . Participants actually reported that movement in PROGRESSIVE animations was drawing their attention. This made them rate this kind of anima-

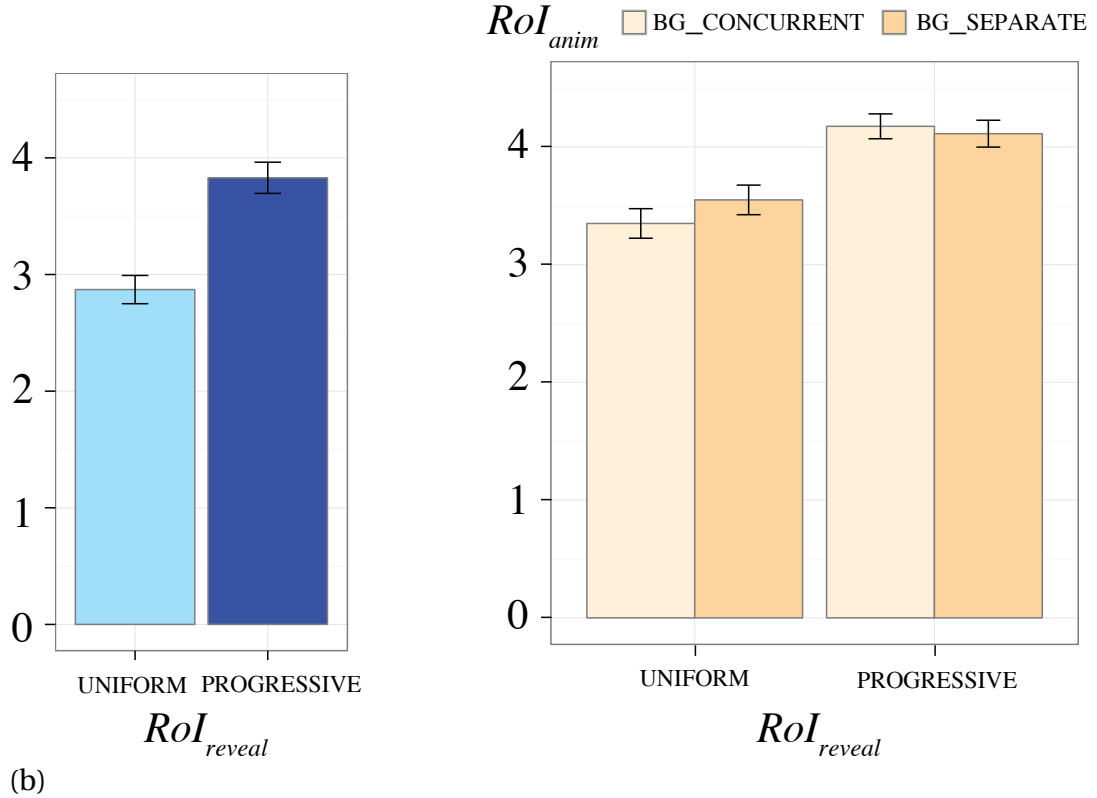


Figure 5.10: a) Average **Realism** rating per strategy for revealing the RoI. (b) Average **Focus** rating per RoI_{reveal} and RoI_{anim} . Error bars represent the 95% confidence intervals.

tion better at focusing their attention than UNIFORM animations. An ANOVA test also revealed an interaction effect of $RoI_{reveal} \times RoI_{anim}$ ($F_{4,925} = 13.7$, $p = 0.0002$), with BG_CONCURRENT being significantly better than BG_SEPARATE ($F_{1,455} = 17.5$, $p < 0.0001$) only for PROGRESSIVE trials. According to participants, in the PROGRESSIVE condition, animations that put changes in the RoI and changes in the background in sequence seemed unnecessarily slow for some image pairs in comparison with animations that concurrently change both regions. The perceived lack of changes during the animation of the background might have led users' attention to wander to other areas of the image.

Over 480 trials, participants preferred BG_CONCURRENT 122 times, and BG_SEPARATE 99 times. As in the case of **Realism**, we looked for an effect of *Change* on binary measures indicating preference, **Concurrent_Preferred** and **Separate_Preferred**. An ANOVA and multiple pairwise comparisons tests only revealed that **Concurrent_Preferred** was significantly higher for DEFORMATION trials in comparison with CONTRACTION and DISAPPEARANCE trials ($F_{4,60} = 3.67$, $p = 0.01$). This is also likely due to the ordering of stages in image pair (6) mentioned earlier.

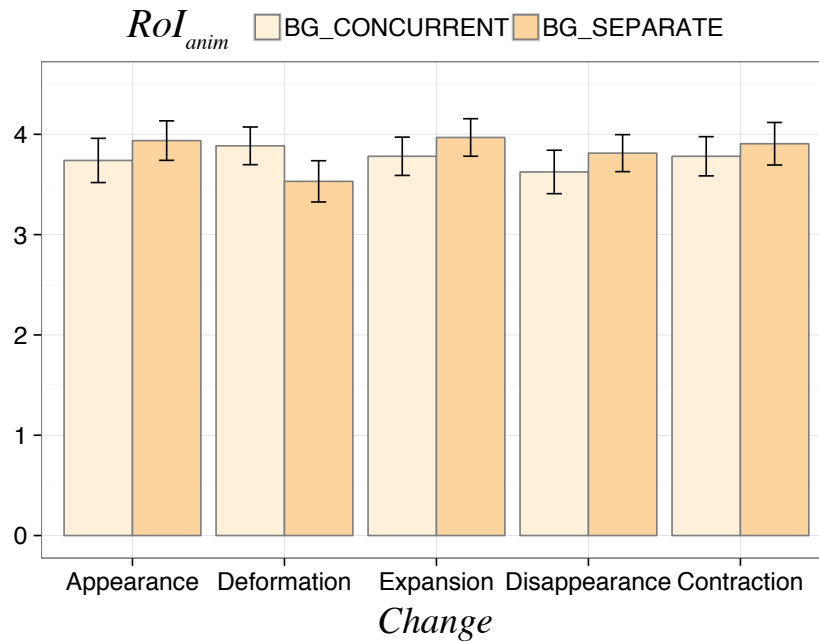


Figure 5.11: Average **Focus** rating per *Change* and *Stages*. Error bars represent the 95% confidence intervals.

5.3.4. DISCUSSION

Participants perceived PROGRESSIVE animations as more realistic than UNIFORM animations because they better convey the evolution between the images. They also found PROGRESSIVE animations more pleasant than UNIFORM animations. A participant acknowledged that even if PROGRESSIVE animations are noticeably computer generated, this type of animation would be good for teaching and demonstrating changes. These results support our initial hypothesis. However, image characteristics and the geographical processes depicted might influence the realism of the animation. For example, some participants mentioned that PROGRESSIVE animations work especially well for evolution of natural entities, such as rivers and lakes evolving, but are less effective in the case of changes caused by human activity, such as building construction.

We also hypothesized that animations that separate the RoI from the rest of the image would be effective at focusing viewers' attention. The analysis of participants' ratings cannot fully support this, and rather suggest that using PROGRESSIVE animations is more important than making use of staging for drawing viewers' attention to specific regions of interest. Nevertheless, several participants mentioned that staging was useful for animating changes in images whose backgrounds differed significantly. This was not the case when the region of interest was large and located at the center of the image, or when the region of interest and the background were highly

contrasted such as in, e.g., image pairs 2 and 9.

The findings from this experiment suggest that PROGRESSIVE animations yield more realistic results, and help viewers focus their attention on the region of interest. Designing PROGRESSIVE animations requires some flexibility. Different types of changes call for different animation strategies. This is the case even when considering only one specific type of change: no single strategy is always the most effective. Indeed, a wide range of image pairs fall in the same general change category, as the examples chosen for the experiment show (Figure 5.8). Collecting representative examples for each *Change* was actually not trivial, and we make our data publicly available to the community as a compilation of all image pairs and animation plans, for replication of this study or use in other studies. We also want to emphasize that, as in other experiments involving remote sensing imagery [?], we could not resort to unrealistic, synthetic data, as this would have threatened the external validity of our study.

5

5.4. AUTHORIZING TOOL

5.4.1. TOOL DESCRIPTION

The above empirical results indicate that animations that go beyond monolithic blending can indeed be effective at illustrating changes between before-and-after images. As discussed above, these results also indicate that there is a need for flexibility in how to achieve animated transitions depending on the nature of the change and the actual imagery. Animation plans provide a framework for expressing a broad variety of staged, locally-bounded blending transitions. However, authoring raw animation plans is a difficult task. It could even be argued that understanding some animation plans represented visually is not straightforward, at least not for a non-expert audience. To address this issue, the *Baia* framework features an authoring tool that enables users to create elaborate sequences of animation plans, hiding the raw plans (which are of little interest to most users) behind a user interface inspired by timeline-based video editing applications such as, e.g., Apple iMovie or the more advanced Adobe Premiere.

5.4.2. USER INTERFACE

Figure 5.12 shows a screenshot of this prototype authoring tool. The user interface features two main panels: one displaying the current before-and-after image pair

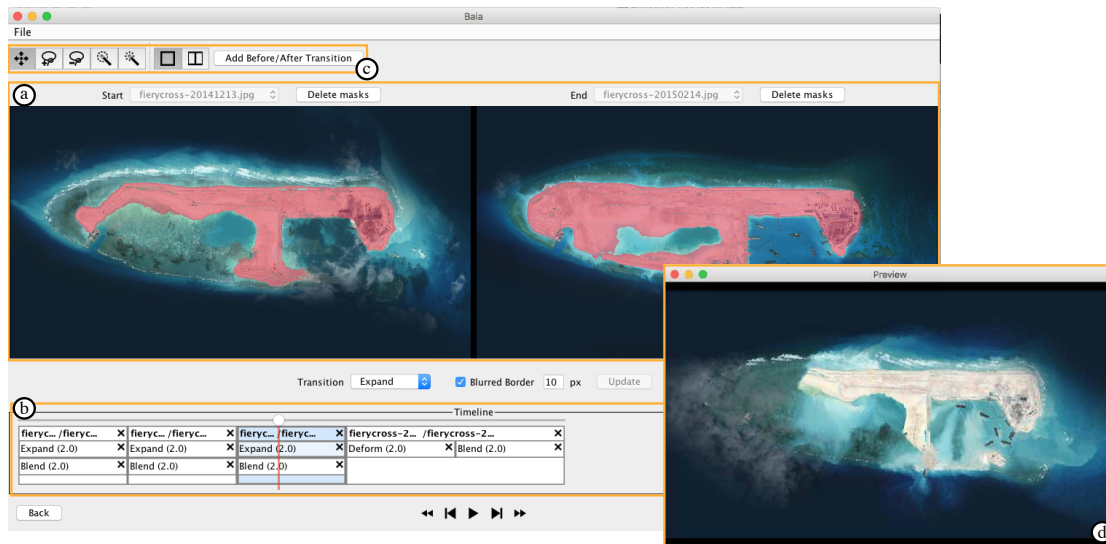


Figure 5.12: *Baia* authoring tool. (a) Current *before* (left) and *after* (right) image pair. (b) Editable timeline holding the sequence of image pairs and associated animation primitives. (c) Toolbar for mask creation and adjustment. (d) Animation preview window.

(Figure 5.12-a), the other displaying the timeline of animated transitions between image pairs (Figure 5.12-b). The former is used to edit the masks (M^s and M^e) that correspond to the locally-bounded regions to be blended using one of the available animation primitives. The latter is used to arrange individual transition clips into a sequence, similar to the tracks found in the above-mentioned timeline-based video editing software applications.

Masks are edited using tools familiar to image editing software users, such as lasso selection or magic wand (Figure 5.12-c). Selections can be transferred between both sides (before and after images), and refined iteratively and independently. This is particularly useful when, for example, creating a two-stage transition: one that blends the region of interest and another that blends the background. Users can create the mask for the region in one stage, copy it to the following stage, and then invert it to restrict the animation to the background. The magic wand tool can operate in either the HSV or CIELab color space, and will consider as input either all pixels in an image, or only contiguous pixels. Once the type of transition specified (*e.g.*, contraction, or directional progression), the animated transition clip can be generated and inserted in the timeline.

Baia can manage sequences that involve more than one image pair, where the *after* image of pair n is the *before* image of pair $n + 1$. For instance, the images used in Figure 5.12 come from a New York Times article [156] that features a total of 9 satellite images, *i.e.*, 8 before-and-after pairs. As an animation plan between the two images of a given before-and-after pair can be composed of more than one stage, the ed-

itable timeline is structured as a two-level hierarchy. The parent level contains the sequence of image pairs (top blocks in Figure 5.12-b labeled *fieryc...*). Each parent block can contain one or more child blocks (the last parent block contains a DEFORMATION block and a BLEND block).

Users can rearrange individual blocks, change their duration, and remove them. They also have the possibility to insert a color-transfer block, that smoothly modifies the *before* image's color histogram to better match that of the *after* image, before running the animation plan. This is useful to avoid distraction due to differences between the two images caused by, *e.g.*, different lighting conditions in the area at the time of data acquisition.

5

Beyond the set of predefined animation primitives discussed earlier, users can also import animation plans generated externally. As discussed earlier, the matrices of an animation plan are stored in the R and G bands of a TIFF file. Any such file can be imported and used as an animation plan, provided that it matches the image pairs' dimensions. This feature is targeted at more expert audiences, who can for instance derive an animation plan from external data, as illustrated in Figure 5.7.

The resulting animation sequence can be previewed (Figure 5.12-d) using classic VCR-like controls, or navigated frame-by-frame by direct manipulation of the red vertical bar indicating the current frame (Figure 5.12-b), in a manner similar to other video-editing software. Once satisfied with the result, sequences can be exported as videos or animated GIF images.

5.4.3. IMPLEMENTATION

Baia's authoring tool is implemented in Java and OpenGL using JOGL. We use OpenCV for image operations such as contour detection. Rendering of both images and animation previews is performed in OpenGL canvases, with animation plans interpreted by shaders to optimize frame rates.

There is one animation plan between each pair of images. Each animation plan is stored as a texture. The time to compute an animation plan depends on the image size and on characteristics of the regions to be animated. Morphological operators are applied iteratively, until the mask in the *before* image matches the mask in the *after* image. Shape morphing creates intermediate masks for the linear interpolation steps. In both cases, the time to compute animation plans thus depends on the difference in size between the corresponding regions in the two images. Computation time when using shape morphing also depends on the number of points on

Table 5.1: Animation plan computation time for representative DEFORMATION animations (apparatus described in Section 5.3.1).

Image pair in Fig. 5.8	Image Size	Iterations	Time (ms)
(5)	720×480	233	515
(2)	720×480	340	643
(9)	1981×1977	259	9,192
(6)	2000×2000	445	15,085

Table 5.2: Animation plan computation time for representative VECTOR MORPH animations (apparatus described in Section 5.3.1).

Image pair in Fig. 5.8	Contour points	Time (ms)
(11)	332	761
(14)	729	2,395

the contours, as calculating vertex correspondence can be time consuming when the contours are complex. As mentioned earlier, these textures get recomputed individually on a need-to basis, each time users press the Update button after having made modifications to the corresponding transition in the timeline. Tables 5.1 and 5.2 provide performance figures for representative examples of DEFORMATION and VECTOR MORPH.

5.4.4. USER STUDY

We conducted a user study to evaluate whether users are able to create elaborate animation plans using *Baia*, and to collect feedback to improve it. We created a tutorial⁵ as a web page that presents a series of short video clips explaining how to use the main editing features, and how to create masks and stages. The tutorial also includes a list of predefined animation primitives, along with illustrative thumbnails as animated GIF, and short video clips explaining how to use them. With the help of this tutorial, participants had to perform seven tasks where they were presented with target animations that we had created, and that they had to reproduce. Participants ended the experiment with one final, more creative task, where they were free to design the animation of their choice to illustrate the transition between a given before-and-after image pair.

PARTICIPANTS AND APPARATUS

Six unpaid volunteers (two females), daily computer users, age 24 to 31 (average: 25.3, median 24.5), served in the experiment. All had normal or corrected-to-normal vision and did not suffer from color blindness. We used the same apparatus as in the

⁵The tutorial can be accessed at <http://ilda.saclay.inria.fr/mapmuxing/baia/authoringtool/instructions.html>

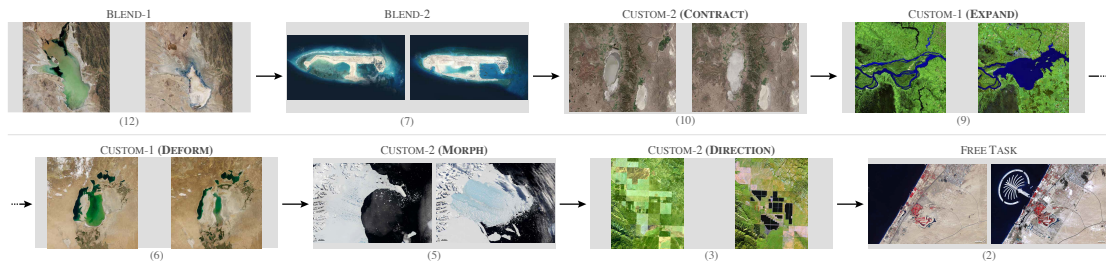


Figure 5.13: The eight tasks in the user study: image pairs, animation strategies and primitives for animating the region of interest.

previous study. Four of them had participated in the first study.

5

PROCEDURE

Participants began the experiment with a training phase. They had access to the *Baia* authoring tool, and to the tutorial, which was open in a separate window. They were instructed to go through the tutorial, and were invited to test any feature in *Baia*'s authoring tool with one of the before-and-after image pairs that are used in the tutorial. Participants could stay in the training phase as long as they wanted, but they were also told that they would be free to get back to the tutorial at any moment during the study. All participants watched all the videos related to *Baia*'s features. They spent 30 minutes on average in the tutorial phase.

During the test phase that followed, a third window was opened to display instructions. For the first seven tasks, the instructions simply consisted of a video clip of the animation to reproduce. Because staging might have a subtle effect, instructions also included a tip indicating whether the animation featured more than one stage, in order to encourage participants to use stages, and observe how successful they were at it. Figure 5.13 shows the sequence of the 7+1 animations participants had to create. These animations were always presented in the same order, as their complexity progressively increased, starting with a single-stage monolithic blend, and ending with a 5-stage animation that featured varying directions. This set of animations uses all primitives but RADIAL, and involves different staging and ordering strategies.

Finally, participants were asked to create a custom animation to illustrate the construction of an island in Dubai (image pair (2) in Figure 5.8). Participants could look at the tutorial whenever they wanted to. The experimenter explicitly encouraged them to do so when they seemed to struggle with a specific functionality. Participants were also encouraged to give feedback about the system's usability and the task's difficulty at any time during the experiment.

RESULTS

All participants completed the 7 reproduction tasks, taking 7.8 minutes on average per animation (Median=6.9, std dev=4.9). Unsurprisingly, complex animations were longer to reproduce than easy ones. Animations of pairs (5) and (3) were particularly challenging. The former probably because of the higher complexity of the VECTOR MORPH primitive, and the latter probably because it involved five different stages.

Even though they were not given any indication about which animation primitives to use, participants managed to either choose the ones that we had originally used to create the target animations, or to use other primitives that yielded similar results in most cases. All participants chose the right primitives for animating image pairs (12), (10), (9) and (3). For image pair (7), only one participant did not use the BLEND primitive for the region of interest, rather choosing the CONTRACTION primitive. As he created the same mask in both canvases, the resulting animation effect was equivalent to that of the target animation. For image pair (6), that relies on the DEFORMATION primitive, four participants used other primitives: three participants used EXPANSION and CONTRACTION in parallel, and one participant only used CONTRACTION. As the DEFORMATION primitive is actually implemented as a combination of EXPANSION and CONTRACTION, using them in parallel results in the same visual effect. Participants had more trouble with image pair (5), which uses the VECTOR MORPH primitive. Half of them tried to use it, but only one managed to do so successfully. As described in Section 5.2, animations generated by the VECTOR MORPH primitive are highly dependent on the algorithm that extracts the contour of the two masks, and that matches the points between the two contours. As users create masks through a color-based selection that depends on the pixel used as a reference, creating a mask that is similar to the one used in the target animation was difficult. After several unsatisfying attempts, two participants gave up using this primitive, preferring the more predictable EXPANSION and DEFORMATION primitives. The other three participants did not try to use VECTOR MORPH, and decided to use either EXPANSION (two participants) or RADIAL (one participant).

Participants were free to look at the tutorial at will, but did so only a limited number of times. On average, participants referred three times to the tutorial during the test phase. The most watched clips were: *“How to Add Masks”*, *“How to Add Stages”*, and *“How to create a Morph Animation”* (each one three times). Otherwise, they referred to the tutorial mostly to identify the right primitive based on the thumbnails in the list of primitives, and to watch the associated video that details the steps to follow.

In order to collect an unbiased measure of the resemblance of participants' animations to the target animations, we asked two external evaluators to watch the target animation, and all animations that our participants had created. Evaluators had to assign a score between 1 and 10 to each animation according to its similarity to the target animation. The mean score was 8.6 (Median= 9, $s=1.74$). Figure 5.14 reports these scores, showing that all designed animations received high scores, with the exception of animations for image pair (5). This probably reflects the difficulties that participants encountered when using the VECTOR MORPH primitive.

Regarding the open task at the end of the experiment, all participants created at least two stages in parallel, isolating the animation of the island from the rest of the image. Beyond that, all participants created different animations using various primitives and strategies to progressively reveal the island. Some of them created quite elaborate animation plans. For example, one participant used four different stages involving the directional progression primitive to convey a progression that made sense in terms of construction work. Interestingly, two participants used the DEFORMATION primitive to make the island appear: even if the island was not on the before image, they selected a small region on the border of the area covered by the island in the before canvas to create the start mask, and used DEFORMATION with an end mask delimiting the island in the after canvas.

FEEDBACK AND POSSIBLE IMPROVEMENTS

Overall, participants' qualitative feedback was positive. They found the system easy enough to use, with a smooth learning curve. One participant spontaneously reported that *Baia* enabled him to quickly produce the target animations, and that the study was entertaining. Another participant particularly appreciated the support for exploring different design alternatives that the tool offers thanks to the possibility to test different animation primitives for a given stage.

However, participants also faced a few difficulties. Two participants mentioned that understanding the effect of a primitive can be difficult in some cases. Our interpretation is that some problems are due to participants not clearly understanding that the system does not have any semantic knowledge about the geographic data. For example, in some cases, participants tried to use the CONTRACTION or the EXPANSION primitive with the same mask in both before and after images, apparently failing to understand that the system cannot automatically infer the evolution of the masked entity. We plan to add more feedback to prevent such errors with, e.g., warning messages in case mask sizes are not consistent with the type of primitive selected. Another source of confusion comes from the lack of visibility of the automatic pairing

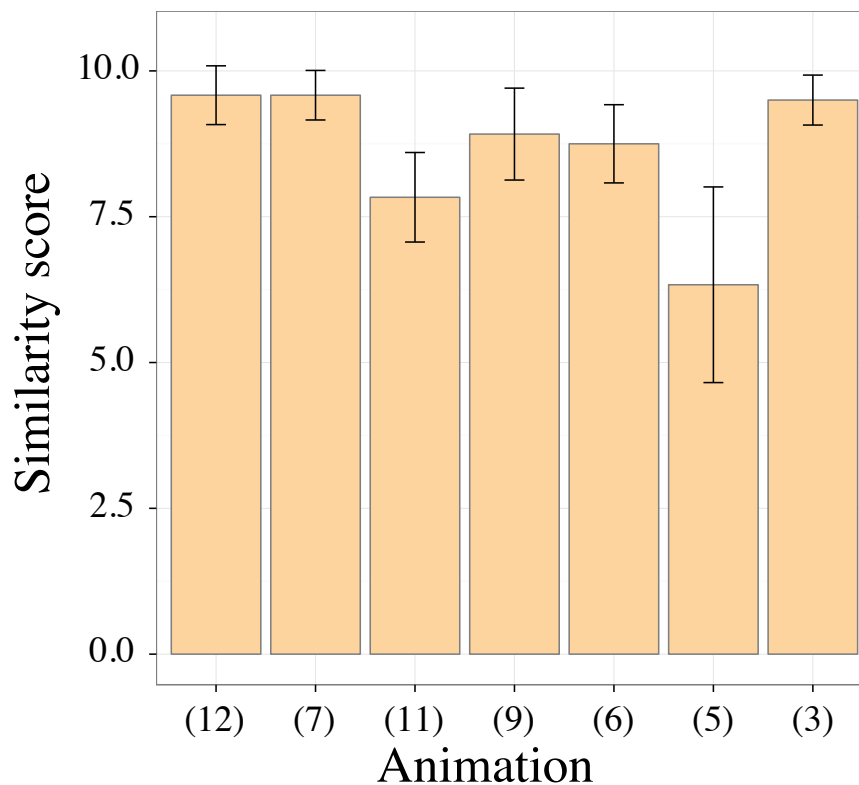


Figure 5.14: Scores given by the two raters per animation.

between contour points when using the VECTOR MORPH primitive. We plan to work on making this pairing visible, and on giving users the possibility to manually fix it in order to achieve the effect that they have in mind.

Some of *Baia*'s pixel-based image processing operations are resource-intensive and cannot be performed on-the-fly, limiting the user interface's capability to maintain a responsive dialogue between the tool and the user. The authoring tool thus features an Update button that users must press to apply changes. In some cases, participants forgot to press it after having made changes to masks, generating some frustration as modifications could get lost if another stage was selected. The existing feedback, which consists of changing the color of this button and adding a colored frame to the preview window to indicate that there are uncommitted changes, seems to be insufficient. We should, at the minimum, pop-up a warning message whenever a stage with uncommitted changes gets unselected. Finally, participants also mentioned that using drag-and-drop actions to change the temporal arrangement of stages would be preferable to the current interaction that requires users to select the stages to put in parallel, invoke a contextual menu, and select the Parallelize stages command.

5.5. DISCUSSION

Baia makes it possible to create advanced animated transitions between sequences of before-and-after satellite images. For this, it relies on *animation plans*, a pixel-based model that gives animation authors significant flexibility in how they transition between images. Compared to monolithic blending, this flexibility enables them to create transitions that have the potential to better focus the audience's attention on specific changes, and that can better approximate the dynamics of the actual changes, as was confirmed by a user study that involved a dataset of 15 real-world before-and-after images. The approach, however, has limitations.

5

First, no matter how elaborate, these animations will in most cases remain *approximations* of the phenomenon they are illustrating, as already mentioned. This approximation will be more or less obvious depending on the considered images, nature of the phenomenon, and data available to derive the animation plan (if any). On one hand, the animation plan derived from the digital elevation model in Figure 5.5 is very close to reality; but on the other hand, the animation for image pair (5) in Figure 5.8 is a rough approximation of what actually happened. While this potential lack of realism is indeed a limitation, being able to create such approximations is actually the very purpose of *Baia*. If enough intermediate frames have been captured *via* remote sensing, or if such frames can be generated via simulations based on relevant physics models, there is no need for authoring transitions. But remote sensing data is typically captured at a low frequency, and running computer-intensive simulations is not always possible or cost-effective. *Baia* offers a trade-off, enabling a higher degree of realism than monolithic blending at a very low cost. A potential danger in some cases might be that animations are perceived as realistic when they are not actually reflecting reality, possibly misleading the audience. But ultimately, it is the decision and responsibility of animation authors, depending on the context in which they are making their presentation.

Another limitation of animation plans is that they provide poor support for pixel displacement, meaning that it is difficult to depict moving entities. Visual comparison techniques based on the explicit encoding of changes [47] are more effective in such cases. Explicit encoding is also a good means to focus viewers' attention. While such techniques are not included in *Baia* at the moment, they are completely orthogonal to, and fully compatible with, our approach as explicit encodings can simply be superimposed on top of *Baia* renderings. We chose to focus our efforts on investigating the specific capabilities of *Baia*'s model, but the integration of explicit encoding techniques and visual highlighting methods [?] in the tool would be straightforward,

providing authors with complementary means to increase the saliency of particular regions during transitions. Future work could investigate the articulation of both approaches.

6

CONCLUSION

As seen in the previous chapters, multiple representations of the same geographical region exist. These representations vary in the nature of the data they contain (object or field), the information they hold (*e.g.*, roads, buildings, digital elevation model), and the ways of representing this information (*i.e.*, the chosen symbolization). The interviews reported in Chapter 3 show that GIS experts often need to relate and compare different geographic layers. Non expert users need to work with multiple layers too, in casual tasks like planning a hike, or exploring the evolution of some familiar location. Existing techniques to combine multiple layers, however, remain limited. They are often limited to toggling between layers, swiping between them and sometimes drilling through them. They do not take advantage of the information semantics to create more interactive transitions to better support users in their tasks.

This thesis addresses the problem of relating different geographical representations of the same region at the same scale through interactive transitions. Transitions can be based on spatial multiplexing (*i.e.*, combining different representations spatially through map composition) or on time multiplexing (*i.e.*, sequencing the representations using smooth animations).

We present three contributions. First, we describe a characterization and an evaluation of interactive map comparison techniques based on spatial multiplexing. Then, we introduce *MapMosaic*, a novel approach for spatial multiplexing. Finally, we present *Baia* a novel approach for time multiplexing to better convey changes between before-and-after satellite images. We give a short summary of each contribution below, and then open directions for future work.

6.1. SUMMARY

6.1.1. CHARACTERIZATION AND EVALUATION OF EXISTING MAP COMPARISON TECHNIQUES

In the third chapter, we identify and evaluate five different techniques that combine layers through spatial multiplexing, based on superimposition or juxtaposition. We characterize these techniques according to user attention, visual interference and the scanning strategy they require (vision-driven or motor-driven). For example, *Translucent Overlay* is a technique that superimposes two representations and enables users to vary the translucency of the top layer. It suffers from a high level of visual interference but it has a low level of divided attention as both representations are in the same space, and requires a scanning that relies mostly on visual search (vision-driven). We also include two other techniques that superimpose the layers: *Blending Lens* that reveals the underlying region in a small area around the cursor and *Swipe* that uses a drawer metaphor to enable users to swipe the top layer to reveal the bottom one. These two techniques rely more on a motor-driven scanning strategy, as they require users to use the pointing device to reposition elements on the screen (the lens or the drawer handle). *Juxtaposition*, a technique that displays both maps side by side, presents the highest level of divided attention but does not suffer from visual interference. We designed the fifth technique, *Offset Lens*, as a compromise between visual interference and divided attention. This technique, as *Blending Lens*, reveals the second representation in a small viewport, but this time is juxtaposed to the source region.

We report on a user study that compares these techniques. Users were asked to find differences between a topographic map and satellite imagery using each one of the five techniques. Results suggest that superimposition works better than juxtaposition. *Translucent Overlay* was the most efficient technique in general, followed by *Blending Lens* that performed even better than *Translucent Overlay* in the tasks that required participants to inspect objects that are visible in the top layer. These results suggest that structuring the search through a motor-driven scanning strategy might facilitate layer comparison for some tasks.

6.1.2. *MapMosaic*: SPATIAL MULTIPLEXING FOR VECTOR AND RASTER GEOGRAPHICAL LAYERS

The techniques considered in the above-mentioned study juxtapose or superimpose two maps in a predefined-shape (a circular lens or the whole viewport). We had the intuition that this approach might not be flexible enough for some tasks. To better understand when and how map experts resort to spatial multiplexing, we conducted a series of interviews with GIS operators and researchers. These interviews revealed four main design requirements to work with multiple layers at a time: limit switches between representations, provide support for local compositing, filter out objects according to their attributes and keep track of the exploration. Taking these design requirements as the starting point, we designed *MapMosaic*: a dynamic compositing model to explore multi-layer representations. *MapMosaic* takes advantage of the semantics of the information contained in the layers. It is based on what we call *compositing regions*, first-class, interactive objects that enable users to combine multiple layers according to their objects and attributes.

We evaluated the *MapMosaic* model in two different ways. First, we used the “cognitive-dimensions” framework to compare *MapMosaic* to two existing solutions: QGIS, a widely-used desktop GIS and MAPublisher, a professional cartographic editor. This evaluation suggests that *MapMosaic* dynamic compositing model is less cognitively demanding than the existing models, which do not directly support on-demand compositing. The second evaluation reports on feedback gathered from workshops with GIS experts. The participants responded positively to the prototype and suggested several detailed scenarios where *MapMosaic* could be useful such as categorizing forest areas, visualizing spatial queries or registering satellite imagery coming from different sources.

6.1.3. *Baia*: TIME MULTIPLEXING FOR BEFORE-AND-AFTER SATELLITE IMAGES

The dynamic compositing model of *MapMosaic* facilitates several tasks that require comparing and relating multiple representations of the same geographical region. However, using spatial multiplexing to present spatiotemporal information may not be the best approach, as it does not convey dynamics. In Chapter 5, we introduce *Baia*: a novel animation framework to better present changes between before-and-after satellite images. This model aims at helping viewers focus on a region of interest and conveying changes’ dynamics. For this, the model defines animations plans,

based on a start, and an end time for the animation per pixel. Animation plans allow users to create *custom* animations by letting them specify at the pixel level when each area should be animated. To facilitate the creation of such animations, we introduce a set of animation primitives to represent common geographical changes [28]. We also present an authoring tool that enables users to use those primitives. Our tool also offers usual editing features and lets users import their data if they want to create more advanced animations.

We evaluated the animations created using *Baia* by comparing them to monolithic blending (*e.g.*, a homogeneous blend from the before image to the end image). We considered both the way in which we animate the region of interest (blending or *custom*) and the staging strategy. In our experiment, we asked participants to rate the animations according to their realism and their capacity to focus viewer's attention on the region of interest. The results suggest that the animations created with *Baia* were both perceived as more realistic and better at focusing viewer's attention than blending. We also evaluated the authoring tool to test if users could create animations and to collect feedback. Participants were able to reproduce animations after a short tutorial, and we collected several possible improvements.

6

6.2. PERSPECTIVES AND FUTURE WORK

6.2.1. FURTHER EVALUATIONS

As it would be very difficult to generate synthetic data that features the same complexity than real maps, we chose to use real maps and images for the user studies conducted in Chapters 3 and 5. This approach ensures a high external validity but it also introduces biases, as we had to select the data we used, and our results cannot be generalized to all kinds of maps and images. The first perspective of future work is replicating our studies with different data, as detailed next.

In the evaluation presented in Chapter 3, users were asked to find six kinds of differences between an artificially-modified topographic map and its corresponding aerial imagery. To complete the tasks, users had to identify and compare two kinds of objects, buildings and roads. We chose predominantly maps of rural areas. The results of the study would probably be different if we had used maps that represent dense urban areas, as participants would have to inspect more objects and the buildings and roads would be harder to identify. Also, we hypothesize that comparing similar maps (*i.e.*, two satellite images, or two topographic maps that have different information but similar symbolization) would also impact the results, as superimpo-

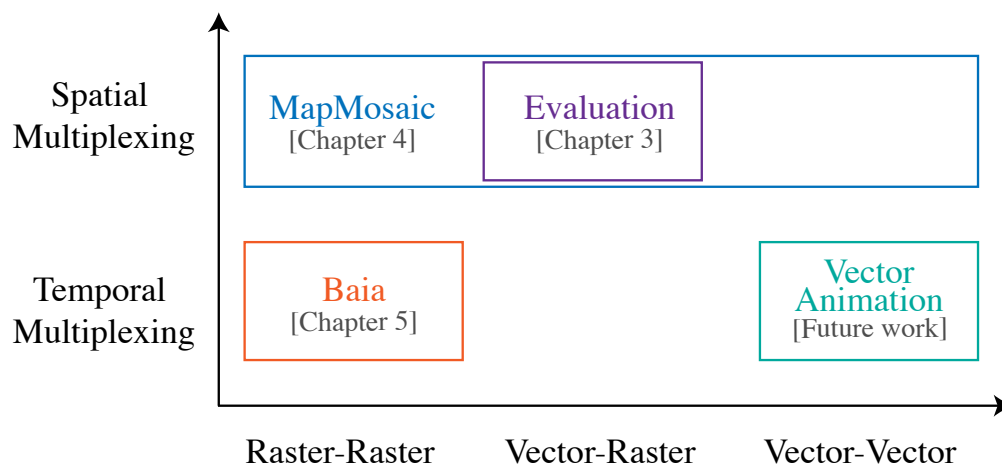


Figure 6.1: Classification of the contributions of the thesis according to the type of multiplexing and the kind of geographical data they handle.

sition techniques might not work as well, considering that it would be more difficult to identify which layer the objects belong to. We would thus like to replicate the study presented in Chapter 3, but taking into account the region type (rural vs urban) and the category to which the layers belong to (topographic vs realistic) and whether the comparison involves layers from the same category or not.

In Chapter 5, we collected real pairs of before-and-after satellite images that display geographical changes (*e.g.*, landscape and human construction evolution) to evaluate the animations created with *Baia*. For each image pair, we identified a region of interest (where users should focus their attention) and a background region. We designed the animations to focus viewers' attention on the region of interest. In most of the images that we used, this region was at the center of the image and featured a high color contrast with the background. We believe that participant's attention tend to naturally focus on the image center and that they easily delineate contours when color contrasts are high. We thus hypothesize that the results of our study would have been different with different pairs of images. In particular, we expect staging to have a stronger impact to focus viewers' attention in more subtle cases. To test this hypothesis, we would like to replicate the study with more images, taking into account the position, the size, and the color contrast of the region of interest with the background.

6.2.2. TEMPORAL MULTIPLEXING FOR VECTOR OBJECTS

The pixel-based model *Baia* relies on enable the creation of transitions that have the potential to better focus the audience's attention on specific changes, and that can better approximate the dynamics of the actual changes. It enables users to create animation with a limited amount of data, typically a couple of images. However, it has some limitations. For example, it is not possible to represent objects' movement, as we only vary each pixel's opacity without changing its position. Also, as we only work with pixel information, we do not have any information about objects' identity, *i.e.*, we cannot track the evolution of an object. For example, we cannot track a lake across several images, because we do not have the vector information that would associate pixels to the lake object. We would like to extend our model to be able to use vector shapes to guide the temporal multiplexing. Having such a model would allow us to explore other uses of animation, such as exploring geographical evolution besides presenting it. Exploratory tasks require not only to see the changes of one object but also to be able to compare entities' behaviors, and to visualize series of different events [12]. We present two directions of future work where animations would support such tasks: using spatial multiplexing to compare animations, and using complementary abstract representations to compare objects' behaviors. Figure 6.1 shows how temporal multiplexing using vector data would further explore the design space of map multiplexing.

Baia enables users to easily create animations that represent common geographic phenomena. The model and the authoring tool work well for presenting information, but, for now, cannot be easily used to explore the data, as users have to define the regions and the primitives used in the animations. Complementing our model with vector information (when available) would enable us to potentially create the transitions automatically. We could use vector databases that contain time-based vector definitions of a geographical object, such as a lake contour at different moments. Animations could be used then to not only display the changes but also to draw attention to some of them, by using staging or by magnifying or highlighting small changes that would be difficult to see in the animation otherwise. We would also like to design interaction techniques to combine spatial and temporal multiplexing. For example, we could enable users to restrict the animation to specific geographical regions, such as the compositing regions featured in *MapMosaic*. This technique would help to focus users attention. Also, it would enable comparison of different regions across time, in order to search for similar patterns, as users could define the starting and ending point, and the speed of the animation. For example, these techniques could help to explore similar urbanization processes that happened in different places, at

different times.

Besides the processes featured in Baia's model, Claramunt's taxonomy [28] also includes processes that involve several entities: functional relationships between entities, and evolution of spatial structures involving several entities. These processes might be particularly interesting for historical geographic databases, for example, to study countries' development (*e.g.*, countries that split or merge). We would like to investigate if we can design visualization techniques that effectively represent this kind of processes, as they do not necessarily include geometric changes like the ones explored with *Baia*. For example, in the case of a country splitting in two, we could imagine an animation that emphasizes the newly created border. However, animations present only one moment of an object's evolution at a time, so it's difficult to have a more global view of an object's evolution. Therefore, we would like to explore if we can complement temporal multiplexing by other more abstract representations, for example, spatiotemporal graphs [119]. This representation explicitly displays the relationships and the different processes the entities are involved in as a graph. It thus facilitates comparing the behaviors of multiple entities, by comparing the graphs that represent their evolution, but lacks the representation of the evolution itself (*e.g.*, it does neither show the geographical position nor the shape of entities). It could be thus interesting to use it combined with an animated view. For example, the graph could serve as a timeline to navigate the animated evolutions of the different entities.

REFERENCES

- [1] J. Alvina, C. Appert, O. Chapuis, and E. Pietriga. RouteLens: Easy Route Following for Map Applications. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pages 125–128. ACM, 2014.
- [2] G. L. Andrienko and N. V. Andrienko. Interactive maps for visual data exploration. *International Journal of Geographical Information Science*, 13(4):355–374, 1999.
- [3] N. Andrienko, G. Andrienko, and P. Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, 14(6):503 – 541, 2003. Visual Data Mining.
- [4] C. Appert, O. Chapuis, and E. Pietriga. High-precision magnification lenses. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 273–282, New York, NY, USA, 2010. ACM.
- [5] B. Bach, E. Pietriga, and J.-D. Fekete. Graphdiaries: Animated transitions andtemporal navigation for dynamic networks. *IEEE Trans. on Visualization and Computer Graphics*, 20(5):740–754, 2014.
- [6] A. Ballatore and M. Bertolotto. Personalizing maps. *Communications of the ACM*, 58(12):68–74, Nov. 2015.
- [7] S. Battersby and K. Goldsberry. Considerations in design of transition behaviors for dynamic thematic maps. *Cartographic Perspectives*, 0(65), 2010.
- [8] P. Baudisch and R. Rosenholtz. Halo: A technique for visualizing off-screen objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 481–488, New York, NY, USA, 2003. ACM.
- [9] M. Beaudouin-Lafon and W. E. Mackay. Reification, polymorphism and reuse: Three principles for designing visual interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00, pages 102–109, New York, NY, USA, 2000. ACM.
- [10] E. A. Bier, M. C. Stone, K. Pier, W. Buxton, and T. D. DeRose. Toolglass and Magic Lenses: The See-through Interface. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80. ACM, 1993.
- [11] A. Blackwell and T. R. G. Green. Notational systems—the cognitive dimensions of notations framework. In J. M. Carroll, editor, *HCI Models, Theories, and*

- Frameworks: Toward an Interdisciplinary Science*, chapter 5, pages 103–134. Morgan Kaufmann, 2003.
- [12] C. Blok, B. Köbben, T. Cheng, and A. A. Kuterema. Visualization of relationships between spatial patterns in time by cartographic animation. *Cartography and Geographic Information Science*, 26(2):139–151, 1999.
- [13] A. Boér, A. Çöltekin, and K. C. Clarke. An evaluation of web-based geovisualizations for different levels of abstraction and realism—what do users predict? In *International Cartographic Conference*, pages 209–220, 2013.
- [14] L. Bonanni, X. Xiao, M. Hockenberry, P. Subramani, H. Ishii, M. Seracini, and J. Schulze. Wetpaint: Scraping through multi-layered images. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 571–574. ACM, 2009.
- [15] J. Böttger, U. Brandes, O. Deussen, and H. Ziezold. Map warping for the annotation of metro maps. *IEEE Computer Graphics & Applications*, 28(5):56–65, 2008.
- [16] F. Bourgeois and Y. Guiard. Multiscale pointing: facilitating pan-zoom coordination. In *CHI’02 Extended Abstracts on Human Factors in Computing Systems*, pages 758–759. ACM, 2002.
- [17] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, 2013.
- [18] J. Brosz, M. A. Nacenta, R. Pusch, S. Carpendale, and C. Hurter. Transmogrification: Causal manipulation of visualizations. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST ’13, pages 97–106. ACM, 2013.
- [19] S. Caquard. Cartography i: Mapping narrative cartography. *Progress in Human Geography*, 37(1):135–144, 2011.
- [20] S. Caquard and W. Cartwright. Narrative cartography: From mapping stories to the narrative of maps and mapping. *The Cartographic Journal*, 51(2):101–106, 2014.
- [21] M. S. T. Carpendale and C. Montagnese. A framework for unifying presentation space. In *Proceedings of the 14th Annual ACM Symposium on User Inter-*

- face Software and Technology*, UIST '01, pages 61–70, New York, NY, USA, 2001. ACM.
- [22] S. Carpendale. Information visualization. chapter Evaluating Information Visualizations, pages 19–45. Springer-Verlag, Berlin, Heidelberg, 2008.
- [23] N. Casey and J. Haner. Climate change claims a lake, and an identity. *The New York Times*, July 2016.
- [24] R. Chang, G. Wessel, R. Kosara, E. Sauda, and W. Ribarsky. Legible cities: Focus-dependent multi-resolution visualization of urban relationships. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1169–1175, 2007.
- [25] F. Chevalier, P. Dragicevic, and S. Franconeri. The not-so-staggering effect of staggered animated transitions on visual tracking. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):2241–2250, 2014.
- [26] F. Chevalier, N. H. Riche, C. Plaisant, A. Chalbi, and C. Hurter. Animations 25 years later: New roles and opportunities. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, pages 280–287. ACM, 2016.
- [27] S. Christophe. Cartographic styles between traditional and original (towards a cartographic style model). In *AutoCarto Conference*, September 2012.
- [28] C. Claramunt and M. Thériault. Managing time in gis an event-oriented approach. In *Recent Advances in Temporal Databases*, pages 23–42. Springer, 1995.
- [29] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys*, 41(1):2:1–2:31, Jan. 2009.
- [30] A. Cockburn and J. Savage. Comparing speed-dependent automatic zooming with traditional scroll, pan and zoom methods. In *People and Computers XVI-IóDesigning for Society*, pages 87–102. Springer, 2004.
- [31] A. Coltekin, I.-E. Lokka, and A. Boér. The utilization of publicly available map types by non-experts-a choice experiment. In *Proceedings of the 27th International Cartographic Conference (ICC2015)*, Rio de Janeiro, Brazil, pages 23–28, 2015.

- [32] H. Couclelis. *People manipulate objects (but cultivate fields): Beyond the raster-vector debate in GIS*, pages 65–77. Springer Berlin Heidelberg, Berlin, Heidelberg, 1992.
- [33] T. J. Cova. Gis in emergency management.
- [34] J. W. Crampton. Interactivity types in geographic visualization. *Cartography and Geographic Information Science*, 29(2):85–98, 2002.
- [35] I. F. Cruz, V. R. Ganesh, C. Caletti, and P. Reddy. GIVA: A Semantic Framework for Geospatial and Temporal Data Integration, Visualization, and Analytics. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 544–547. ACM, 2013.
- [36] W. Dong, J. Ran, and J. Wang. Effectiveness and efficiency of map symbols for dynamic geographic information visualization. *Cartography and Geographic Information Science*, 39(2):98–106, 2012.
- [37] J. Dykes, A. M. MacEachren, and M.-J. Kraak. *Exploring geovisualization*. Elsevier, 2005.
- [38] J. A. Dykes. Exploring spatial data representation with dynamic graphics. *Computers & Geosciences*, 23(4):345 – 370, 1997. Exploratory Cartographic Visualisation.
- [39] R. Eccles, T. Kapler, R. Harper, and W. Wright. Stories in geotime. *Information Visualization*, 7(1):3–17, 2008.
- [40] R. Edsall, G. Andrienko, N. Andrienko, and B. Buttenfield. Interactive maps for exploring spatial data. *ASPRS Manual of GIS*, 2008.
- [41] M. Elias, J. Elson, D. Fisher, and J. Howell. Do I Live in a Flood Basin?: Synthesizing Ten Thousand Maps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 255–264, New York, NY, USA, 2008. ACM.
- [42] N. Elmqvist, Y. Riche, N. Henry-Riche, and J.-D. Fekete. Mélange: Space folding for visual exploration. *IEEE TVCG*, 16(3):468–483, May 2010.
- [43] ESRI. Telling stories with maps: a white paper, February 2012. [Online; accessed 07-September-2016].

- [44] T. Fechner, D. Wilhelm, and C. Kray. Ethermap: Real-time collaborative map editing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3583–3592. ACM, 2015.
- [45] N. Feldt, H. Pettersson, J. Johansson, and M. Jern. Tailor-made exploratory visualization for statistics sweden. In *Coordinated and Multiple Views in Exploratory Visualization, 2005.(CMV 2005). Proceedings. Third International Conference on*, pages 133–142. IEEE, 2005.
- [46] S. Fuhrmann, P. Ahonen-Rainio, R. Edsall, O. S. Fabrikant, E. L. Koua, C. Tolon, C. Ware, and S. Wilson. Making useful and useable geovisualization: Design and evaluation issues. In *Exploring Geovisualization*, pages 541–554. Elsevier, New York, NY, USA, 2005.
- [47] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, Oct. 2011.
- [48] K. Goldsberry and S. Battersby. Issues of change detection in animated choropleth maps. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 44(3):201–215, 2009.
- [49] M. F. Goodchild. Geographical information science. *International Journal of Geographical Information Systems*, 6(1):31–45, 1992.
- [50] M. F. Goodchild, M. Yuan, and T. J. Cova. Towards a general theory of geographic representation in gis. *International Journal of Geographical Information Science*, 21(3):239–260, 2007.
- [51] T. R. G. Green and M. Petre. Usability analysis of visual programming environments: a “cognitive dimensions” framework. *Journal of Visual Languages & Computing*, 7(2):131–174, 1996.
- [52] A. L. Griffin, A. M. MacEachren, F. Hardisty, E. Steiner, and B. Li. A comparison of animated maps with static small-multiple maps for visually identifying space-time clusters. *Annals of the Association of American Geographers*, 96(4):740–753, 2006.
- [53] A. L. Griffin and A. C. Robinson. Comparing color and leader line highlighting strategies in coordinated view geovisualizations. *IEEE transactions on visualization and computer graphics*, 21(3):339–349, 2015.

- [54] A. L. Griffin, T. White, C. Fish, B. Tomio, H. Huang, C. R. Sluter, J. V. M. Bravo, S. I. Fabrikant, S. Bleisch, M. Yamada, and P. Picanço. Designing across map use contexts: a research agenda. *International Journal of Cartography*, 0(0):1–25, 0.
- [55] E. Grosso, J. Perret, and M. Brasebin. Geoxygene: an Interoperable Platform for Geographical Application Development. In *Innovative Software Development in GIS*, chapter 3, pages 67–90. John Wiley & Sons, Inc., 2013.
- [56] S. Gustafson, P. Baudisch, C. Gutwin, and P. Irani. Wedge: Clutter-free visualization of off-screen locations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 787–796, New York, NY, USA, 2008. ACM.
- [57] B. L. Harrison, H. Ishii, K. J. Vicente, and W. A. S. Buxton. Transparent layered user interfaces: An evaluation of a display design to enhance focused and divided attention. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pages 317–324, New York, NY, USA, 1995. ACM Press/Addison-Wesley.
- [58] M. Harrower. Visualizing change: using cartographic animation to explore remotely-sensed data. *Cartographic Perspectives*, (39):30–42, 2001.
- [59] M. Harrower. Tips for designing effective animated maps. *Cartographic Perspectives*, 0(44), 2003.
- [60] M. Harrower. A look at the history and future of animated maps. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 39(3):33–42, 2004.
- [61] M. Harrower. The cognitive limits of animated maps. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 42(4):349–357, 2007.
- [62] M. Harrower and S. Fabrikant. *The Role of Map Animation for Geographic Visualization*, pages 49–65. John Wiley & Sons, Ltd, 2008.
- [63] M. Harrower, A. MacEachren, and A. L. Griffin. Developing a geographic visualization tool to support earth science learning. *Cartography and Geographic Information Science*, 27(4):279–293, 2000.
- [64] M. Harrower and B. Sheesley. Designing better map interfaces: A framework for panning and zooming. *Transactions in GIS*, 9(2):77–89, 2005.

- [65] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, Nov. 2007.
- [66] C. Hoarau and S. Christophe. Color and texture interpolation between orthoimagery and vector data. In *ISPRS Annals of Photogrammetry, Remote Sensing and the Spatial Information Sciences*, 2015.
- [67] C. Hoarau and S. Christophe. Cartographic continuum rendering based on color and texture interpolation to enhance photo-realism perception. *ISPRS Journal of Photogrammetry and Remote Sensing*, 127:27–38, 2017.
- [68] C. Hoarau, S. Christophe, and S. Mustière. Mixing, blending, merging or scrambling topographic maps and orthoimagery in geovisualizations ? In *Proceedings of the 26th International Cartographic Conference (ICC'13), Dresden, Germany*. International Cartographic Association, 25-30 August 2013.
- [69] K. Hornbæk, B. B. Bederson, and C. Plaisant. Navigation patterns and usability of zoomable user interfaces with and without an overview. *ACM ToCHI*, 9(4):362–389, 2002.
- [70] C. Hurter, A. Telea, and O. Ersoy. Moleview: An attribute and structure-based semantic lens for large element-based plots. *IEEE TVCG*, 17(12):2600–2609, 2011.
- [71] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair, and T. Möller. A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2818–2827, Dec. 2013.
- [72] M. Iwanowski. Image methamorphosis based on universal morphological interpolator. *Przegląd Elektrotechniczny*, 87(9a):234–237, 2011.
- [73] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In *Visualization Symposium (PacificVis), 2012 IEEE Pacific*, pages 1–8. IEEE, 2012.
- [74] W. Javed and N. Elmqvist. Exploring the design space of composite visualization. In *Proceedings of the 2012 IEEE Pacific Visualization Symposium, PacificVis*, pages 1–8, Washington, DC, USA, 2012. IEEE.
- [75] W. Javed, S. Ghani, and N. Elmqvist. Polyzoom: Multiscale and Multifocus Exploration in 2D Visual Spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '12*, pages 287–296. ACM, 2012.

- [76] C. E. Jones, M. Haklay, S. Griffiths, and L. Vaughan. A less-is-more approach to geovisualization – enhancing knowledge construction across multidisciplinary teams. *International Journal of Geographical Information Science*, 23(8):1077–1093, 2009.
- [77] S. Jul and G. W. Furnas. Critical zones in desert fog: Aids to multiscale navigation. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pages 97–106, New York, NY, USA, 1998. ACM.
- [78] A. Kawasaki, M. L. Berman, and W. Guan. The growing role of web-based geospatial technology in disaster response and support. *Disasters*, 37(2):201–221, 2013.
- [79] T. H. Kim, T. J. Cova, and A. Brunelle. Exploratory map animation for post-event analysis of wildfire protective action recommendations. *Natural Hazards Review*, 7(1):1–11, 2006.
- [80] T. Koch. The map as intent: variations on the theme of John Snow. *Cartographica: Int. Jour. for Geographic Information and Geovisualization*, 39(4):1–14, 2004.
- [81] B. c. Kwon, W. Javed, N. Elmqvist, and J. S. Yi. Direct manipulation through surrogate objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 627–636, New York, NY, USA, 2011. ACM.
- [82] K. Lab. Storymapjs: Maps that tell stories, 2013. [Online; accessed 07-September-2016].
- [83] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1520–1536, Sept. 2012.
- [84] H. Lee, S. Lee, N. Kim, and J. Seo. JigsawMap: Connecting the Past to the Future by Mapping Historical Textual Cadasters. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 463–472. ACM, 2012.
- [85] W. Liang. *Numerical Simulation of Czochralski Bulk Crystal Growth Process: Investigation of Transport Effects in Melt and Gas Phases*. PhD thesis, Université Catholique de Louvain, Belgium, 8 2008. Chapter 2.3: Unstructured 2D mesh generation.

- [86] H. Lieberman. Powers of ten thousand: Navigating in large information spaces. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology*, UIST, pages 15–16, New York, NY, USA, 1994. ACM.
- [87] L. Liu, G. Wang, B. Zhang, B. Guo, and H.-Y. Shum. Perceptually based approach for planar shape morphing. In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pages 111–120. IEEE, 2004.
- [88] D. Lloyd and J. Dykes. Human-Centered Approaches in Geovisualization Design: Investigating Multiple Methods Through a Long-Term Case Study. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2498–2507, Dec. 2011.
- [89] A. Lobben. Classification and application of cartographic animation. *The Professional Geographer*, 55(3):318–328, 2003.
- [90] A. Lobben. Classification and application of cartographic animation. *The Professional Geographer*, 55(3):318–328, 2003.
- [91] A. Lobben. Influence of data properties on animated maps. *Annals of the Association of American Geographers*, 98(3):583–603, 2008.
- [92] M.-J. Lobo, C. Appert, and E. Pietriga. Mapmosaic: dynamic layer compositing for interactive geovisualization. *International Journal of Geographical Information Science*, 31(9):1818–1845, 2017.
- [93] M.-J. Lobo, E. Pietriga, and C. Appert. An Evaluation of Interactive Map Comparison Techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3573–3582. ACM, 2015.
- [94] P. Longley. *Geographic information systems and science*. John Wiley & Sons, 2005.
- [95] S. Luz and M. Masoodian. Readability of a background map layer under a semi-transparent foreground layer. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces, AVI '14*, pages 161–168. ACM, 2014.
- [96] A. MacEachren, D. Xiping, F. Hardisty, D. Guo, and G. Lengerich. Exploring high-d spaces with multiform matrices and small multiples. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 31–38. IEEE, 2003.

- [97] A. M. MacEachren. The roles of maps, from some truth with maps: A primer on symbolization and design. *The Map Reader: Theories of Mapping Practice and Cartographic Representation*, pages 244–251, 2011.
- [98] A. M. MacEachren, F. P. Boscoe, D. Haug, and L. Pickle. Geographic visualization: Designing manipulable maps for exploring temporally varying georeferenced statistics. In *Proceedings of the Symposium on Information Visualization*, pages 87–94. IEEE, 1998.
- [99] S. Maggi, S. I. Fabrikant, J.-P. Imbert, and C. Hurter. How do display design and user characteristics matter in animations? an empirical study with air traffic control displays 1. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 51(1):25–37, 2016.
- [100] B. Marshall, B. Jacobs, and A. Shaw. *Losing ground*, 2014.
- [101] M. Matsuoka, T. T. Vu, and F. Yamazaki. Automated damage detection and visualization of the 2003 bam, iran, earthquake using high-resolution satellite images. In *Proc. 25th Asian Conf. Remote Sens*, pages 841–845, 2004.
- [102] A. McLean, K. Quealy, M. Ericson, and A. Tse. Satellite photos of japan, before and after the quake and tsunami. *The New York Times*, March 2011.
- [103] W. Meulemans, J. Dykes, A. Slingsby, C. Turkay, and J. Wood. Small multiples with gaps. *IEEE transactions on visualization and computer graphics*, 23(1):381–390, 2017.
- [104] C. C. Miller. A beast in the field: The google maps mashup as gis/2. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 41(3):187–199, 2006.
- [105] C. E. Murphy. Intellectual highlighting of remote sensing imagery for better image map design. In *Proceedings of the 27th International Cartographic Conference (ICC'15), Rio de Janeiro, Brazil*, 2015.
- [106] K. Ooms, A. Coltekin, P. De Maeyer, L. Dupont, S. Fabrikant, A. Incoul, M. Kuhn, H. Slabbinck, P. Vansteenkiste, and L. Van der Haegen. Combining user logging with eye tracking for interactive and dynamic applications. *Behavior Research Methods*, 47(4):977–993, Dec 2015.
- [107] T. Opach, T. Midtbø, and A. Nossun. A new concept of multi-scenario, multi-component animated maps for the visualization of spatio-temporal landscape

- evolution. *Miscellanea Geographica-Regional Studies on Development*, 15:215–229, 2011.
- [108] A. Ourednik. The third dimension of political mapping: exploiting map interactivity for a better understanding of election and referendum results. *Cartography and Geographic Information Science*, 44(4):284–295, 2017.
- [109] L. Palen, R. Soden, T. J. Anderson, and M. Barrenechea. Success & scale in a data-producing organization: The socio-technical evolution of openstreetmap in response to humanitarian events. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 4113–4122. ACM, 2015.
- [110] F. Paul. Revealing glacier flow and surge dynamics from animated satellite image sequences: examples from the karakoram. *The Cryosphere*, 9(6):2201–2214, 2015.
- [111] E. Pietriga. A Toolkit for Addressing HCI Issues in Visual Language Environments. In *Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing, VLHCC '05*, pages 145–152, Washington, DC, USA, 2005. IEEE.
- [112] E. Pietriga, C. Appert, and M. Beaudouin-Lafon. Pointing and beyond: An operationalization and preliminary evaluation of multi-scale searching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '07*, pages 1215–1224, New York, NY, USA, 2007. ACM.
- [113] E. Pietriga, O. Bau, and C. Appert. Representation-Independent In-Place Magnification with Sigma Lenses. *IEEE TVCG*, 16(3):455–467, 2010.
- [114] E. Pietriga, O. Bau, and C. Appert. Representation-Independent In-Place Magnification with Sigma Lenses. *IEEE Transactions on Visualization and Computer Graphics*, 16(3):455–467, May 2010.
- [115] C. Pindat, E. Pietriga, O. Chapuis, and C. Puech. Jellylens: content-aware adaptive lenses. In *Proceedings of the 25th annual ACM symposium on User interface software and technology, UIST '12*, pages 261–270, New York, NY, USA, 2012. ACM.
- [116] M. D. Plumlee and C. Ware. Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *ACM ToCHI*, 13(2):179–209, 2006.

- [117] T. Porter and T. Duff. Compositing digital images. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84, pages 253–259, New York, NY, USA, 1984. ACM.
- [118] M. T. Rahman, M. Al-Amin, J. B. Bakkre, A. R. Chowdhury, and M. A.-A. Bhuiyan. A novel approach of image morphing based on pixel transformation. In *International Conference on Computer and Information Technology*, pages 1–5. IEEE, 2007.
- [119] G. raldine Del Mondo, J. G. Stell, C. Claramunt, and R. my Thibaud. A graph model for spatio-temporal evolution. *Journal of Universal Computer Science*, 16(11):1452–1477, 2010.
- [120] P. Raposo and C. Brewer. Comparison of topographic map designs for overlay on orthoimage backgrounds. In *International Cartographic Conference*, pages 3–8, 2011.
- [121] D. F. Reilly and K. M. Inkpen. Map morphing: Making sense of incongruent maps. In *Proceedings of Graphics Interface 2004*, GI '04, pages 231–238, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004.
- [122] A.-K. Reuschel and L. Hurni. Mapping literature: Visualisation of spatial uncertainty in fiction. *The Cartographic Journal*, 48(4):293–308, 2011.
- [123] G. G. Robertson, S. K. Card, and J. D. Mackinlay. Information visualization using 3d interactive animation. *Commun. ACM*, 36(4):57–71, 1993.
- [124] A. C. Robinson. Highlighting in geovisualization. *Cartography and Geographic Information Science*, 38(4):373–383, 2011.
- [125] M. Rønne Jakobsen and K. Hornbæk. Sizing up visualizations: Effects of display size in focus+context, overview+detail, and zooming interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1451–1460, New York, NY, USA, 2011. ACM.
- [126] R. E. Roth. Cartographic interaction primitives: Framework and synthesis. *The Cartographic Journal*, 49(4):376–395, 2012.
- [127] R. E. Roth. An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2356–2365, 2013.

- [128] R. E. Roth. An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization. *IEEE TVCG*, 19(12):2356–2365, 2013.
- [129] R. E. Roth. Interactive maps: What we know and what we need to know. *Journal of Spatial Information Science*, (6):59–115, 2013.
- [130] R. E. Roth, A. Çöltekin, L. Delazari, H. F. Filho, A. Griffin, A. Hall, J. Korpi, I. Lokka, A. Mendonça, K. Ooms, and C. P. van Elzaker. User studies in cartography: opportunities for empirical research on interactive maps and visualizations. *International Journal of Cartography*, 0(0):1–29, 0.
- [131] R. E. Roth and M. Harrower. Addressing map interface usability: Learning from the lakeshore nature preserve interactive map. *Cartographic Perspectives*, (60), 2008.
- [132] R. E. Roth and K. S. Ross. Extending the google maps api for event animation mashups. *Cartographic Perspectives*, (64):21–40, 2009.
- [133] R. E. Roth, K. S. Ross, B. G. Finch, W. Luo, and A. M. MacEachren. A user-centered approach for designing and developing spatiotemporal crime analysis tools.
- [134] R. E. Roth, K. S. Ross, and A. M. MacEachren. User-centered design for interactive maps: A case study in crime analysis. *ISPRS International Journal of Geo-Information*, 4(1):262–301, 2015.
- [135] P. Ruchikachorn and K. Mueller. Learning visualizations by analogy: Promoting visual literacy through visualization morphing. *IEEE transactions on visualization and computer graphics*, 21(9):1028–1044, 2015.
- [136] F. Schmid. Distancetouch @ opensciencemap towards task-specific map interaction. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on MapInteraction*, MapInteract '13, pages 63–65. ACM, 2013.
- [137] J. Schmidt, M. E. Gröller, and S. Bruckner. Vaico: Visual analysis for image comparison. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2090–2099, Dec 2013.
- [138] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, Dec. 2012.

- [139] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):1139–1148, 2010.
- [140] S. Sharma, L. Karklis, and G. Thorp. How the islamic state is carving out a new country. *The Washington Post*, June 2014.
- [141] B. Shneiderman. Direct manipulation: a step beyond programming languages. *IEEE Computer*, 16(8):57–69, 1983.
- [142] B. Shneiderman. Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11(6):70–77, Nov. 1994.
- [143] A. Slingsby, J. Dykes, J. Wood, and R. Radburn. Designing an exploratory visual interface to the results of citizen surveys. *International Journal of Geographical Information Science*, 28(10):2090–2125, 2014.
- [144] T. Slocum, R. Sluter, F. Kessler, and S. Yoder. A qualitative evaluation of maptime, a program for exploring spatiotemporal point data. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 39(3):43–68, 2004.
- [145] B. Speckmann and K. Verbeek. Necklace Maps. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):881–889, 2010.
- [146] J. Sweller, J. J. G. van Merriënboer, and F. G. W. C. Paas. Cognitive architecture and instructional design. *Educational Psychology Review*, 10(3):251–296, 1998.
- [147] C. Tominski, S. Gladisch, U. Kister, R. Dachselt, and H. Schumann. A survey on interactive lenses in visualization. *EuroVis State-of-the-Art Reports*, 3, 2014.
- [148] G. Touya and J.-F. Girres. ScaleMaster 2.0: a ScaleMaster extension to monitor automatic multi-scales generalizations. *Cartography and Geographic Information Science*, 40(3):192–200, 2013.
- [149] E. R. Tufte and G. M. Schmieg. The visual display of quantitative information. *American Journal of Physics*, 53(11):1117–1118, 1985.
- [150] B. Tversky. Some ways that maps and diagrams communicate. In *Spatial Cognition II, Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications*, pages 72–79, London, UK, UK, 2000. Springer-Verlag.
- [151] B. Tversky, J. B. Morrison, and M. Betrancourt. Animation: Can it facilitate? *Int. J. Hum.-Comput. Stud.*, 57(4):247–262, Oct. 2002.

- [152] E. L. Usery, J. Choi, and M. P. Finn. *Modeling Sea-Level Rise and Surge in Low-Lying Urban Areas Using Spatial Data, Geographic Information Systems, and Animation Methods*, pages 11–30. Springer, 2010.
- [153] I. Vasiliev, S. Freundschuh, D. M. Mark, G. D. Theisen, and J. McAvoy. What is a map? *The Cartographic Journal*, 27(2):119–123, 1990.
- [154] D. Vronay and S. Wang. Designing a compelling user interface for morphing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 143–149. ACM, 2004.
- [155] C. Ware and M. Lewis. The DragMag Image Magnifier. In *Conference Companion on Human Factors in Computing Systems*, CHI '95 companion, pages 407–408, New York, NY, USA, 1995. ACM.
- [156] D. Watkins. What china has been building in the south china sea. *The New York Times*, Feb 2016.
- [157] D. C. Wilson, H. R. Lipford, E. Carroll, P. Karr, and N. Najjar. Charting new ground: modeling user behavior in interactive geovisualization. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, page 61. ACM, 2008.
- [158] A. J. Wimmers and C. S. Velden. Mimic: A new approach to visualizing satellite microwave imagery of tropical cyclones. *Bulletin of the American Meteorological Society*, 88(8):1187, 2007.
- [159] J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 143–146, New York, NY, USA, 2011. ACM.
- [160] J. Wood, D. Badawood, J. Dykes, and A. Slingsby. Ballotmaps: Detecting name bias in alphabetically ordered ballot papers. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2384–2391, 2011.
- [161] J. Wood, J. Dykes, A. Slingsby, and K. Clarke. Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1176–1183, Nov. 2007.

- [162] J. Wood, J. Dykes, A. Slingsby, and K. Clarke. Interactive visual exploration of a large spatio-temporal dataset: Reflections on a geovisualization mashup. *IEEE transactions on visualization and computer graphics*, 13(6):1176–1183, 2007.
- [163] K. Yatani, K. Partridge, M. Bern, and M. W. Newman. Escape: A target selection technique using visually-cued gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 285–294, New York, NY, USA, 2008. ACM.
- [164] J. S. Yi, Y. ah Kang, and J. Stasko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE transactions on visualization and computer graphics*, 13(6):1224–1231, 2007.

A

APPENDIX

A.1. MAPS USED IN THE FIRST EVALUATION

All maps are available in (<http://ilda.saclay.inria.fr/mapmuxing/chi2015/>). Figure A.1 shows a screenshot of the website.

A.2. IMAGES AND MATERIAL USED FOR THE *Baia* EVALUATION

A.2.1. IMAGES AND ANIMATIONS USED IN THE FIRST EXPERIMENT

All the images and animations used in the first experiment are available in (<http://ilda.saclay.inria.fr/mapmuxing/baia/study>). For each before-and-after image we present the before and after images, the region of interest and the four animations. Figure A.2 shows a screenshot of the website.

A.2.2. TUTORIAL USED FOR *Baia*'S PROTOTYPE EVALUATION

The tutorial created for the user study of the authoring tool is also available, in (<http://ilda.saclay.inria.fr/mapmuxing/baia/authoringtool/instructions>). Figure A.3 shows the tutorial topics and Figure A.4 shows an example of two of the tutorial videos.

A



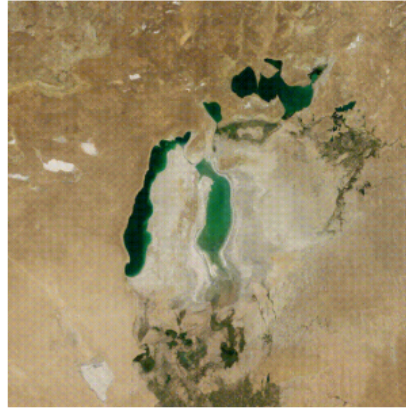
Figure A.1: Screenshot of the webpage containing all the maps used in the experiment presented in Chapter 3.

	Before	After	ROI	Blend (1 stage)	Blend (2 stages)	Custom (1 stage)	Custom (2 stages)
01							
<p>Description: Paris-Saclay is a scientific and technological cluster 20km away from Paris currently under construction. The earliest settlements are from the 1950s, and this area was subsequently extended several times during the 1970s and 2000s. Several projects are underway to continue the development of the campus. Here we display the evolution of a small part of it between 2008 and 2016, revealing the construction of the Digiteo and PCRI buildings (Source: IGN). Rank the following animations according to — <i>appearance of these buildings</i>.</p> <p>Source: http://professionnels.ign.fr/orthoimages (48°42'46.1"N 2°09'57.3"E)</p>							
02							
<p>Description: To expand the possibilities for beachfront tourist development, Dubai, part of the United Arab Emirates, undertook a massive engineering project to create hundreds of artificial islands along its Persian Gulf coastline. These islands were shaped into recognizable forms, including two large palm trees. The first Palm Island constructed was Palm Jumeirah. We display its evolution from 2000 to 2011. (Source: NASA Earth Observatory). Rank the following animations according to — <i>appearance of the island</i>.</p> <p>Source: http://earthobservatory.nasa.gov/Features/WorldOfChange/dubai.php</p>							
03							
<p>Description: The Topaz Solar Farm in Central California is one of the largest in the U.S. It consists of 9 million solar panels across 24.6 square kilometers. Construction began in 2012 and was completed in late 2014. We display the evolution between 2011 and 2015. In 2011 the land had been used for agriculture (green blocks). In 2015 the solar installations appear as dark blocks. (Source: NASA Images of change). Rank the following animations according to — <i>appearance of the Topaz Solar Farm</i>.</p> <p>Source: http://climate.nasa.gov/images-of-change?id=541#541-topaz-solar-farm-installation-california</p>							
04							
<p>Description: Construction on the Lower Se San 2 Dam is underway on the Tonle Sapok, a tributary of the Mekong River. The dam is expected to generate 400 megawatts of power, but some scientists warn that completion of the dam will reduce the productivity of fisheries and biodiversity in the Mekong River Basin. This pair of images from the NASA/USGS Landsat 8 satellite (December 22, 2014) and Planet (January 15, 2015) shows the rapid progress of construction at the site (Source: Planet Gallery). Rank the following animations according to — <i>changes in the Lower Se San 2 Dam</i>.</p> <p>Source: https://www.planet.com/gallery/ (Lower Se San 2 Dam, Cambodia)</p>							
05							
<p>Description: In the Southern Hemisphere summer of 2002, scientists monitoring daily satellite images of the Antarctic Peninsula watched in amazement as almost the entire Larsen B Ice Shelf splintered and collapsed in just over one month. Here we present its evolution between January 31 and March 7. By March 7, the shelf had disintegrated into a blue-tinted mixture (mélange) of partly-melted snow and icebergs (Source: NASA Earth Observatory). Rank the following animations according to — <i>changes in the Larsen B Ice Shelf</i>.</p> <p>Source: http://earthobservatory.nasa.gov/Features/WorldOfChange/larsenb.php</p>							
06							
<p>Description: In the 1960s, the Soviet Union undertook a major water diversion project on the arid plains of Kazakhstan, Uzbekistan, and Turkmenistan. The region's two major rivers, fed by snowmelt and precipitation in faraway mountains, were used to transform the desert into farms for cotton and other crops. Before the project, the rivers flowed down from the mountains and made the Aral Sea, once the fourth largest lake in the world. Although irrigation made the desert bloom, it devastated the Aral Sea. We present the evolution between August 2000 and August 2010. (Source: NASA Earth Observatory). Rank the following animations according to — <i>changes in the Aral Sea</i>.</p> <p>Source: http://earthobservatory.nasa.gov/Features/WorldOfChange/aral_sea.php</p>							
07							
<p>Description: China has placed runways and radar facilities on new islets in the South China Sea, built by piling huge amounts of sand onto reefs. China has focused its efforts on building ports, three airstrips, radar facilities and other military buildings on the islands. We present the building evolution of one island between December 2014 and February 2015. (Source: The New York Times). Rank the following animations according to — <i>new island's growth</i>.</p> <p>Source: http://www.nytimes.com/interactive/2015/07/30/world/asia/what-china-has-been-building-in-the-south-china-sea-2016.html?_r=0</p>							
08							
<p>Description: Below the densely forested slopes of the Appalachian Mountains in southern West Virginia in the United States, there is coal under layers. To uncover this coal profitably, mining companies engineer large surface mines. We display the growth of the Hobet mine from 1984 to 1986. The natural landscape of the area is dark green forested mountains, crossed by streams and indented by hollows. The active mining areas appear off-white, while areas being reclaimed with vegetation appear light green. (Source: NASA Earth Observatory). Rank the following animations according to — <i>growth of the Hobet mine</i>.</p> <p>Source: http://earthobservatory.nasa.gov/Features/WorldOfChange/hobet.php</p>							
09							
<p>Description: The Paraguay-Parana River system is the second largest river system in South America — second only to the Amazon. More than 100 million people and some of the rarest species on Earth depend on its waters for survival. We display the evolution between 1985, shortly after construction began on the Yacireta Dam, a joint hydroelectric project between Paraguay and Argentina, and June 2010. River levels rose dramatically upon completion of the dam, initially displacing 15,000 residents and endangering the homes of 800,000 more (Source NASA: Images of change). Rank the following animations according to — <i>river levels rise</i>.</p> <p>Source: http://climate.nasa.gov/images-of-change?id=335#335-yacireta-dam-impact-paraguay</p>							

Figure A.2: Screenshot of the webpage containing all the images and animations used in the first experiment presented in Chapter 5.

Baia Tutorial

Baia is a system for creating animated transitions between a source and a target satellite images, such as the animation below that illustrates the drying of the Aral Sea.



The following videos address all the necessary steps for using these animation primitives, and creating more advanced transitions that e.g., restrict the scope of an animation or feature multiple stages:

Creating a simple transition

- [How to create a new Before/After Transition?](#)
- [How to add masks?](#)
- [How to use the Color Threshold selection tool?](#)
- [How to use the Lasso selection tool?](#)

Staging

- [How to change the duration of a stage?](#)
- [How to add stages?](#)
- [How to use other animation primitives?](#)
- [How to reorder stages?](#)

Aesthetics

- [How to create a blurred border?](#)

Figure A.3: Screenshot of the tutorial index available to participants in the second experiment of Chapter 5.

How to add masks?



How to use the Color Threshold selection tool?

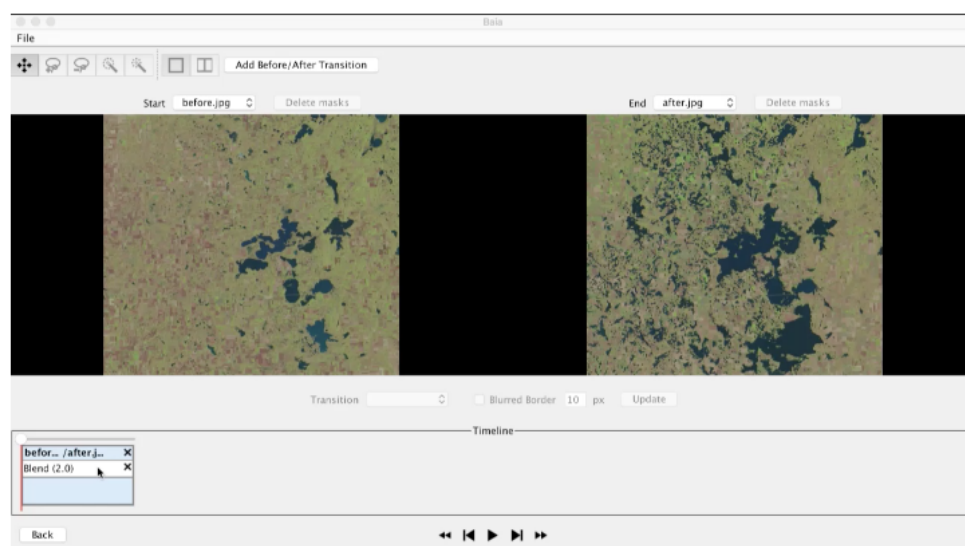


Figure A.4: Screenshot of two example of the videos presented in the tutorial.

Titre : Transitions interactives pour des Applications Cartographiques

Mots clés : Géovisualization, Techniques d'interaction, Applications Cartographiques

Résumé : Différentes représentations d'une région géographique doivent être combinées pour des tâches variées, par exemple mettre à jour une base de données géographiques. Cette thèse étudie comment combiner ces représentations dans une scène interactive. On présente trois contributions. D'abord, on présente une évaluation empirique de cinq techniques existantes de comparaison de cartes. Les résultats suggèrent que les techniques basées sur la superposition sont plus efficaces que celles basées sur la juxtaposition. Ensuite, on présente MapMosaic, une nouvelle approche pour combiner des couches dans l'espace. Cette technique permet de combiner les représentations en considérant leurs objets et attributs. Finalement, on introduit Baia: un modèle d'animation pour créer des transitions entre des images satellite avant-après pour mieux représenter les changements. Les animations créées avec Baia sont perçues comme plus réalistes et plus efficaces pour attirer l'attention des spectateurs que le fondu classique.

Title : Interactive transitions for Map Applications

Keywords : Geovisualization, Interaction Techniques, Cartographic Applications

Abstract : Different tasks, such as updating a geographical database, require users to combine multiple representations of a geographical region. This thesis studies how to combine these different representations into one interactive scene, either spatially or temporally. We present three contributions. First, we evaluate empirically five existing techniques to compare maps. We find that techniques that superimpose layers are most efficient than techniques that juxtapose them. Then, we present a novel approach to combine layers spatially, MapMosaic. This technique enables users to combine maps according to their objects and attributes, and was well received by experts. Finally, we introduce Baia: an animation framework and an authoring tool to create transitions between satellite images to better convey geographic evolution. The animations created used Baia are perceived as more realistic and better at focusing viewer's attention than blending.