MapMosaic: Dynamic Layer Compositing for Interactive Geovisualization

María-Jesús Lobo, Caroline Appert, Emmanuel Pietriga

RESEARCH ARTICLE

MapMosaic: Dynamic Layer Compositing
for Interactive Geovisualization

María-Jesús Lobo[a,b,c*], Caroline Appert[b,a,c] and Emmanuel Pietriga[a,b,c]

[a]*INRIA*; [b]*Université Paris-Sud & CNRS*; [c]*Université Paris-Saclay, Orsay, France*

GIS software applications and other mapping tools enable users to correlate data from multiple layers and gain insight from the resulting visualizations. However, most of these applications only feature basic, monolithic layer compositing techniques. These techniques do not always support users effectively in their tasks, as we observed during interviews with GIS experts. We introduce MapMosaic, a novel approach based on dynamic visual compositing that enables users to interactively create and manipulate local composites of multiple vector and raster map layers, taking into account the semantics and attribute values of objects and fields in the compositing process. We evaluate MapMosaic's interaction model against that of QGIS (a widely-used desktop GIS) and MAPublisher (a professional cartography tool) using the "Cognitive Dimensions" framework and through an analytical comparison, showing that MapMosaic's model is more flexible and can support users more effectively in their tasks. We also report on feedback obtained from experts, which further confirms the potential of this highly dynamic approach to map layer compositing.

Keywords: Geovisualization; Interactive Layer Compositing; User Interfaces.

*Corresponding author. Email: maria-jesus.lobo@inria.fr

1. Introduction

One of the main human-computer interaction challenges in geographical information systems is to enable users to quickly access, correlate and make sense of the many, possibly very heterogeneous, datasets that must be brought together to achieve a given task (Elias *et al.* 2008, Lobo *et al.* 2015). Indeed, while cartographic representation helps users think and communicate visually (MacEachren 2011), visual representation alone is often insufficient to cope with the quantity and diversity of datasets brought together in application areas such as: disaster mapping (Palen *et al.* 2015) and emergency response management; transportation networks (Bast *et al.* 2014); multi-criteria decision making (Jankowski *et al.* 2001); or the news media (Gao *et al.* 2014). User interaction can be seen as complementary to visual representation and helps alleviate this problem. It enables users to adapt the visual representation so as to emphasize relevant subsets of the data depending on the task at hand (Andrienko and Andrienko 1999, Roth 2013a,b).

Some geovisualization systems provide elaborate interactive data visualizations (see Chang *et al.* (2007), Slingsby *et al.* (2014), Speckmann and Verbeek (2010), Wood *et al.* (2007) for representative examples), but those systems are domain-specific. They mostly enable interactions dedicated to the specific data managed by a particular application. When considering general-purpose GIS user interface front-ends, from both research projects (Cruz *et al.* 2013, Grosso *et al.* 2013) and industrial-strength products such as ArcGIS (`http://www.esri.com/software/arcgis`) and QGIS (`http://www.qgis.org`), 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 layer composites is cumbersome, involving many indirect manipulations. 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 (Jones *et al.* 2009).

As we will see in Section 3, many of these tasks require users to reconfigure which layers are visible and how they are arranged. But having to repeatedly toggle between layers represents a significant impediment from the users' perspective, and there are situations where basic, monolithic toggling of layers is not the most relevant way of visualizing and navigating in the data. We argue that digital maps are highly-structured, semantics-rich user interface components that have the potential to offer more elaborate forms of interaction between users and map-based systems (Wilson *et al.* 2008, Schmid 2013). Based on the observation that layer composition plays an essential role in many tasks (Elias *et al.* 2008), we introduce MapMosaic, an approach to dynamic visual compositing that takes advantage of these semantics and structure, enabling novel techniques to interactively create and manipulate local composites of multiple vector and raster map layers.

After discussing our motivation in taking this approach, we report on interviews conducted with GIS experts. These interviews helped identify interactive layer compositing, which is based on the technique known as digital image compositing in the field of computer graphics (Porter and Duff 1984), as a key aspect of users' interaction with map content. We then introduce our main contribution: a dynamic map compositing model, and describe a prototype implementation (see Figure 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* (Green and Petre 1996), 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
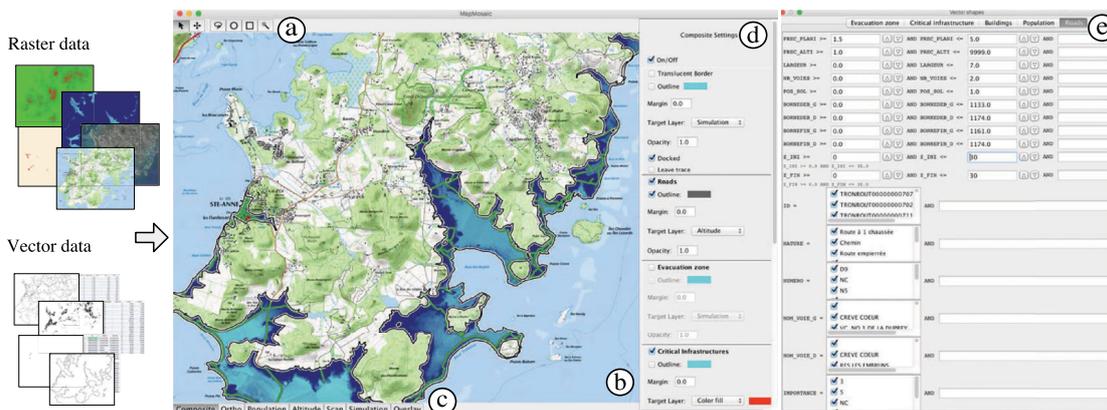
Figure 1. A proof-of-concept implementation of the MapMosaic dynamic compositing model and inter-action techniques. (a) Toolbar to navigate the map, and to create & select areas. (b) Map viewer. (c) Access to individual layers. (d) Compositing area inspector. (e) Query builder for compositing region filters.

from workshops with GIS expert users. We conclude with directions for future work.

2. Background and Motivation

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 (Elias *et al.* 2008), and communicate them.

Beyond the compositing techniques commonly encountered in interfaces for comparing two maps or images (Lobo *et al.* 2015), recent work in cartographic research has produced methods to overlay ortho-rectified imagery and topographic maps or other vector data (Hoarau and Christophe 2015, Hoarau *et al.* 2013, Raposo and Brewer 2011). Such methods make use of digital compositing (Porter and Duff 1984), 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. The resulting map-compositing techniques are quite elaborate in terms of graphics rendering, but are designed to produce static composite maps. They do not support the interactive exploration tasks we discuss in the next sections effectively. On the industry side, MapStack (`http://mapstack.stamen.com`) provides some compositing capabilities, limited in scope to three predefined areas (water, buildings, parks). Such simple approaches are targeted at lay users who want to easily create basic personalized maps (Christophe 2012, Ballatore and Bertolotto 2015).

GIS and cartography tools such as MAPublisher offer much 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 (Lobo *et al.* 2015), 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 6, 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 (Sweller *et al.* 1998) 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 (Shneiderman 1983). 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.

3.   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.

3.1.   *GIS Operators*

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.

3.1.1.   *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.

3.1.2. 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.

3.1.3. 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.

3.1.4. 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.

3.2. 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.

3.2.1. 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 (Touya and Girres 2013), 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*.

3.2.2. 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 two main phases: *before* 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* (Palen *et al.* 2015) 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 (Palen *et al.* 2015), 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.

3.3. *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 (Porter and Duff 1984); and combining two dense and visually complex layers such as, *e.g.*, a topographic map and ortho-imagery, will require interactive navigation (Lobo *et al.* 2015), 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*.

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 (Cockburn *et al.* 2009), 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 (Jul and Furnas 1998) 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 object 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.   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 (Jones *et al.* 2009).

MapMosaic'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 (Shneiderman 1983): MapMosaic exposes those masks as first-class, interactive objects (Kwon *et al.* 2011) that users can create and iteratively modify without resorting to complex tools or dialog boxes. Parameters 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.

4.1. *Layers*

As shown in Figure 1, users can load both raster and vector layers in the system. Each raster layer is made available directly in the Map viewer (Figure 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 (Elias *et al.* 2008). Users can also access a rendering of each layer in isolation by clicking on the corresponding tab (Figure 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.

---

[1]Drawing 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.

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 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.

4.2. *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 (Beaudouin-Lafon and Mackay 2000) 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"* (Beaudouin-Lafon and Mackay 2000), 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 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 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 3), so as to achieve an effect similar to that enabled by blending lenses (Pietriga *et al.* 2010). 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 2-1), choosing SIMULATION as the target layer in order to identify the zones that would be flooded in case of a tsunami.
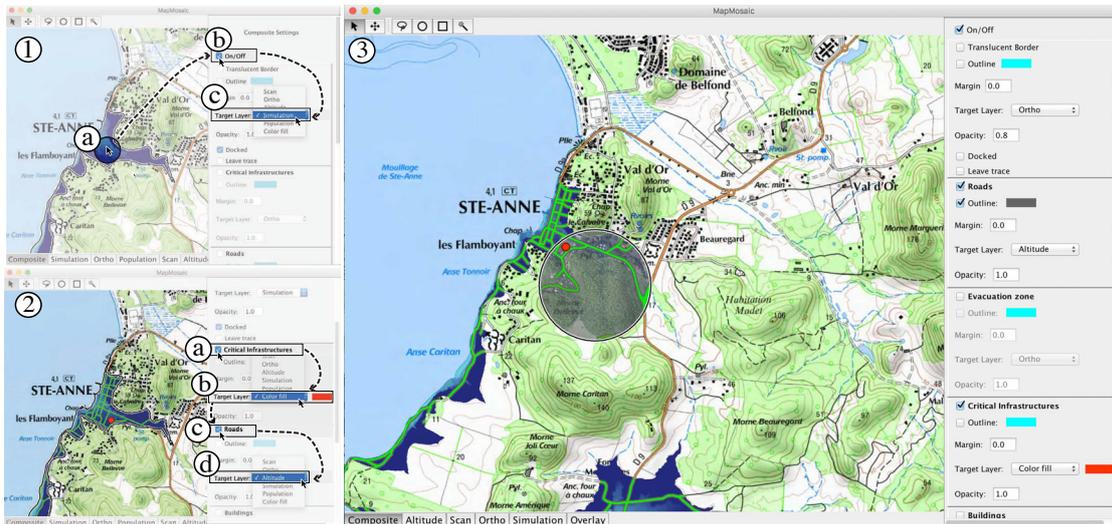
Figure 2. (1) Creating a compositing region from an existing vector object: (a) the EVACUATION ZONE is selected by clicking on it, (b) the selection is turned into a compositing region, and (c) SIMULATION gets chosen as the target layer. (2) Activating filters in the region: (a) the CRITICAL INFRASTRUCTURE 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.

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 1-d). The changes are reflected on-the-fly in the map view, as parameters get adjusted (Shneiderman 1994), 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 (Bier *et al.* 1993). 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) 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. *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 subpanel in the region's inspector (see, *e.g.*, Critical infrastructure or Roads in Figure 2-2). More complex rules that involve elaborate selection criteria (for instance, "all buildings at least 10 meters high", Figure 4-b) are created using the query builder (Figure 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 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 (Beaudouin-Lafon and Mackay 2000), 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.

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 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 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 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 3-b). Finally, he creates one last compositing region using the POPULATION grid as a target layer (Figure 3-c) to check that the refuge zone is large-enough to accommodate all people in the area.
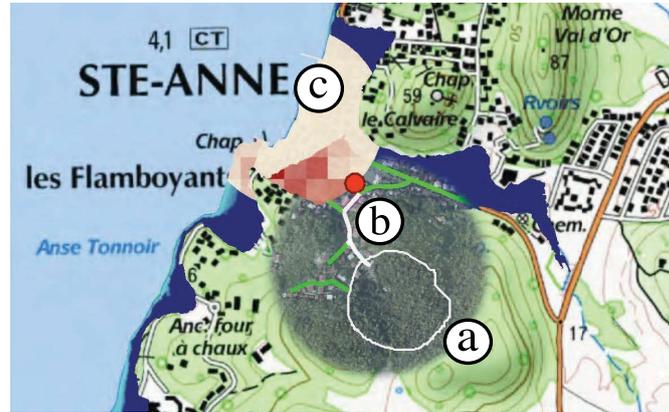
Figure 3. (a) Refuge zone found by examining aerial imagery. (b) Road connecting the critical infrastructure (symbolized by the red spot) to the refuge zone. (c) Compositing region showing population grid in the evacuation zone.

5. 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"* (Roth 2013b). 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. Map-Mosaic'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 (Liang 2008).

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

Table 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) Min / Max	597.8 (2027.0) 11 / 9713	8.6 (6.8) 5 / 89	6.9 (4.6) 5 / 77
Pixel area Mean (Standard deviation) Min / Max	9526 (32265.8) 1.8 / 180567.3	16.8 (30.2) 0.32 / 369.81	3.6 (4.0) 0.2 / 72.3
PERFORMANCE			
Creating *CR* (in ms)	16	230	915
Editing *CR* (in ms)	2	23	73
Moving *CR* (in fps)	170[*]	170[*]	170[*]

[*] Constant, depends on the GPU's characteristics.

Table 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) Min / Max	7.4 (7.2) 2 / 52	10.7 (11.6) 2 / 154
Pixel length Mean (Standard deviation) Min / Max	16.2 (23.9) 0.45 / 249.16	20.0 (22.6) 0.42 / 295.18
PERFORMANCE		
Creating *CR* (in ms)	69	620
Editing *CR* (in ms)	15	37
Moving *CR* (in fps)	170[*]	170[*]

[*] Constant, depends on the GPU's characteristics.

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 1 and 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.

6. Comparison with MAPublisher and QGIS

As mentioned earlier, some industrial-strength GIS front-ends also feature advanced graphics editing and compositing capabilities. So does MAPublisher[TM], which works as a plugin for

Adobe Illustrator^{TM}, 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 graphics 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 Appendix A: 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 Green and Petre (1996). 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.

6.1. *Viscosity, Premature Commitment and Progressive Evaluation*

These four 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" (Blackwell and Green 2003)*.

As detailed in Appendix A, 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 Appendix A, 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 Appendix A 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 the appendix. 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*.

6.2. *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 Appendix A. 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.

6.3. *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 Appendix A, 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.

6.4. *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.

7.   Experts' feedback

Different articles have been promoting empirical research in the visualization community starting approximately ten years ago – see, *e.g.*, (Carpendale 2008) and (Sedlmair *et al.* 2012) – and an increasing number of papers are now reporting on user experience or user performance evaluations (Lam *et al.* 2012). Formal testing sessions that involve basic tasks, while required to evaluate *user performance*, seldom yield meaningful results (Fuhrmann *et al.* 2005) 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 (Carpendale 2008), and because of the need for users to feel engaged with the data (Lloyd and Dykes 2011). *User experience* evaluation, *"eliciting subjective feedback and opinions [about a visualization system]"* (Isenberg *et al.* 2013), is more relevant in this context. Some studies evaluate usability by asking users to perform specific tasks (Roth and Harrower 2008), possibly using tailor-made applications (Jones *et al.* 2009). However, we were interested not only in evaluating MapMosaic for certain tasks or usage scenarios (Sedlmair *et al.* 2012), 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"* (Isenberg *et al.* 2013). Participants were encouraged to talk about their specific activities and then engaged into discussions about how MapMosaic could be helpful in such contexts.

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 3. 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 ($\sim$2 hours each) about scenarios in which MapMosaic would be useful. This section summarizes these discussions.

7.1.   *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-

Figure 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.
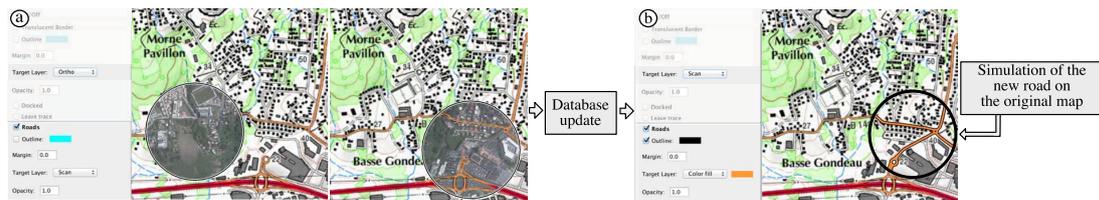

Figure 5. (a) Scanning ortho-imagery for new roads. (b) A compositing region with styling rules to preview the new road on the map.

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 results 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 re-

Figure 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.

alizing 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 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). 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 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.

7.2. *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 3): 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 (Palen *et al.* 2015) through Web sites such as Tomnod (http://www.tomnod.com).

Figure 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 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 required 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.

8.   Summary and Future Work

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 (Green and Petre 1996). 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* (Murphy 2015), which can all be achieved effortlessly with compositing regions. MapMosaic could also be useful for collaborative mapping (Fechner *et al.* 2015) to, *e.g.*, inspect the history of objects, or the activity of collaborators by organizing changes into layers, per revision or per user.

Future work could focus on coupling compositing regions with recent focus+context techniques, including JellyLenses (Pindat *et al.* 2012), that dynamically adapt their shape to the geometry of the focused objects, and RouteLenses (Alvina *et al.* 2014), that make it easier to

follow paths. Another avenue for future work would be to investigate how vector tiling (Dufilie and Grinstein 2014) can make MapMosaic scale to very large data layers.

Acknowledgement(s)

Appendix A. Comparison between QGIS, MAPublisher and MapMosaic for

[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 *raster clipping tool* 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.
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).	*Compositing regions are not georeferenced. Dragging the region will change its position on the canvas, but will not update its geographical scope.*	*Compositing regions are georeferenced. They can be dragged to update their geographical scope.*
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 *vector clipping tool* in the vector menu. 4. Select the BUILDINGS and REGION layers 5. Specify an output vector layer LOCALFILTER. 6. Select the *raster clipping tool* 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 *Art Selection* 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 *Apply as New Selection*), before selecting the other to refine it (using *Get Subset of Current Selection*). 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 *buildings* checkbox in the inspector. 3. Change the minimal height value in the query builder.
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.	*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.*	*Compositing regions are interactive objects whose filters remain editable in the inspector.*
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.

References

Alvina, J., *et al.*, 2014. RouteLens: Easy Route Following for Map Applications. *In*: *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, Como, Italy ACM, 125–128.

Andrienko, G.L. and Andrienko, N.V., 1999. Interactive maps for visual data exploration. *International Journal of Geographical Information Science*, 13 (4), 355–374.

Ballatore, A. and Bertolotto, M., 2015. Personalizing Maps. *Communications of the ACM*, 58 (12), 68–74.

Bast, H., Brosi, P., and Storandt, S., 2014. Real-time Movement Visualization of Public Transit Data. *In*: *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Dallas, Texas ACM, 331–340.

Beaudouin-Lafon, M. and Mackay, W.E., 2000. Reification, Polymorphism and Reuse: Three Principles for Designing Visual Interfaces. *In*: *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '00, Palermo, Italy New York, NY, USA: ACM, 102–109.

Bier, E.A., *et al.*, 1993. Toolglass and Magic Lenses: The See-through Interface. *In*: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, Anaheim, CA ACM, 73–80.

Blackwell, A. and Green, T.R.G., 2003. Notational systems–the cognitive dimensions of notations framework. *In*: J.M. Carroll, ed. *HCI Models, Theories, and Frameworks: Toward an Interdisciplinary Science*. Morgan Kaufmann, chap. 5, 103–134.

Carpendale, S., 2008. Information Visualization. *In*: A. Kerren, J.T. Stasko, J.D. Fekete and C. North, eds. . Berlin, Heidelberg: Springer-Verlag, chap. Evaluating Information Visualizations, 19–45.

Chang, R., *et al.*, 2007. Legible cities: Focus-dependent multi-resolution visualization of urban relationships. *IEEE Transactions on Visualization and Computer Graphics*, 13 (6), 1169–1175.

Christophe, S., 2012. Cartographic Styles between traditional and original (towards a cartographic style model). *In*: *AutoCarto Conference*, September.

Cockburn, A., Karlson, A., and Bederson, B.B., 2009. A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys*, 41 (1), 2:1–2:31.

Cruz, I.F., *et al.*, 2013. 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*, Orlando, Florida ACM, 544–547.

Dufilie, A. and Grinstein, G., 2014. Feathered Tiles with Uniform Payload Size for Progressive Transmission of Vector Data. *Web and Wireless Geographical Information Systems*., Vol. 8470 of *LNCS* Springer, 19–35.

Elias, M., *et al.*, 2008. Do I Live in a Flood Basin?: Synthesizing Ten Thousand Maps. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Florence, Italy ACM, 255–264.

Fechner, T., Wilhelm, D., and Kray, C., 2015. Ethermap: Real-time Collaborative Map Editing. *In*: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, Seoul, Republic of Korea ACM, 3583–3592.

Fuhrmann, S., *et al.*, 2005. Making Useful and Useable Geovisualization: Design and Evaluation Issues. *Exploring Geovisualization*. New York, NY, USA: Elsevier, 541–554.

Gao, T., *et al.*, 2014. NewsViews: An Automated Pipeline for Creating Custom Geovisualizations for News. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, Toronto, Ontario, Canada ACM, 3005–3014.

Green, T.R.G. and Petre, M., 1996. Usability analysis of visual programming environments: a "cognitive dimensions" framework. *Journal of Visual Languages & Computing*, 7 (2), 131–174.

Grosso, E., Perret, J., and Brasebin, M., 2013. Geoxygene: an Interoperable Platform for Geographical Application Development. *Innovative Software Development in GIS*. John Wiley & Sons, Inc., chap. 3, 67–90.

Hoarau, C. and Christophe, S., 2015. Color and Texture Interpolation between Orthoimagery and Vector Data. *In*: *ISPRS Annals of Photogrammetry, Remote Sensing and the Spatial Information Sciences*.

Hoarau, C., Christophe, S., and Mustière, S., 2013. Mixing, blending, merging or scrambling topographic maps and orthoimagery in geovisualizations ?. *In*: *Proceedings of the 26th International Cartographic Conference (ICC'13), Dresden, Germany*, 25-30 August.

Isenberg, T., *et al.*, 2013. A Systematic Review on the Practice of Evaluating Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19 (12), 2818–2827.

Jankowski, P., Andrienko, N., and Andrienko, G., 2001. Map-centred exploratory approach to multiple criteria spatial decision making. *International Journal of Geographical Information Science*, 15 (2), 101–127.

Jones, C.E., *et al.*, 2009. A less-is-more approach to geovisualization – enhancing knowledge construction across multidisciplinary teams. *International Journal of Geographical Information Science*, 23 (8), 1077–1093.

Jul, S. and Furnas, G.W., 1998. 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, San Francisco, California, USA New York, NY, USA: ACM, 97–106.

Kwon, B.c., *et al.*, 2011. Direct Manipulation Through Surrogate Objects. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, Vancouver, BC, Canada New York, NY, USA: ACM, 627–636.

Lam, H., *et al.*, 2012. Empirical Studies in Information Visualization: Seven Scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18 (9), 1520–1536.

Liang, W., 2008. Numerical Simulation of Czochralski Bulk Crystal Growth Process: Investigation of Transport Effects in Melt and Gas Phases. Thesis (PhD). Université Catholique de Louvain, Belgium Chapter 2.3: Unstructured 2D mesh generation.

Lloyd, D. and Dykes, J., 2011. 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.

Lobo, M.J., Pietriga, E., and Appert, C., 2015. An Evaluation of Interactive Map Comparison Techniques. *In*: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, Seoul, Republic of Korea ACM, 3573–3582.

MacEachren, A.M., 2011. 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*, 244–251.

Murphy, C.E., 2015. 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*.

Palen, L., *et al.*, 2015. 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*, Seoul, Republic of Korea ACM, 4113–4122.

Pietriga, E., Bau, O., and Appert, C., 2010. Representation-Independent In-Place Magnification with Sigma Lenses. *IEEE Transactions on Visualization and Computer Graphics*, 16 (3),

455–467.

Pindat, C., *et al.*, 2012. JellyLens: Content-aware Adaptive Lenses. *In*: *Proceedings of the 25th annual ACM symposium on User interface software and technology*, Cambridge, Massachusetts, USA ACM, 261–270.

Porter, T. and Duff, T., 1984. Compositing Digital Images. *In*: *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '84 New York, NY, USA: ACM, 253–259.

Raposo, P. and Brewer, C., 2011. Comparison of topographic map designs for overlay on orthoimage backgrounds. *In*: *International Cartographic Conference*, 3–8.

Roth, R.E., 2013a. An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization. *IEEE Transactions on Visualization and Computer Graphics*, 19 (12), 2356–2365.

Roth, R.E., 2013b. Interactive maps: What we know and what we need to know. *Journal of Spatial Information Science*, (6), 59–115.

Roth, R.E. and Harrower, M., 2008. Addressing Map Interface Usability: Learning from the Lakeshore Nature Preserve Interactive Map. *Cartographic Perspectives*, (60).

Schmid, F., 2013. DistanceTouch @ OpenScienceMap Towards Task-specific Map Interaction. *In*: *Proceedings of the 1st ACM SIGSPATIAL International Workshop on MapInteraction*, MapInteract '13, Orlando, Florida ACM, 63–65.

Sedlmair, M., Meyer, M., and Munzner, T., 2012. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18 (12), 2431–2440.

Shneiderman, B., 1983. Direct manipulation: a step beyond programming languages. *IEEE Computer*, 16 (8), 57–69.

Shneiderman, B., 1994. Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11 (6), 70–77.

Slingsby, A., *et al.*, 2014. Designing an exploratory visual interface to the results of citizen surveys. *International Journal of Geographical Information Science*, 28 (10), 2090–2125.

Speckmann, B. and Verbeek, K., 2010. Necklace Maps. *IEEE Transactions on Visualization and Computer Graphics*, 16 (6), 881–889.

Sweller, J., van Merrienboer, J.J.G., and Paas, F.G.W.C., 1998. Cognitive Architecture and Instructional Design. *Educational Psychology Review*, 10 (3), 251–296.

Touya, G. and Girres, J.F., 2013. ScaleMaster 2.0: a ScaleMaster extension to monitor automatic multi-scales generalizations. *Cartography and Geographic Information Science*, 40 (3), 192–200.

Wilson, D.C., *et al.*, 2008. Charting new ground: modeling user behavior in interactive geovisualization. *In*: *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, p. 61.

Wood, J., *et al.*, 2007. 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.