Vision, Modeling, and Visualization (2015)
D. Bommes, T. Ritschel, and T. Schultz (Eds.)

A Taxonomy of Integration Techniques for Spatial and Non-Spatial Visualizations

J. Sorger[1], T. Ortner[1], H. Piringer[1], G. Hesina[1], and M. E. Gröller[2]

[1]VRVis Research Center, Vienna, Austria
[2]TU Wien, Austria

Abstract

Research on visual data representations is traditionally classified into methods assuming an inherent mapping from data values to spatial coordinates (scientific visualization and real-time rendering) and methods for abstract data lacking explicit spatial references (information visualization). In practice, however, many applications need to analyze data comprising abstract and spatial information, thereby spanning both visualization domains. Traditional classification schemes do not support a formal description of these integrated systems. The contribution of this paper is a taxonomy that describes a holistic design space for integrating components of spatial and abstract visualizations. We structure a visualization into three components: Data, Visual, and Navigation. These components can be linked to build integrated visualizations. Our taxonomy provides an alternative view on the field of visualization in a time where the border between scientific and information visualization becomes blurred.

1. Introduction

Visual representation of data is traditionally classified into methods assuming an inherent mapping from data values to spatial coordinates (as in scientific visualization and real-time rendering applications in 3D spatial environments) and methods for data lacking explicit spatial references, where the spatialization is chosen [TM04] (as in information visualization). In practice, however, users often need to analyze data that contains multiple facets, like spatio-temporal and multivariate data characteristics [KH13]. In flow visualization, for instance, information visualization views (abstract) are used to select and highlight interesting attribute values in a volumetric flow representation (spatial) [Dol07]. In traffic simulation and road planning, traffic can be assessed directly in a 2D or 3D map of the city while being analyzed statistically using information-visualization views [WYL*14].

The benefits of visualizations that integrate spatial and non-spatial (abstract) data facets have been repeatedly emphasized in the visualization literature [SRH*09, TM04]. They were also the topic of panel discussions at the IEEE Vis conferences in 2003 and 2006 [HWM*06, RTM*03]. Fuchs and Hauser make a strong case for the application of multi-method visualization: "a tight integration of multiple techniques gives a key advantage towards understanding the investigated data" [FH09]. The authors identify three main advantages of multi-method visualization: improved effectiveness (each part is visualized by the most appropriate method), minimizing visual clutter, and separation between the questions of how and what to visualize. Kehrer and Hauser state that there is a lack of general concepts for handling the heterogeneity of multifaceted data [KH13].

While basic coordination techniques like brushing & linking are quite common, the ways in which spatial and abstract visualizations can benefit through integration are manifold. With the absence of a proper formalism though, it is difficult to describe and discuss this design space.

We therefore propose such a formalism by describing visualizations from the spatial and abstract domain based on a separation into a Data, Visual, and Navigation component. Our taxonomy can classify systems in terms of how the components of each domain are integrated. The contributions of this paper are:

- The specification of a taxonomy that defines the design space for integrating abstract and spatial visualizations.
- A discussion of component combinations within this design space based on state-of-the-art methods.
- An application of our model for the classification and comparison of existing integration approaches.

2. Related Work

Publications that deal explicitly with the combination of visualization methods from the abstract and the spatial domain are still rare. In their state-of-the-art report on visualization of multivariate scientific data, Fuchs and Hauser classify techniques by data type (scalar, vector field/flow, and tensor field visualization) and by the stages of the visualization pipeline where these techniques are applied [FH09]. Kehrer and Hauser give a survey of multifaceted scientific data visualization [KH13]. The authors describe five different facets of scientific data by which the discussed techniques are categorized (spatiotemporal, multivariate, multimodal, multirun, multimodel). For each facet, techniques are distinguished according to approaches for representation, computational analysis and interaction. Both surveys stress the importance of multi-method visualizations.

In their high level visualization taxonomy [TM04], Tory and Möller classify visualization algorithms based on whether they handle data discretely or continuously, and whether the spatialization is chosen, constrained or given. With their taxonomy, they aim to inspire research ideas in hybrid visualization areas.

Boukhelifa et al. [BRR03] propose a model for describing coordination in exploratory multiple view visualizations. The model uses the visualization pipeline to show which pipeline stages in connected views are linked through a coordination object. The authors specify the rudiments of coordination in a system (coordination entities, type, chronology, scope, granularity, initialization, updating, and realization). This work is an important source of inspiration to us, as we apply the concept of coordinated multiple views to the integration of visualization domains and components. However, our model is not restricted to multiple view systems, as all types of composite visualizations can be described.

The design space of composite visualization is described by Javed and Elmqvist [JE12]. The authors form a theoretical model that unifies the coordinated multiple-view paradigm with other strategies for combining visual representations, i.e., juxtaposition, superimposition, overloading, nesting, and integration. Integrated interaction, as we describe it, can take place between all types of composite visualizations. The works by Javed and Elmqvist and Boukhelifa et al. treat the issue of handling multiple visualizations in a single framework from a visual and an interactive viewpoint respectively. However they do not explore the resulting design space, much less from the perspective of heterogeneous visualization domains.

Balabanian et al. [BVG10] categorize techniques in their own application according to a 3x3 matrix that describes whether an interaction and its resulting visualization happen in the spatial, abstract or integrated domain. The idea is similar to our approach but on a higher abstraction level, as it distinguishes domains and not individual visualization components. Further, we do not consider an *integrated space* as a third category besides the *spatial* and the *abstract* one, as we partition such a space into its spatial and abstract aspects.

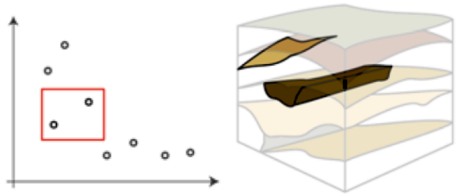

Figure 1: *Selection in a scatterplot (abstract source domain) causes highlighting of the corresponding objects in the 3D view (spatial target domain).*

In summary, the visualization literature describes the design space of combined visualizations at a visual [JE12] and an interaction level [BVG10, BRR03], but, save for the work of Balabanian et al. [BVG10], does not differentiate between spatial and abstract facets of the individual visualizations. This is a void that is worth to be explored.

3. Model-Based Taxonomy

3.1. Overview

We define integration as the functional linking of two visualization components where each component stems from a different visualization domain, i.e., the abstract or the spatial domain (as defined in Sec. 3.2). The domain that a user interacts with is referred to as the *source domain*. The domain that is affected through the integration is referred to as the *target domain*. If both components belong to the same domain, i.e., source and target domain are the same, we speak of coordination of components rather than integration. The selection of objects in a scatterplot (abstract source domain) that causes highlighting of the corresponding objects in a three-dimensional view (spatial target domain) (Fig. 1) is therefore an integration, for instance.

As the key idea of the taxonomy, different types of integration can be discriminated based on which visualization components are combined in the source and target domain.

We describe a visualization by three components: the Data component, the Visual component, and the Navigation component. These components represent a high-level view of the visualization pipeline. In our taxonomy, components can be combined like building blocks, in order to form an integration. Simple integration types consist of only two components. However, also more complex combinations are possible (see Sec. 4.4).

Integration is always triggered through user interaction. In the aforementioned example, the selection of objects in a scatterplot corresponds to an interaction on the abstract domain's Data component. The integration affects the Visual component in the spatial (target) domain by highlighting the selected objects in the 3D view.

Each component has certain input and output modalities that allow for certain types of interaction. The Data component handles, for instance, the selection of data entries, while the Visual component is responsible for visually highlighting (encoding) selected entries.

As interaction is a well studied topic in the visualization community, we rely on established definitions, i.e., the user intents by Yi et al. [YaKSJ07], for describing the types of interactions supported by the individual components (Fig. 2). Brehmer and Munzner [BM13] give a good overview of other works on interaction terminology, as well as a comparison of definitions that are equivalent to the ones of Yi et al.

3.2. Definition of Domains

For visualization in the **spatial domain**, data points are mapped to their inherent positions in three-dimensional space, e.g., volume- and flow-visualization, real-time rendering, or GIS. Mapping to two-dimensional space is also considered as spatial, if the third dimension is negligible, e.g., slicing in volume visualization, 2D maps in GIS, certain flow visualization scenarios. The mapping from data to 2D or 3D space is therefore inherent [TM04].

We define the **abstract domain** as encompassing all types of visualizations where the spatialization of the data's representation is chosen [RTM*03]. Explicit spatial references of data visualized in this domain are either missing or ignored. Temperature in a climate simulation, for example, can be visualized in an abstract context as a histogram or in its inherent spatial context at the position in the volume where it was measured. Depending on the data type, representations in the abstract domain may include multivariate visualizations (e.g., parallel coordinates, glyphs), hierarchical visualizations (e.g., TreeMaps [Shn92]), graph visualizations, and others (e.g., text visualizations).

3.3. Notation

We abbreviate an integration between two visualization components by their initial letters, joined by an arrow. The example from Section 3.1, is therefore denoted as $D{\to}V$. The integration direction, e.g., spatial to abstract or abstract to spatial, is indicated in the subscript: $D_s{\to}V_a$, $D_a{\to}V_s$, and $D_{a/s}{\to}V_{s/a}$ for bidirectional integration.

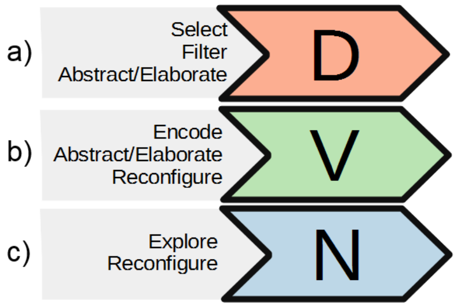

Figure 2: *Visualization Components and their supported user intents: a) Data, b) Visual, c) Navigation.*

3.4. Visualization Components

We base our model components on the visualization pipeline, since it is an established concept for describing the individual stages of a visualization. The pipeline is typically described by Original Data, Processed Data, Mapping, Rendering, and Image stages [FH09, HM90]. For a more streamlined model, we summarize the Data related stages (Original Data, Processed Data) into the Data component, and the Image related stages (Mapping, Rendering, Image) into the Visual component.

In order to encompass not only interaction with the visualization pipeline stages but also with the actual view on the produced 2D or 3D representation, some models incorporate an additional View Transform (Navigation) stage [BRR03, CR98]. We also include Navigation as a proprietary Visualization component, as it supports interactions that are not directly carried out on the Data or the Visual component.

Interactions are carried out *directly or indirectly* [Rob07] on a component in order to fulfill supported user intents. Indirect interaction is handled via menus or widgets (e.g., sliders and buttons) and can happen onscreen and off-screen. Direct interaction takes place directly on the elements of a visualization (e.g., brushing points in a scatterplot) and can therefore only happen onscreen.

3.4.1. Data Component

The Data component (Fig. 2 a) handles the question of *what* to visualize [FH09]. It comprises all parts of a visualization that are directly related to the data which the visualization is based on (data acquisition, enhancement, processing, and filtering) [FH09].

Interaction with the Data component supports the user intents of *select, filter*, and *abstract/elaborate* [YaKSJ07]. **Select** marks interesting data for further examination, while **filter** removes data according to user specified conditions. **Abstract/elaborate** corresponds to the aggregation of data, as well as the derivation of new data.

As an integration *source* (D→X), the Data component can supply other components with information on which data they should process. As an integration *target* (X→D), the Data component can receive information on how to process existing data as well as new data that was generated by other components.

3.4.2. Visual Component

The Visual component (Fig. 2 b) is concerned with the question of *how* to visualize the supplied data. It comprises all parts of a visualization responsible for generating its final output image (mapping, rendering, image stage) [FH09].

The visual component defines, which data attributes are mapped to which visual variables (e.g., color, shape, or size [WGK10]) in order to determine the visual appearance of a data object. While the abstract domain offers the full spectrum of visual variables for conveying information, the spatial domain is restricted. As the spatial mapping is inherent, manipulation of the spatial attributes of an object, such as the position of a voxel in a computed tomography dataset, can be misleading. However, in some cases, the shape and size of 3D objects can be exaggerated to convey information.

Interaction with the Visual component supports the user intents of *encode, abstract/elaborate*, and *reconfigure* [YaKSJ07]. **Encode** corresponds to assigning visual variables to data attributes, e.g., changing the color of data points in a flow visualization in dependence of their velocity. **Abstract/elaborate** (a/e) enables the users to add or remove detail from a visualization, e.g., by encoding more or fewer attributes in a glyph. A/e also corresponds to image processing methods that derive new data from visual attributes, like visibility. **Reconfigure** is used to gain another perspective on the displayed data, e.g., by changing the order of axes in a parallel coordinates plot to reveal hidden patterns, or by exploding a view in a spatial visualization to avoid occlusion.

As an integration source (V→X), the Visual component provides the target domain with derived information from image processing, as well as, information from the mapping function, i.e., which *visual* attributes are being mapped to which *data* attributes. As an integration target (X→V), the Visual component can give visual feedback to interactions in the source domain, e.g., by changing the color or transparency of related input information.

3.4.3. Navigation Component

The Navigation component (Fig. 2 c) is responsible for changing the viewing position and/or direction on the visualized data. It is simultaneously concerned with the questions of the Data and the Visual components, i.e., what to see (view port) and how to look at it (viewing distance, and angle) but without directly affecting the Data or Visual components. Especially in the spatial domain, Navigation is an essential component, as due to the size of a scene or due to the (self-)occlusion of objects not all relevant data can be displayed simultaneously.

Navigation of a visualization supports the user intents of *explore* and *reconfigure*. **Explore** corresponds to interactions like panning a network graph, or flying through a 3D scene. **Reconfigure** corresponds to view rotations in the spatial domain, e.g., rearranging a view on a volume through rotation.

As an integration source (N→X), i.e., active navigation, the Navigation component can supply the target components with updates about its viewing position or direction. This information can then be used, for instance, to steer data aggregation or the level of detail in the visual mapping. As an integration target (X→N), i.e., passive navigation, the Navigation component can update the viewing distance or direction in relation to information supplied by the source domain, for instance, by transforming the camera so that it encompasses specified data points.

4. Integration Techniques

Depending on its inherent properties, each component has its own input and output modalities, i.e., it can supply specific information to or process specific information from other components when integrated.

The descriptions that we give in this section, represent the state-of-the-art in integration techniques that we encountered during our literature research. However it does not represent a complete listing of all possible types of integration. Instead, the descriptions should provide the reader with the knowledge on how to devise the required type of integration for a given task. The visualization model that our taxonomy is based on is capable of describing the coordination and integration of visualization components, i.e., linking of components within the same domain as well as across domains. However, we focus on the discussion of examples for integration techniques. Still, each principle that we present in this section remains also valid for the coordination of visualization components.

In the following, we cluster component integration into three categories that represent their high level functionality: Data Operations, Data Indication, and Visual Consistency.

4.1. Data Operations

Data Operations describe all integration types that affect the Data component in the target domain. Data operations can be categorized into direct data manipulation ($D \to D$), data derivation from visual or data attributes ($V \to D$, $D \to D$), and data aggregation ($D \to D$, $V \to D$, $N \to D$).

D→D enables techniques where operations on the Data component in the source domain can be reflected on the Data component in the target domain.

Data manipulation: The visualization of traffic trajectory data by Wang et al. [WYL*14] lets users load additional traffic streams into the spatial view by brushing the abstract representations in a histogram ($D_a \to D_s$). The tool for geographical data analysis by Turkay et al. [TSH*14] lets the user issue spatial queries by drawing a path on a map. Abstract data that correspond to the path positions is then loaded into a graph matrix for further analysis ($D_s \to D_a$).

Data aggregation: In the visualization of fiber tracts by Jianu et al. [JDL09], the manipulation of clusters in a dendrogram is reflected in the change of the bundling of 3D fiber tracts ($D_a \to D_s$).

Data derivation: In their visualization of mobility of public transportation systems, Zeng et al. [ZFA*14] generate an abstract isoflow tree of traffic data through selection of a spatial starting point on a traffic map ($D_s \to D_a$).

V→D enables techniques where the result of image processing operations in the source domain can extend the Data component in the target domain for further processing.

Data aggregation: Bruckner and Möller [BM10] developed a tool for *visual parameter steering* that supports artists in designing complex visual effects based on particle simulations, such as fire or smoke. After running numerous particle simulations, the results are clustered based on their visual appearance ($V_s \to D_a$).

N→D: enables techniques where navigation in the source domain can steer the Data component in the target domain. Data in relation to the positional or directional updates, can be selected, filtered, or aggregated in the target domain.

Chang et al. [CWK*07] use navigation in a 3D view to control the clustering of demographic data, which is in return visualized in a matrix view and a parallel coordinates plot ($N_s \to D_a$). This type of integration occurs often in combination with N→V (see Sec. 4.2 and Sec. 4.4).

4.2. Data Indication

Data indication encompasses integration types that highlight (indicate) data objects in the target domain that are related to interactions in the source domain. By visually connecting target and source domain through data indication, orientation between abstract and spatial views is facilitated.

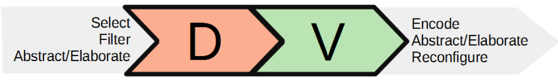

Figure 3: *The integration of two components supports the combination of the respective user intents.*

D→V encompasses techniques where operations on data in one view are visually indicated in another view (Fig. 3). Brushing & linking is a common example for this type of integration. In WEAVE [GRW*00], for instance, abstract visualizations are used to highlight features in 3D volumes. The volume can also be brushed for highlighting the corresponding data point in the abstract views ($D_{a/s}$→$V_{s/a}$).

Since in complex 3D visualizations objects can be occluded or lie outside the view frustum, it can be challenging for users to locate the 3D representations of data that has been selected in an abstract view. Berge et al. [SzBBKN14], for instance, apply a special render technique to make volume segments visible after they have been selected in the abstract domain (D_a→V_s).

Another common application of *D→V* is mapping data values to visual attributes. For many applications, D_a→V_s is essential to explore the spatial distribution of abstract values. In scientific visualization, for instance, it is common to use a transfer function for mapping color or transparency values to abstract or spatial data attributes, such as temperature values in a volumetric data set of an engine block (D_a→V_s) [MFNF01]. Jianu et al. [JDL09] encode the spatial similarity of brain fiber tracts in a color that is shared among spatial and abstract fiber tract representations.

D→N encompasses techniques where the data component supplies the navigation component with information, e.g., about which objects are selected. The navigation component in the target domain then transforms the selected objects into the view, in the sense of *guided navigation*. Guided navigation can help to alleviate issues of localization and occlusion in both domains. A technique by Viola et al. [VFSG06] selects an optimal viewpoint from pre-computed camera positions for a specified volume segment (D_a→N_s).

The inherent complexity of 3D spatial data (visibility, orientation) poses a challenge on finding a proper metric for determining if a viewpoint is optimal. In spatial to abstract integration, the situation is simpler. In BrainGazer [SBS*13], for instance, a list view automatically scrolls to the entry of a segment that has been selected in the 3D view (D_s→N_a).

N→V can change the displayed visual information based on user navigation (similar to *N→D*). In their comparative blood flow visualization, van Pelt et al. [vPGL*14] annotate the spatial vessel visualization with abstract glyphs about local blood flow information. Zooming in the spatial view changes the type of visualization in the annotations according to the distance or available screen space (N_s→V_a).

N→N corresponds to synchronized navigation, i.e., navigational slaving [WBWK00], across domains. It allows users to simultaneously explore data in both domains.

To enable an *N→N* integration, the respective views need to have at least one dimension in common. In *Biopsy Planner* [HMP*12], for instance, users can specify a needle pathway into the brain. This opens a line graph showing the distance to the closest blood vessel along this needle pathway. Consequently the slice views showing the pathway and the line graph share a dimension. Users can now navigate along the x-axis of the line graph, which adapts the slicing position of the slice views (N_a→N_s).

V→N encompasses techniques where results from image processing of the Visual component are used for steering the Navigation component. Viola et al. [VFSG06] apply this technique in volume visualization by transforming the camera to the optimal viewing position of an occluded object.

4.3. Visual Consistency

Visual Consistency encompasses integration types that uphold the *Rule of Consistency* [WBWK00] in terms of visual appearance. Wang Baldonado et al. state that visual consistency facilitates the use of coordinated multiple views by making comparisons easier.

V→V enables techniques that visually link objects from the source domain to their counterparts in the target domain, e.g., by using the same visual mapping for the same data attributes in both domains. A recent example is the focus-and-context visualization by Berge et al. [SzBBKN14] where histogram bars representing volume segments and the actual segments share the same color across domains ($V_{a/s}$→$V_{s/a}$).

Another application of V→V is the usage of rendered (screen space) information from the source domain as input, e.g., for texture mapping in the target domain. In their visualization of sparse traffic trajectory data, Wang et al. [WYL*14] show overlays with abstracted traffic flow data on a 2D street map (V_a→V_s). NeuroLines [AABS*14] displays 3D volume renderings as annotation of an abstract neural pathway representation (V_s→V_a).

4.4. Integration of Multiple Components

In some scenarios it makes sense that an interaction with a component has multiple parallel or sequential effects, i.e., requires the integration of multiple components. An example for *parallel integration* is integrated semantic zooming. Here the distance to a data object determines the aggregation (N_s→D_a) and the visual mapping (N_s→V_a) of the data representations in a target domain (e.g., applied by Chang et al. [CWK*07]). *Sequential integration* is a chain of simple integrations where each target component becomes the source for the next component in the chain. On a technical level, brushing&linking can be described as a sequence of, for instance, D_a→V_a→V_s.

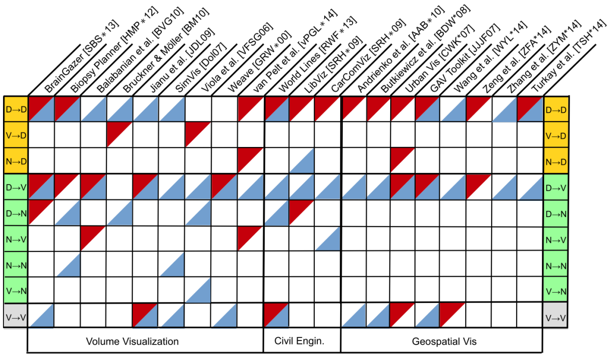

Table 1: *A comparison of integrated interactions derived from the cited literature in this paper. The different publications, listed as columns, are grouped by field. The rows represent nine types of integration, which are grouped according to the categorization shown in Section 4 - Data Manipulation (yellow), Data Indication (green), Visual Consistency (gray). Blue arrows designate abstract-to-spatial, and red arrows spatial-to-abstract integration.*

5. Discussion

5.1. Model Application

In Table 1 we compare the integrated interactions that we derived from the cited literature. The different publications, listed as columns, are grouped by field - Scientific Visualization, Civil Engineering, and Geospatial Visualization. The rows represent nine common integration types, which are grouped according to the categorization shown in Section 4 - Data Manipulation (yellow), Data Indication (green), Visual Consistency (gray). To further illustrate the applicability of our model, we analyze two systems as representative examples of the two dominant integration directions.

SimVis [Dol07] is a framework for the interactive visual analysis of large, multi-dimensional flow data that result from Computational Fluid Dynamics (CFD) simulations. Multiple abstract views, such as scatterplots, histograms, or parallel coordinates, enable the exploration of the simulated flow attributes. The abstract views are linked to a spatial view that displays a three-dimensional visualization of the flow data. In SimVis, the abstract views are used to explore the spatial view, i.e., to highlight patterns in the volume data that could be of interest. The dominance of interaction originating from the abstract domain is also reflected in the relation of blue to red cells in the respective column of Table 1. Integrated interaction mainly takes place on the abstract Data and Visual components, and affects mainly the Data and Visual components in the spatial domain.

UrbanVis [CWK*07] is a visualization tool for the exploration of multi-dimensional data in an urban context. The tool provides a 3D view for the spatial exploration of an urban environment, and an abstract view for exploring the multi-dimensional information that is associated with spatial clusters in the 3D scene. The scenario here is the opposite case to the previous example, i.e., the spatial view is used to explore the abstract data. This manifests itself in a 4:1 ratio of red to blue cells in the respective column of Table 1. Interaction in the spatial view determines, what data ($\rightarrow D$) is chosen and how ($\rightarrow V$) it is displayed in the abstract view. The abstract view itself can then be explored independently, with no implications on the spatial domain.

5.1.1. Integration Patterns

The dominant integration direction in the discussed applications depends on the task that a user should be able to achieve. The task also determines the role that the source and target domains will assume.

In SimVis, the task is to analyze simulation results, e.g., in order to find anomalies in the measurements. In a pattern that can be described as *"Explore & Feedback"*, the source domain is used for data exploration, while the target domain gives visual feedback in order to support the interaction in the source domain. This pattern is, for example, applied in volume visualization applications where abstract representations are selected and mapped to 3D spatial glyphs. In Table 1, this is reflected in the dominance of abstract-to-spatial integration types (blue triangles) for volume visualization applications.

In UrbanVis, the user's task is to analyze census data in order to draw conclusions about relationships between living conditions and locations. In a *"Derive & Analyze"* pattern, the source domain is used for dynamic data extraction and derivation, whereas the target domain is responsible for visualization and analysis of the derived abstract data. This pattern is employed in applications that emphasize spatial exploration. Here the abstract view holds additional information about user defined regions in the spatial scene. This results in a strong integration from the spatial to the abstract domain, as it can be found in applications from civil engineering and geospatial visualization. In Table 1, this is reflected in the first row, by the strong spatial-to-abstract integration (red triangles) for direct data-to-data integrations ($D \rightarrow D$).

If integration techniques are applied so that they complement each other, we speak of a *feedback loop* [War00] or *balanced integration* across domains. One or both of the "Explore & Feedback" and "Derive & Analyze" patterns are employed in each integration direction, enabling users to explore the spatial and the abstract domain in a back and forth fashion.

5.1.2. Visualization Components

The suitability of a visualization component as an integration source or target depends on the component's input and output modalities, i.e., the user intents that they support in terms of interaction, and the information that they can receive, process, and output in order to support these intents.

Since the user typically interacts directly with the data ($D{\rightarrow}X$) or navigates a view on the data ($N{\rightarrow}X$), D and N are common as integration source. Interactions on the data prevail in Table 1 ($D{\rightarrow}D$ and $D{\rightarrow}V$). From a technical perspective, it does not matter if the input information that a component receives comes directly from a user's interaction or from the output of another component. A component that is a suitable integration source therefore also represents a suitable integration target.

From a technical perspective it does not matter if the incoming information comes directly from a user's interaction or from the output of another component. A component that is a suitable integration source therefore also represents a suitable integration target.

The Visual component is well suited as an integration target since it can transform the incoming information into visual feedback. Table 1 clearly shows the strong occurrence of the Visual component as integration target in the analyzed applications, especially for interactions on Data in the source domain ($D{\rightarrow}V$).

Direct interactions with the Visual component as integration source, however, are less frequent, except for $V{\rightarrow}V$. Here the Visual component is used in making discoveries through manipulation of the mapping, e.g., via transfer functions. The output of the Visual component, however, is not easily transformed into input for other components. A suitable integration target therefore is not automatically a suitable integration source.

The goal of a system designer should be to pick the type of integration that supports the user intents required for fulfilling a given task.

5.2. Model Validation

Beaudouin-Lafon describes three metrics by which interaction models can be evaluated [BL04]: descriptive power, evaluative power, and generative power.

We argue that **descriptive power** is given, since our taxonomy's model is based on the general visualization pipeline. This means the taxonomy can describe the coordination and integration of visualization components regardless of visualization method and visualization domain. Moreover, the taxonomy is not restricted to any particular application domain but may be applied to all types of data, for example, from computational fluid dynamics, or urban planning.

With the model application in Section 5.1, we aim to demonstrate that **evaluative power** is given. Different systems can be compared by analyzing their integration patterns. Alternative implementations can be suggested after identifying the user intents that are required to fulfill a certain task as well as the compatible integration types.

We argue that with the disclosure of the integration design space, **generative power** is given, as well. By describing the properties of individual visualization components in each domain and how they can be combined with each other, the design space is laid open to the system designer.

6. Conclusion & Outlook

The integration of visualization domains enables efficient exploration of data with inherent spatial and abstract attributes. In this work, we present a model that describes the design space that results from the combination of visualization components from the spatial and the abstract visualization domain. We showed that our taxonomy can be applied to draw conclusions and identify patterns and correlations across applications that employ composite visualizations with abstract and spatial facets.

There is a need to focus on the topic of integration of two heterogeneous visualization domains (i.e., spatial and abstract) due to the lack of related research in this area. Our taxonomy represents a holistic design space for linking visualizations, regardless if within the same domain, across different domains, within a single view, or across multiple views. This makes it possible to analyze coordination and integration within the same theoretical framework and differentiate between them without excluding one or the other.

In future work, we would like to apply this taxonomy to a broader suite of literature in the scope of a state-of-the-art report. By providing a stable base for the development of novel ideas, we hope to contribute to the understanding of this branch of research that, while finding more and more applications in today's scientific community, has never been clearly discussed and structured on a detailed enough level.

Acknowledgments

This work has been supported in the scope of the FWF-funded project P24597-N23 (VISAR) as well as by the Austrian Funding Agency (FFG) within the scope of the COMET K1 program. Special thanks go to J. Waser, A. Konev, and M. Sedlmair for their valuable comments.

References

[AAB*10] ANDRIENKO G., ANDRIENKO N., BREMM S., SCHRECK T., VON LANDESBERGER T., BAK P., KEIM D.: Space-in-time and time-in-space self-organizing maps for exploring spatiotemporal patterns. In *Computer Graphics Forum* (2010), vol. 29, pp. 913–922. 7

[AABS*14] AL-AWAMI A. K., BEYER J., STROBELT H., KASTHURI N., LICHTMAN J. W., PFISTER H., HADWIGER M.: Neurolines: A subway map metaphor for visualizing nanoscale neuronal connectivity. *Visualization and Computer Graphics, IEEE Transactions on 20*, 12 (2014), 2369–2378. 5

[BDW*08] BUTKIEWICZ T., DOU W., WARTELL Z., RIBARSKY W., CHANG R.: Multi-Focused Geospatial Analysis Using Probes. *Visualization and Computer Graphics, IEEE Transactions on 14*, 6 (2008), 1165–1172. 7

[BL04] BEAUDOUIN-LAFON M.: Designing interaction, not interfaces. In *Proceedings of the working conference on Advanced visual interfaces* (New York, NY, USA, 2004), pp. 15–22. 7

[BM10] BRUCKNER S., MÖLLER T.: Result-driven exploration of simulation parameter spaces for visual effects design. *Visualization and Computer Graphics, IEEE Transactions on 16*, 6 (2010), 1468–1476. 4

[BM13] BREHMER M., MUNZNER T.: A multi-level typology of abstract visualization tasks. *Visualization and Computer Graphics, IEEE Transactions on 19*, 12 (Dec 2013), 2376–2385. 3

[BRR03] BOUKHELIFA N., ROBERTS J., RODGERS P.: A coordination model for exploratory multiview visualization. In *Coordinated and Multiple Views in Exploratory Visualization, International Conference on* (2003), pp. 76–85. 2, 3

[BVG10] BALABANIAN J.-P., VIOLA I., GRÖLLER E.: Interactive illustrative visualization of hierarchical volume data. In *Proceedings of Graphics Interface 2010* (2010), Canadian Information Processing Society, pp. 137–144. 2

[CR98] CHI E. H.-H., RIEDL J.: An operator interaction framework for visualization systems. In *Information Visualization, 1998. Proceedings. IEEE Symposium on* (1998), pp. 63–70. 3

[CWK*07] CHANG R., WESSEL G., KOSARA R., SAUDA E., RIBARSKY W.: Legible Cities: Focus-Dependent Multi-Resolution Visualization of Urban Relationships. *Visualization and Computer Graphics, IEEE Transactions on 13*, 6 (2007), 1169–1175. 4, 5, 6

[Dol07] DOLEISCH H.: SIMVIS: interactive visual analysis of large and time-dependent 3D simulation data. In *Proceedings of the 39th conf. on Winter simulation* (2007), pp. 712–720. 1, 6

[FH09] FUCHS R., HAUSER H.: Visualization of Multi-Variate Scientific Data. *Computer Graphics Forum 28*, 6 (2009), 1670–1690. 1, 2, 3

[GRW*00] GRESH D. L., ROGOWITZ B. E., WINSLOW R. L., SCOLLAN D. F., YUNG C. K.: WEAVE: a system for visually linking 3-D and statistical visualizations, applied to cardiac simulation and measurement data. In *Proceedings of the conference on Visualization '00* (2000), pp. 489–492. 5

[HM90] HABER R. B., MCNABB D. A.: Visualization idioms: A conceptual model for scientific visualization systems. *Visualization in scientific computing 74* (1990), 93. 3

[HMP*12] HERGHELEGIU P.-C., MANTA V., PERIN R., BRUCKNER S., GRÖLLER E.: Biopsy planner–visual analysis for needle pathway planning in deep seated brain tumor biopsy. In *Computer Graphics Forum* (2012), vol. 31, pp. 1085–1094. 5

[HWM*06] HAUSER H., WEISKOPF D., MA K.-L., VAN WIJK J. J., KOSARA R.: Scivis, infovis bridging the community divide. *IEEE Visualization (Panel)* (2006), 52–55. 1

[JDL09] JIANU R., DEMIRALP C., LAIDLAW D. H.: Exploring 3d dti fiber tracts with linked 2d representations. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (2009), 1449–1456. 4, 5

[JE12] JAVED W., ELMQVIST N.: Exploring the design space of composite visualization. In *Pacific Visualization Symposium (PacificVis), 2012 IEEE* (2012), pp. 1–8. 2

[JJJF07] JERN M., JOHANSSON S., JOHANSSON J., FRANZEN J.: The gav toolkit for multiple linked views. In *Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07. Fifth International Conference on* (2007), IEEE, pp. 85–97. 7

[KH13] KEHRER J., HAUSER H.: Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey. *Visualization and Computer Graphics, IEEE Transactions on 19*, 3 (2013), 495–513. 1, 2

[MFNF01] MANSSOUR I. H., FURUIE S. S., NEDEL L. P., FREITAS C. M. D. S.: A framework to interact and visualize with multimodal medical images (st). In *Volume Graphics* (2001). 5

[Rob07] ROBERTS J.: State of the Art: Coordinated Multiple Views in Exploratory Visualization. In *Coordinated and Multiple Views in Exploratory Visualization* (2007), pp. 61–71. 3

[RTM*03] RHYNE T.-M., TORY M., MUNZNER T., WARD M. O., JOHNSON C., LAIDLAW D. H.: Information and Scientific Visualization: Separate but Equal or Happy Together at Last. *IEEE Visualization (Panel)* (2003), 611–614. 1, 3

[RWF*13] RIBICIC H., WASER J., FUCHS R., BLÖSCHL G., GRÖLLER E.: Visual Analysis and Steering of Flooding Simulations. *Visualization and Computer Graphics, IEEE Transactions on 19*, 6 (2013), 1062–1075. 7

[SBS*13] SORGER J., BÜHLER K., SCHULZE F., LIU T., DICKSON B.: neuroMap - Interactive Graph Visualization of the Fruit Fly's Neural Circuit. In *IEEE Symposium on Biological Data Visualization* (2013), pp. 73–80. 5

[Shn92] SHNEIDERMAN B.: Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on graphics (TOG) 11*, 1 (1992), 92–99. 3

[SRH*09] SEDLMAIR M., RUHLAND K., HENNECKE F., BUTZ A., BIOLETTI S., O'SULLIVAN C.: Towards the Big Picture: Enriching 3D Models with Information Visualisation and Vice Versa. In *Smart Graphics*, vol. 5531. Springer Berlin Heidelberg, 2009, pp. 27–39. 1

[SzBBKN14] SCHULTE ZU BERGE C., BAUST M., KAPOOR A., NAVAB N.: Predicate-based focus-and-context visualization for 3d ultrasound. *Visualization and Computer Graphics, IEEE Transactions on 20*, 12 (2014), 2379–2387. 5

[TM04] TORY M., MÖLLER T.: Rethinking Visualization: A High-Level Taxonomy. In *IEEE Information Visualization* (2004), pp. 151–158. 1, 2, 3

[TSH*14] TURKAY C., SLINGSBY A., HAUSER H., WOOD J., DYKES J.: Attribute signatures: dynamic visual summaries for analyzing multivariate geographical data. *Visualization and Computer Graphics, IEEE Transactions on 20*, 12 (2014), 2033–2042. 4

[VFSG06] VIOLA I., FEIXAS M., SBERT M., GRÖLLER E.: Importance-Driven Focus of Attention. *Visualization and Computer Graphics, IEEE Transactions on 12*, 5 (2006), 933–940. 5

[vPGL*14] VAN PELT R., GASTEIGER R., LAWONN K., MEUSCHKE M., PREIM B.: Comparative blood flow visualization for cerebral aneurysm treatment assessment. In *Computer Graphics Forum* (2014), vol. 33, pp. 131–140. 5

[War00] WARE C.: *Information visualization*, vol. 2. Morgan Kaufmann San Francisco, 2000. 6

[WBWK00] WANG BALDONADO M. Q., WOODRUFF A., KUCHINSKY A.: Guidelines for using multiple views in information visualization. In *Conference on Advanced visual interfaces* (2000), pp. 110–119. 5

[WGK10] WARD M., GRINSTEIN G., KEIM D.: *Interactive Data Visualization: Foundations, Techniques, and Applications*. A. K. Peters, Ltd., Natick, MA, USA, 2010. 3

[WYL*14] WANG Z., YE T., LU M., YUAN X., QU H., YUAN J., WU Q.: Visual exploration of sparse traffic trajectory data. *Visualization and Computer Graphics, IEEE Transactions on 20*, 12 (2014), 1813–1822. 1, 4, 5

[YaKSJ07] YI J. S., AH KANG Y., STASKO J., JACKO J.: Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *Visualization and Computer Graphics, IEEE Transactions on 13*, 6 (2007), 1224–1231. 3, 4

[ZFA*14] ZENG W., FU C.-W., ARISONA S. M., ERATH A., QU H.: Visualizing mobility of public transportation system. *Visualization and Computer Graphics, IEEE Transactions on 20*, 12 (2014), 1833–1842. 4

[ZYM*14] ZHANG J., YANLI E., MA J., ZHAO Y., XU B., SUN L., CHEN J., YUAN X.: Visual analysis of public utility service problems in a metropolis. *Visualization and Computer Graphics, IEEE Transactions on 20*, 12 (2014), 1843–1852. 7