Capturing Knowledge During Exploration of Datasets of Small Multiples

Aljaz Kotnik and Graham Pullan

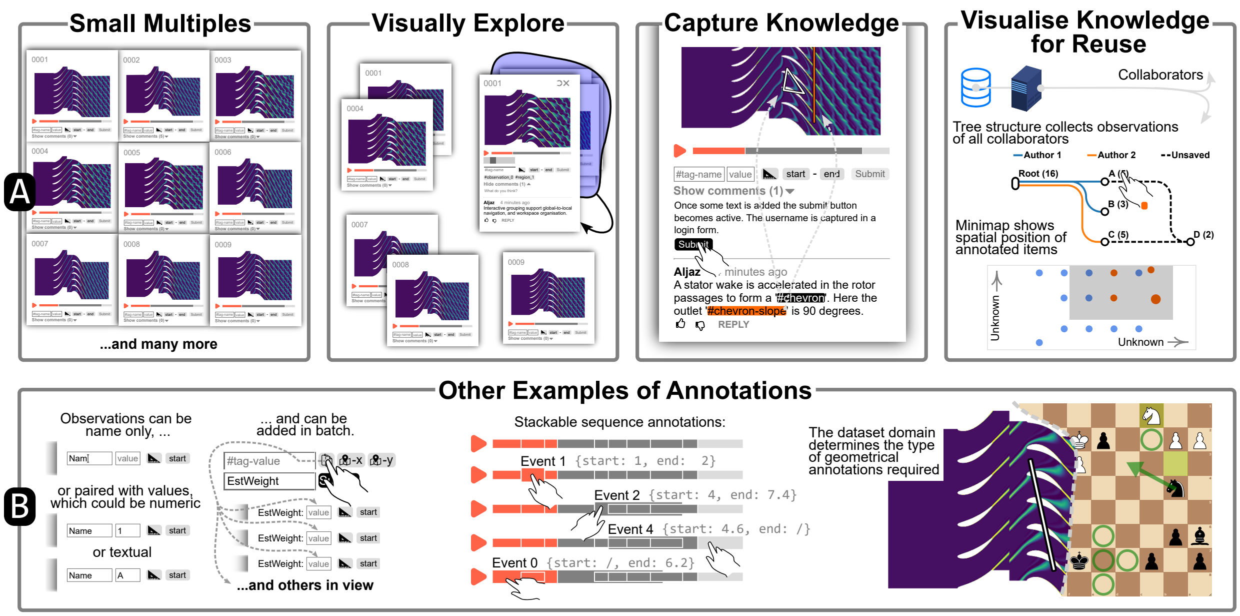

Fig. 1. (A) An aerospace engineering example of 2D unsteady CFD simulation cases [26] displayed as small multiples. Interactions support grouping similar small multiples together, and a spatial encoding metaphor allows the user to capture and use their interpretation of the small multiples to find correlations with the associated metadata. Small multiples support the capture of comments and annotation tags, which are both stored in a central database, and shared with collaborators in real-time. The set of all available tag annotations are synthesised in a Knowledge Tree Visualisation, which gives an overview of all tag annotations made by all collaborators, and also provides navigation between the groups created by the collaborators. (B) Tag annotation types include simple tags, and tag-value, tag-sequence, and tag-geometry pairs.

Abstract—Exploratory visual analysis is a key initial step in the derivation of transferable learning from a dataset. In many cases, the data being studied is best represented by images, or small multiples, that can be manipulated and analysed interactively. During this process, the experienced domain practitioner (a scientist or engineer, for example) is alert to the possibilities that may be uncovered and makes many decisions and hypothesis tests as part of an exploration session. The majority of these ideas are never captured and are lost; they cannot be considered, either by the same individual or a wider team, during future exploration. Where knowledge is captured, it is often recorded in documents that are detached from the data that supports the conclusions drawn. To accelerate exploration and enable knowledge capture, we propose a Visual Analytics environment for analysing image-based ensemble datasets, with image spatial arrangement and piling as the fundamental interactions. The testing of the ideas and hypotheses made by the user is supported using correlations between the spatial arrangement of small multiples and their associated metadata. The environment is capable of capturing observations (such as tags, value estimates, tagged image segments, and time delimited events) and text descriptions, and allows for real-time collaboration of multiple users. A tree visualisation is used to synthesise different types of annotations, support several points of view, and allow navigation between different groups of images created by the user. The utility of this framework for small multiple exploration and knowledge capture is illustrated with example datasets from chess game analysis and engineering simulation.

Index Terms—Scalar Field Data ; Process/Workflow Design ; Domain Agnostic ; Collaboration ; Data Analysis, Reasoning, Problem Solving, and Decision Making ; Feature Detection, Extraction, Tracking & Transformation

• *Aljaz Kotnik and Graham Pullan are with the Whittle Laboratory, Department of Engineering, University of Cambridge, UK. Emails: {ak2164, gp10006}@cam.ac.uk*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

The role of the domain expert in exploratory data analysis is to develop, and begin to test, ideas and hypotheses that are supported by the new data at hand. These ideas, and their connection to the underlying data that inspired them, are precious and hence efficient mechanisms for knowledge capture are vital. The type of exploratory data analysis considered in this paper is the study of an ensemble of images (represented by small multiples). The data could be real-world images (photographs) or any kind of two-dimensional representation of scalar field data. Small multiples arise naturally in many circumstances: data from simulations of candidate designs in engineering, data from

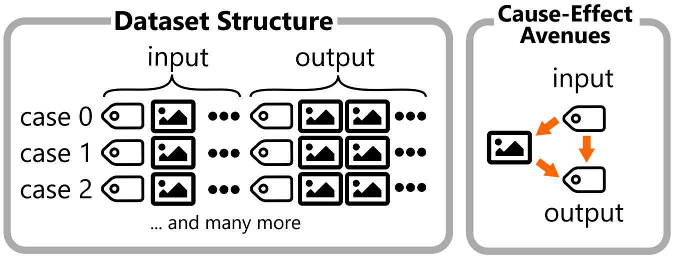

Fig. 2. Ensemble datasets consist of metadata and more detailed image-based data capturing domain specific behaviour. The analyst is interested in finding meaningful correlations and validating them using detailed data.

equipment in the field, from a laboratory study, or from a manufacturing process. In each case, we seek a framework that allows the experienced practitioner both to explore the data and also to capture knowledge (the insights and ideas inspired by the data) in a friction-less manner.

Knowledge is typically captured in documents, such as a scientific publications, or company reports. Although different types of document structure or media (electronic or physical) exist [19], a common attribute is that most documents store the knowledge separate from the underlying data, making follow-up analyses more time-consuming. Interactive data-driven documents [10, 25, 35] aim to keep a live connection between the data, visualisations and the derived knowledge. Editable notebooks allow the user to capture observations as text descriptions in code blocks next to the visualisations. Users can consolidate the observations and insights into knowledge supported by interactive visualisations. However, the text fields are not directly connected to the data they are based on. For example, it is not straightforward to connect text to a feature shown in a subset of a large number of images.

In this work, we present a framework for the exploration of ensembles of small multiples that enables intuitive knowledge capture while maintaining a link to the data. Interactivity is key not only for the manipulation of the small multiples being analysed, but also in the knowledge capture itself. The captured knowledge (ideas and their connection to the data) is stored and presented as a hierarchical tree that fosters both asynchronous and real-time collaboration.

The characteristics of ensemble datasets that influence the analysis workflow and thus the VA (Visual Analytics) environment design are discussed in Sect. 1. An overview of what is meant by ‘knowledge’, and the challenges in capturing, storing, sharing, navigating, and reusing knowledge, is given in Sect. 2. Based on the nature of the dataset, and the knowledge concepts from literature, Sect. 3 codifies the main goals and requirements of the proposed environment, and Sect. 4 discusses specific implementation decisions. The versatility of the proposed VA environment is illustrated with two contrasting examples, from engineering and chess, in Sect. 5.

1 Dataset characteristics

The datasets of interest contain ensembles (typically 100’s) of members, each comprised of image-based data and associated metadata. Metadata typically includes up to 100 parameters [46]. Cause-and-effect avenues exist between all pairs of metadata and image-based data, as shown in Fig. 2. Ensembles of image-based data can be visualised as small multiples.

A practical approach to multimedia ensemble dataset analysis is to initially focus on analysing the set of metadata. Data filtering [6, 7, 38] enables the reduction of the ensemble size to a subset of cases by interactive brushing of metadata plots. Hierarchical filtering allows the user to analyse progressively more detailed aspects of an ensemble dataset. The datasets of interest also extend to time-varying image-based data (e.g. results from computational simulations presented as movies) [26, 39].

The focus of the present work is the initial exploration of the ensemble. At this stage, a domain expert is tasked with analysing the data, identifying significant or unexpected features and their correlation with the metadata, and capturing transferable knowledge. Although, in principle, the interrogation of image-based data can be supported with feature detection algorithms [40], during the exploration phase such features may not have been identified *a priori*.

2 Knowledge concepts

In this section we discuss the knowledge concepts underpinning our work. The two fundamental **knowledge types** are first discussed, followed by the cyclic process of **knowledge creation**. **Annotations** are the key method of connecting knowledge to the data of the small multiples. Finally, having captured knowledge, we also highlight the challenges associated with **knowledge transfer and reuse** and **knowledge visualisation and navigation**.

Knowledge types Two fundamental knowledge types are identified tacit, and explicit knowledge [19]. Tacit knowledge comes from ‘learning by doing’ and can be difficult to represent in words, making the transfer of tacit knowledge challenging. Explicit knowledge can be captured in documents, and is readily transferable. However, explicit knowledge is insufficient without tacit knowledge, and the loss of tacit knowledge could mean that technologies become uninvented [19]. Tacit knowledge that has been transformed into explicit knowledge has been called implicit knowledge [18]. Transfer of knowledge, of any type, is a source of competitive advantage [3, 4, 29].

Knowledge creation Unstructured information becomes knowledge when it is connected into a valid meaning [37, 42]. The scientific method of producing knowledge begins with observations. Prior knowledge is used to formulate hypotheses, and data gathered from experiments is used to confirm or reject the hypotheses by comparing the predictions to the results of the experiment. The findings are then reused as additional observations, or additional knowledge. A similar process can be applied to the data analysis step: the user first observes some features in the data; then, based on prior experience, they make a hypothesis for what the features represent and their correlation to metadata; finally, the user looks for additional data to support or reject the hypothesis. An analyst may have to go through several cycles before they identify and understand all relevant dataset behaviours, and capture them as knowledge.

Annotations An annotation is an “observation, made by exploring a visual representation of data, that is recorded either as text or visual selection (or both)” [53]. The initial observations, the hypothesis, and the outcomes of individual cycles are all knowledge building blocks and should be captured on their own to support others to reuse them in their analyses. Asynchronous interactive collaboration workspace environments capable of supporting discussion and view sharing are social spaces in addition to analytic spaces [23], and can make groups more productive [31].

Annotations can be captured during several steps: data preprocessing, cleaning and exploration [44]. In this paper it is assumed that the ensemble dataset is clean and preprocessed, and just the exploration phase, during which annotation is especially important [45], remains. Annotations can be data-based, or view-based [5]. Data-based annotations are directly linked to individual data items, like temporal annotations of audio/video recordings [22, 24]. View-based annotations require a particular view state [23, 31, 63]. Description annotations can be further classified by the nature of their content [53].

Practical implementations of observations include: tags/labels [8, 48, 55, 62], comments [58], categorical and ordinal tag-value pairs [24], temporal annotations [22, 24], and tag-area pair annotations [55]. Observations can be captured as typed text [58], or as a free-hand sketch [23]. Free-hand sketches provide expressive freedom, while typed annotations allow for easier searching using key words.

Knowledge transfer and reuse occurs between knowledge owners and knowledge seekers. This can be senior-junior employee relationships, or different branches/organisations transferring knowledge. Connecting knowledge seekers with knowledge owners is the first step in knowledge transfer. Social web-based Q&A communities that support knowledge capture, sharing, and reuse, such as Quora [41] and StackExchange [49], are increasingly popular choices for knowledge sharing [54,56]. The user base is prompted by the knowledge seeker-posting a question, and relevant knowledge holders may post responses. The community votes on the quality of the questions and answers, thus curating the content. Knowledge seekers need not be different from knowledge owners - users can answer their own questions. In scientific research, or engineering processes, the same person that poses the question often answers it.

"While most Q&A sites were initially aimed at providing useful answers to the question asked, there has been a marked shift towards question answering as a community-driven knowledge creation process whose end product can be of enduring value to a broad audience" [2]. Users on StackExchange engage more frequently and quickly than users on comparable mailing lists. The key difference is the gamification by awarding users social status through reputation points and badges for participation [54].

The expertise level of the knowledge seeker determines how the repository should be structured for successful knowledge reuse [32]. For example, executive summaries need to communicate the general knowledge, and prevent misinterpretation by focusing only on the important features. Experts, on the other hand, require detailed data (which they may wish to access and analyse themselves), alongside observations captured in work logs, to answer questions they may have. Therefore, expert repositories must contain annotated data, while high level repositories should only contain curated content. The environment proposed in this paper is designed to enhance the data exploration workflow of domain experts.

Knowledge Visualisation and Navigation. "Commercial exploratory data analysis systems that support annotations still present user-authored comments as lists; in a linear and textual form" [62]. The challenge for large repositories of knowledge components is finding the relevant entries, even for members of teams that originally produced them [32]. Unstructured knowledge repositories require the knowledge seekers to engage with them over prolonged periods of time to acquire a sense of all the knowledge. Instead, the knowledge should be structured, visualised, and easily navigable while supporting different points of view, and not silencing minorities [34].

Annotation graphs are a way of visually representing user-created annotations [62], where the annotations are seen as combinations of three components: tags, comments, and data selections. Each of the components is drawn as a node, and nodes based on the same annotation are connected with a line, thus the lines are implicitly encoded by the user. The spatial arrangement of the nodes is determined by the selected layout, and similarity metrics specific to the three annotation component types.

Hierarchical trees can visualise the dataset partitioning into subsets [30,51]. BaobabView [51] allows the user to interact with the decision tree and grow, optimise, and prune the decision tree. Different users may arrive at different hierarchies, and several methods of visualising, combining, and comparing them exist [20]. Similar hierarchical trees can be merged using specially designed node glyphs [12] and this assists a team in reaching an agreed 'common-ground' hierarchy. The order in which annotations are created can also be leveraged. Using tree structures to visualise and navigate along this order aids the user in recalling past explorations faster [21,47,50,63].

3 Goals and Requirements

We present a specification for our Visual Analytics environment to capture knowledge during the exploration of small multiples data as a set of over-arching goals detailed requirements. The two goals are:

The tasks of knowledge creation, capture, and reuse should be integrated to allow the analyst to quickly and effortlessly switch between them.
An existing workflow may be split between several dedicated software suites, causing context switches that can derail the analyst's train of thought [1]. Similarly, reusing knowledge captured in text documents involves manually finding connections from knowledge to the data, and vice versa. Discussing results that are scattered between different software applications is difficult, and "... the analysis process could benefit from a system for sharing, annotating, and discussing the visualized data" [23].

The knowledge created should be captured, visualised on-the-fly, and shared in an engaging, intuitive, and interactive collaborative environment.
Visualising knowledge components (e.g. tags, descriptions, image segmentations) allows the user to find connections between them, and can be used as a knowledge navigation aid. During collaborative work, potentially competing interpretations of the data can arise, and must be supported. Captured knowledge components should be stored in a central shared repository to allow the creation of 'feature libraries', which would allow subsequent re-users to find all knowledge relevant to specific features.

To create an exploratory Visual Analytics environment capable of capturing and sharing knowledge, we identify several specific requirements:

R1 Interactive exploration: analyse qualitative and quantitative differences between ensemble members and find correlations with metadata.

R2 Knowledge capture: provide user interfaces for interactive capture of tags, value estimates, descriptions, and geometric image annotations such as lines and areas.

R3 Knowledge meta-analysis: collect, synthesise, and visualise all available knowledge components, while supporting differing points of view.

R4 Knowledge reuse: use a shared repository to support knowledge component reuse for both asynchronous and real-time collaboration.

4 Design and Implementation

We present the design of our Visual Anlaytics environment in two parts. The first part is the Small Multiples Workspace - where users explore data, develop ideas and test hypotheses that connect these ideas to the metadata parameters. The second part is Knowledge Capture and Management - friction-less annotation of the data and interactive representation of the knowledge captured.

4.1 Small Multiples Workspace

We begin by providing an overview of the Small Multiples Workspace **concept**. The on-screen **layout** is then described, followed by the key interaction and encoding mechanisms of the **spatial metaphor** and **piling**. The Small Multiples Workspace addresses the **(R1)** requirement of Sect. 3.

Conceptual overview Three main approaches have been identified for interactive exploration of visual parameter space analysis of ensemble datasets [46]: 'Trial-and-Error', 'Local-to-Global', and 'Global-to-Local'. Trial-and-error and local-to-global approaches start with analysing a single small multiple. In trial-and-error, the user proceeds to investigate the impact of changing a particular parameter. In local-to-global, the user tries to generalise findings based on one small multiple to other ensemble members. In the global-to-local approach the user first creates an overview of the ensemble, and only later drills down to the details. We envision an analysis process that allows for both the Global-to-Local and Local-to-Global approaches. In the Global-to-Local approach the user first identifies qualitative differences between ensemble members, and separates them into distinct groups. Within the context of a particular group they then employ the Local-to-Global approach by selecting one member, analysing the behavior in detail, and abstracting the findings to other group members. The user can then determine correlations that help to identify underlying cause-and-effect avenues.

The basic questions that can be asked in our proposed Visual Analytics environment are: 'Are there qualitative differences between sets of small multiples?' and 'What are the quantitative differences between

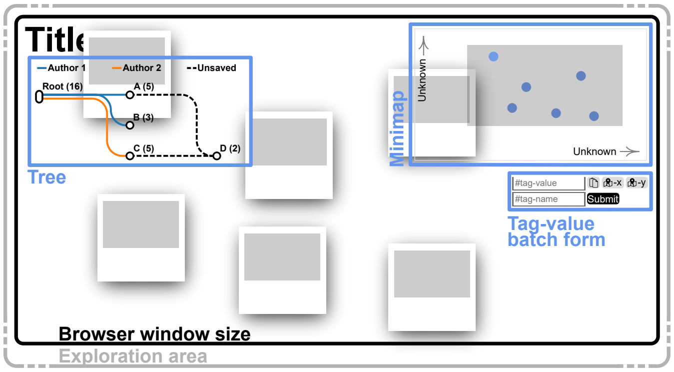

Fig. 3. The proposed VA environment spans the entire available browser window space, and is split between the 'exploration area', and the interactive GUI overlay, the components of which are highlighted in blue. Small multiples can be dragged everywhere in the exploration area, including underneath the GUI components. The minimap helps expand the exploration area beyond the confines of the browser window. The title is also a part of the GUI, but small multiples can be placed on top of it to reclaim its space.

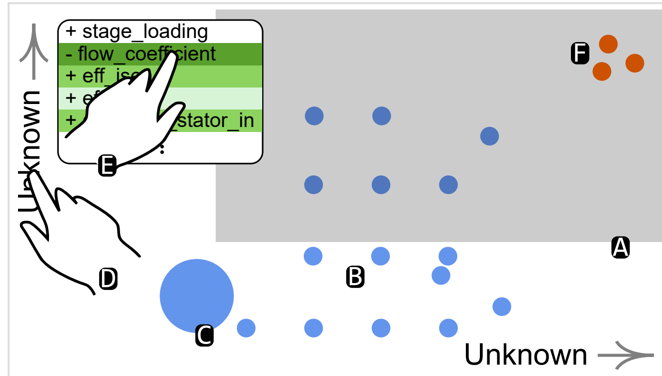

Fig. 4. The minimap shows the global arrangement of all small multiples. The gray rectangle (A) denotes the current view, and can be dragged to change it. Individual small multiples are represented as small circles (B), while groups are circles with an area proportional to the number of members (C). Clicking on the axes labels (D) shows/hides the correlation dropdown menu. Dropdown menu entries visualise the Spearman rank correlation coefficient between specific metadata attributes and the small multiple arrangement in the axes direction. The color encodes the absolute value of the coefficient, and the sign encodes whether it is positive/negative. When mousing over the knowledge tree nodes the corresponding small multiples are highlighted on the minimap (F).

qualitatively similar small multiples?' The Small Multiples Workspace provides two principle interactions to enhance the user's exploration of the data: **piling** allows the user to separate out and encode qualitatively separate groups of small multiples; a **spatial metaphor** underpins the search for correlations between metadata attributes and the user's interpretation of the small multiple visualisations. Both the groupings and the correlations, as well as the expert's associated hypotheses and commentary, are then stored by the Knowledge Capture and Management system, as discussed in Sect. 4.2.

Layout Knowledge visualisaton is often done using a combination of data, knowledge, and navigation views [47,50,62]. The knowledge and navigation GUI components take up space that is valuable for ensemble small multiple visualisation, therefore we embed them on top of the exploration environment, as shown in Fig. 3. A minimap that provides an overview of the exploration area is also implemented. Small multiple dragging is shown on the minimap in real time. A practical example of the full Visual Analytics layout is shown in Fig. 12.

Fig. 3 shows that the full exploration area extends beyond the screen, and its extent is not limited. The user can navigate through the exploration area by zooming and panning on the background whitespace. These spatial navigation interactions affect the size and position of the small multiples, but don't resize or reposition the GUI elements shown in Fig. 3.

The minimap, shown in more detail in Fig. 4, provides an overview of the exploration session, and is continuously updated. The minimap domain is scaled to show all small multiples in the exploration area at all times, regardless of the spatial navigation interactions performed.

The minimap uses icons to visualise the relative positions of small multiples, and a shaded 'current-view' rectangle representing the exploration area currently displayed in the browser window. The current-view rectangle also supports zoom and pan spatial navigation, which is particularly useful when the user wishes to navigate long distances.

Spatial metaphor On-screen arrangement can encode meaning. There are several ways to arrange small multiples on-screen. A popular approach is to use dimensionality reduction techniques, such as t-SNE [52] and UMAP [33], to calculate a $2/3D$ map showing the structure of the data, which can then be used to position the small multiples on-screen. Such spatialisations allow the visualisaton of the data structure. The user can interactively change the spatialisation by changing the underlying ML model parameters and recalculating the embedding. This requires the analyst to understand the model parameters, constrains interactivity, and doesn't allow the analyst to use their interpretation of the small multiples to directly manipulate the arrangement [16].

Semantic interaction [9, 14–16] allows the users to directly interact with the on-screen positions of small multiples, which informs the underlying model, which in turn updates the on-screen arrangement. The user is shielded from interacting directly with the model parameters. The spatial metaphor used is 'near = similar' - the similarity of small multiples positioned by the user depends on the on-screen distance between them. Semantic interaction combines the expertise of the analyst and the power of ML models to accelerate creation of meaningful on-screen arrangements of small multiples.

The spatial arrangement positions can also be considered as direct encodings of the user's interpretation of the small multiples [26]. Each small multiple is associated with its x and y arrangement coordinates, as shown in Fig. 5, which can be used to encode the size of observed features, or more complex estimates, such as weights of depicted objects, by arranging the small multiples into a sensible order along either the x or y axes. Direct spatial encoding is to value capture what semantic interaction is to dimensionality reduction.

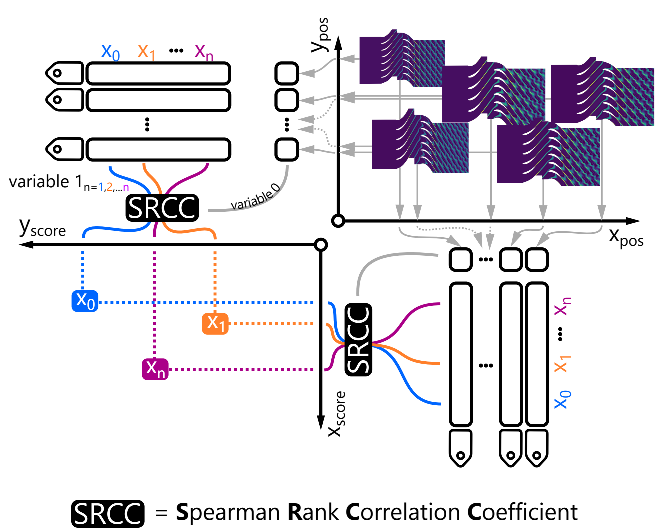

Fig. 5. Direct spatial encoding and correlations adapted from [26]. The Spearman rank correlation is used as it is easy to interpret the results.

The use of statistical models for spatial arrangement requires input data, which will depend on the nature of the ensemble data. Textual ensembles are treated by reducing individual texts into an array of keywords, connecting the small multiples by matching keywords, and assigning weights to the keywords that are adjusted to incorporate user interactions [16]. Image-based ensembles may use the metadata attributes associated with the images [11], neural net interpretations of the images [9], or the rendering data itself [59].

Discovering correlations with observed patterns in the small multiple visualisations is possible using spatial metaphors. The GLEE VA environment [11] displays ordered weights of attributes used as inputs of the underlying arrangement model, allowing the identification of dominant attributes. Direct spatial encoding can be used to correlate the spatial positions to the associated metadata attributes. In our implementation, the correlations of metadata attributes and spatial arrangement are done solely for the small multiples currently in the users view, which allows rapid constraining or relaxing of the exploration context.

Piling Partitioning is a key task in the exploration of ensemble datasets [46], and individual users will develop different and distinct clusters when analysing the same ensembles [13]. Similarly to spatial arrangement, fully automatic [17, 43, 60], mixed initiative [57, 61], and fully manual approaches [28] exist. "Clustering is in the eye of the beholder,..." [17], and allowing the user to directly intervene in clustering is the main advantage of mixed initiative and manual approaches.

Pollux [57] implements a ML-supported mixed initiative clustering. Th clustering is done using k-means, using an initial 'optimal' k that can then be adjusted by the user. Metadata attributes are used to calculate the distances between individual ensemble members, and the weights of attributes in the distance metric are used to capture the user intent. The user can drag and drop items between clusters to reclassify them. Semantic interaction is implemented and updates the arrangement if the user repositions an ensemble member without reclassifying it. Individual ensemble members can be in a single group at any time.

II-20 [61] implements a more manual-oriented approach. The user can define groups through a dedicated menu. Images are arranged in a grid, the user can click on several images to create selections, and add them to a particular group. A statistical model uses the assigned images to calculate recommendations. A fast-forward technique automatically finds the N most likely candidates for a group and adds them to it. A 'tetris' mode provides another way for classifying images one at a time.

The PILING.JS [28] framework supports rendering, interactive arrangement, and grouping (a stack of several small multiples on top of each other is called a 'pile') of several different types of small multiples, such as matrices, images, and static SVG visualisations. The grouping interaction is done using a lasso tool.

"PILING.JS renders items into textures and subsequently operates on these textures during gesturing to decouple the state management and gesture handling from the domain-specific visualizations." [28]. The rendering into textures is done via PixiJS [36], a 2D rendering system based on WebGL.

Small multiple visualisations based on scientific data may require additional on-the-fly interactions, such as changing the colormap and camera point-of-view change, which require re-rendering. A single set of event listeners could be attached to the canvas, and then the interactions could be processed in the background to determine which small multiple they correspond to. Alternately, our approach adds HTML frames to the canvas views, and events are attached to them. This is particularly advantageous when distinguishing between dragging events, and view changing events. The rendering is done using WebGL directly.

Piling small multiples on top of each other results in some small multiples being at least partially occluded, and only one small multiple is shown in its entirety, which serves as the pile cover [28]. Previewing UI elements can allow the cover to be changed interactively [28], as shown in Fig. 6. Custom pile covers can also be specified. For example, group cover variance visualisations [27] allow the user to spot areas with the largest member-to-member differences. Every time a group is

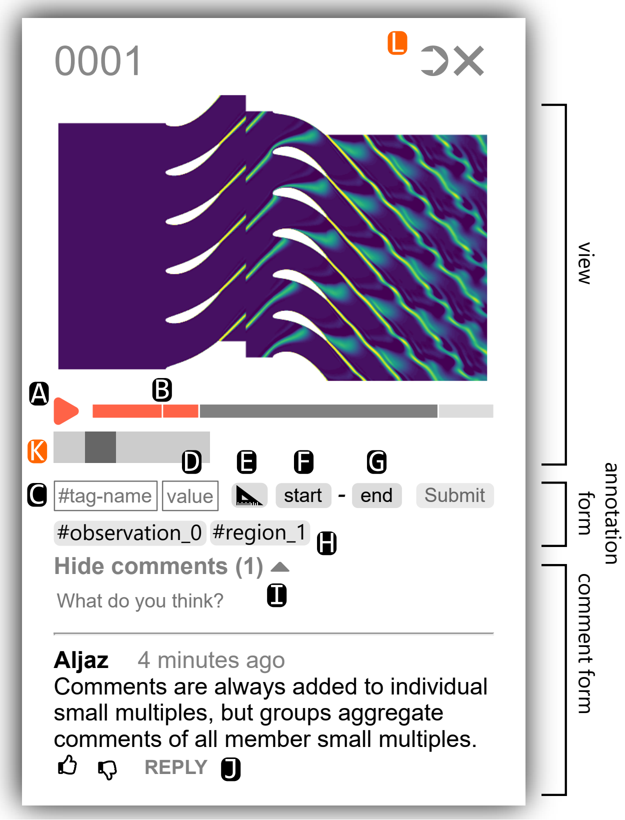

Fig. 6. The HTML small multiple frame supports click-and-drag repositioning of the small multiple. The view module supports panning and zooming, and contains a data appropriate playbar (A), which indicates the start/end times of sequence annotations (B). The annotation form allows the user to input the mandatory tag name (C), and optional value estimate (D), geometry (E) and start/end times (F/G). The added tags appear below (H), and can be used to filter the comments mentioning them in the commenting section. The commenting form (I) submit button appears after text has been entered. Alternately, to post a reply to the existing comment the user can instead submit the comment by pressing the reply button (J). Small multiples representing groups additionally include preview icons (K), and buttons to enter and disband the group.

created, or navigated to, a new small multiple is created to represent it, and the appropriate members are allocated to it. The group is responsible for providing the preview elements, and the interactions to either enter or disband the group, as shown in Fig. 6.

Small multiple component implementation The data visualisations used as small multiples in this paper require additional interactions to be effectively explored. Furthermore, the ensemble datasets may have fundamental differences between them that required specialised interactions with the small multiples. We now discuss how our implementation of a small multiple component supports this.

Small multiples are made up of the visualisation of the image-based data, and the user interfaces required to capture the knowledge created by the user. A HTML frame is used to host the user interface elements, provide the click-and-drag spatial arrangement interactions, and interactions with the view itself. The advantage of HTML frames in our implementation is that similar events used for different purposes (e.g.: zooming/panning of the exploration session vs. zooming/panning of the small multiple view) are distinguished more easily. The layout of the HTML frame is shown in Fig. 6. Furthermore, capturing and displaying knowledge requires interactions. HTML has many standard input elements along with associated interactions readily available. Displaying text via HTML elements allows additional interactions, such as support for referencing previous annotations made.

Visualisations of fundamentally different data types also require different interactions, or even dynamically changing interaction rules, or sets of interactions (see Sect. 5.2). Similarly, geometry capture may also need to support different kinds of geometries (see Fig. 8, and Fig. 13). As all the view interactions are assigned in the view module of the small multiples, only this module needs to be adjusted to support different dataset types.

4.2 Knowledge Capture and Management

The Small Multiples Workspace described in the previous section allows the user to uncover similarities, and correlations between the images and the underlying metadata. We now discuss how this knowledge is captured, managed and maintained. **Annotation types** are presented first, followed by **annotation storage** and **knowledge visualisation**. The Knowledge Capture and Management functionality addresses the **(R2)**, **(R3)** and **(R4)** requirements of Sect. 3.

Annotation types Annotations considered in this paper are split between 'tags', and 'comments', which are all connected to a particular small multiple. The term 'tag' represents an identification of a data-based feature, which can be characterised using a simple tag (name), tag-value pairs, tag-geometry pairs, and tag-sequence annotations. The term 'comments' refers to longer text which can represent hypothesis, questions, or knowledge, and can facilitate discussions through the analysis environment. Comments are directly attached to the small multiple. When several small multiples are grouped, their comments are aggregated in the comment section.

Simple tags allow users to add classifications to small multiples, and allow groups to be captured and maintained. Instead of explicitly saving a group along with its members, the members are each annotated with a tag. Each unique tag represents a possible group, and to recreate it all the small multiples with the particular tag are found and merged into a group on-the-fly. Tags are the basis for all observations, and the user interface does not allow observations without tags to be saved.

Time-varying data may include event sequences that span a fraction of all available time. By capturing these sequences as observations and visualising them on the playbar, the user can navigate back to already-discovered events faster. Sequence annotations associate start/end values with the event tag name. Both start and end values can be associated with a sequence annotation, but it is enough to specify just one (when specifying only the start/end value the other is assumed to be the available maximum/minimum). The process of adding a start value is shown in Fig. 7. Multiple sequences can be captured, and are visualised on the playbar by stacking them on top of each other, starting with the earliest starting annotation at the bottom, and the subsequent events on top.

Geometry annotations allow capture of simple geometries along with the tag name. For $2D$ based visualisations, the geometries of interest are the point, line, and area. A single tool, inspired by the Google Maps measurement tool, has been implemented, and the process of drawing the geometry is shown in Fig. 8.

The geometry can be fully represented by a series of points. When the user pans and zooms the geometry should re-position correctly. When the user selects a point its x and y coordinates are transformed into the data range. The transforming of the point coordinates, and interpreting them for drawing the geometry annotation, are done by the view creation module of the small multiple. During subsequent zooming and panning the same camera transformations can be used to redraw the observation geometry. When adapting the view module for other types of datasets, the tool data transformations and geometry annotation drawing can be customised to fit with the relevant domain, as shown in Sect. 5.2.

Tag-value annotations can be added if the user wishes to capture observations that are directly comparable. The tag name additionally becomes a quasi metadata attribute, and the value is used to quantify the observation represented by the name. Small multiples have a value input form, as shown in Fig. 6. Comparability implies that several small multiples have the same observation to compare, and therefore to accelerate tag-value capture a batch form, shown in Fig. 9, is available on the global GUI, below the minimap. Both text labels, numeric values, and nonmalised on-screen position can be used as values.

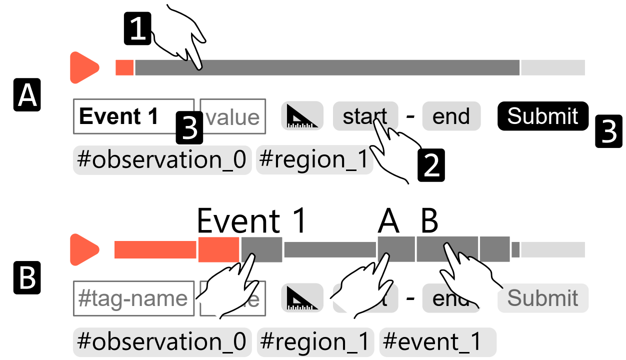

Fig. 7. The playbar outline is shown in light grey, the buffered extent in dark grey, and the current frame in red. To add a sequence annotation (A) the user navigates to a specific frame, and designates it as a start/end time, or both. Only when the annotation is given a tag name, does the submit button become active. After submitting a sequence annotation (B) it appears in the tags below the form, and as a chapter on the playbar. On mouseover the sequence annotation is highlighted, and any intersecting annotations are stacked.

Comments have their own submission form, and can be added either as stand-alone comments, or replies to other comments. The text captured in the comment form can be added as a stand-alone comment by pressing the submit button that appears just below the form. To add the same text as a reply the user must instead click the reply button. A single level of replies is available. To support identification of particularly valuable comments a voting system is available.

The user can reference previously made observations of the small multiple by putting a '#' key in front of the observation tag name. When the comment is displayed the referenced tag names are recognised, and highlighted. Mousing over a sequence or geometrical annotation reference highlights the corresponding visual representations.

Cross-referencing annotations in comments also allows the filtering of comments. For example, by clicking on 'observation_0' the user hides all comments not mentioning it. By clicking on 'region_1' the user additionally shows the relevant comments.

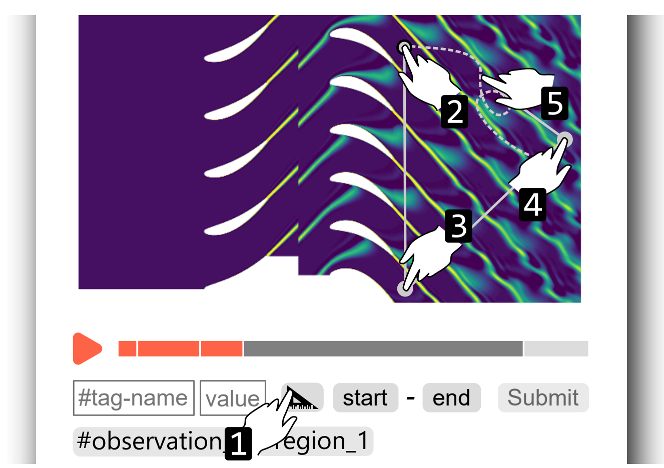

Fig. 8. Geometry selection tool inspired by the Google Maps measurement tool. After activating the selection (1) the user can click on the view to create an origin vertex (2), and then click elsewhere on the view to create additional vertices (3,4). Straight segments are drawn between vertices even if the user follows a curved path between them (5). A line geometry is created by default, but if the user ends the selection at the origin a region selection is made instead.

Fig. 9. Tag-value form is located below the minimap. When the user specifies the tag-*name* in the form (A) an additional form appears on the small multiples (D) allowing the specification of the corresponding tag-*value*. The user can specify a value through the tag-value input (A), and batch assign it to all small multiples in the view (B). Similarly, the user can assign the normalised x and y on-screen positions (C). Only after the name and the value have been assigned to all small multiples in the view can the observation be submitted.

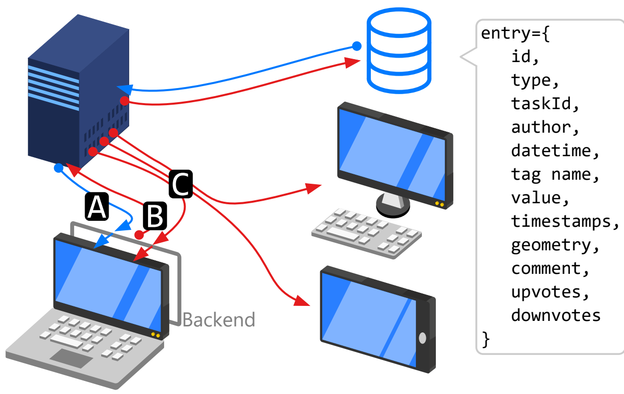

Fig. 10. Network structure, and the data exchange paths. (A) During initialisation the server passes all relevant entries from the database to the client. (B) When the user submits a knowledge entry it is sent to the server. (C) The server stores the entry in a database, and passes the entry to all connected clients - including the originating client, which only updates the view once the entry is received from the server.

All annotations created are attached directly to the relevant ensemble member. Simple tags, and tag-values added via a group small multiple, are added to all the members of the group. Sequence and geometrical annotations, and comments remain specific to a particular ensemble member, and cannot be attached to the whole group at once.

Annotation storage is done using a server running a database (a lightweight SQLite database is sufficient) separate from the small multiple metadata. The central annotation storage allows sharing annotations between collaborators, which in turn allows knowledge reusers to follow the steps of previous analysts, thus absorbing some implicit knowledge. In this way, implicit knowledge [19] can be captured implicitly, as opposed to transforming it to explicit knowledge before storing it.

The annotation database comprises a single table that contains the entries of all different annotation types. Communication between users is achieved using the WebSocket API, which allows the annotations from one user to be pushed to all other users when an annotation is made. The client side requests all available annotations upon startup, then subsequently all individual changes are pushed to keep the records up to date, as shown in Fig. 10.

Geometrical annotation data consists of an array of points representing the vertices. The definition of a point depends on the data domain: for 2D scalar fields locations along two orthogonal axes may be used (e.g. x and y), but in some cases a string representation is advantageous (e.g. Sect. 5.2). Whether the JSON array of points consists of numeric value pairs, or string representations, the array is converted into a single string and stored as text in the database. When the client needs to draw the geometry it is the responsibility of the small multiple component view module to interpret the string data correctly.

The voting system keeps track of the usernames that have upvoted/downvoted a comment in separate arrays. Keeping track of the usernames allows the system to correctly account for when the user wants to change their vote. When users vote on a comment, the

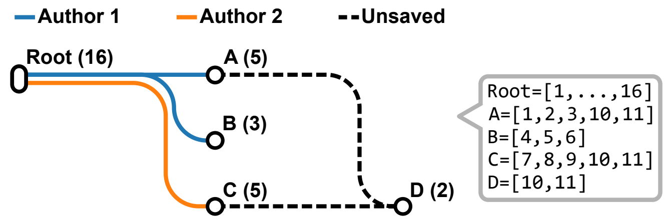

Fig. 11. An example knowledge tree visualisation of available annotations. Nodes represent groups of annotations, and the lines represent superset-subset connections. Colours represent different authors, and the dashed line represents unsaved groups in the current session. Node groups may have several superset and subset groups. Clicking on nodes hides all the subset nodes, and clicking on the markers navigates to the group.

data passed to the server are an id unique to the specific comment, the username, and the change required. The server interprets the request, retrieves the relevant table entry, and adjusts it. The update is then pushed to all connected clients.

Knowledge visualisation. The set of annotations and comments stored in the database constitutes captured knowledge which itself can be visualised; our approach is via a Knowledge Visualisation Tree, and an example is shown in Fig. 11. The two basic tasks for the visualisation tree are (1) the visual representation of all different tag annotation types made by several users, and (2) to allow for navigation between the different groups of annotation tags.

Members of the annotation database are unconnected data-based tags, and every small multiple can have many associated tags. It is possible that hundreds of tags and comments will be added referencing the small multiple dataset. Some tags may be equivalent, and can therefore be grouped. Grouped tag annotations also represent groups created and saved during exploration. A single author may create intersecting tag groups, and other authors may have different views.

In PILING.js [28] the navigation between groups is done via breadcrumbs, which visualise the current exploration branch, and allow navigation along it. Visualising a single branch can constrain the navigation. Furthermore, only groups made on the current branch are retained after breadcrumb navigation, and collaborative exploration is not possible. The Knowledge Visualisation Tree allows multiple branches to be visualised, navigated along, and retained.

The tree is generated dynamically, based on the available tag annotation data. Because the visualisation tree must differentiate between annotations made by different people, groups of equivalent tags are created using tag-name & author combinations of annotations. After all possible groups are created, groups with exactly the same membership are merged. Fig. 11 shows the data underpinning the example tree shown. The resulting groups are arranged into a hierarchy, by positioning each group at least a level below a group that contains it in its entirety. After the levels are assigned, the tree visualisation is dimensioned. Each group is represented by a node, and connected to all groups that contain it as a subset. Each group may have more than one parent group. Solid lines represent saved knowledge, and the colors represent different authors. The dashed line represents the current unsaved exploration session.

The tree supports interactive hiding of branches - clicking on a node hides/shows all subset nodes. When clicking on the node name marker the session navigates to a view within the parent group. When the user mouses over the node name marker, the relevant small multiples are highlighted on the minimap, as shown in Fig. 4 and Fig. 12.

5 Example Applications

The Visual Analytics framework described in the preceding section is now demonstrated by two application examples from contrasting domains: Aerospace Engineering; and, Chess. Interactive demos of these examples are available at `https://aljazkotnik.github.io/spatialknowledge/`.

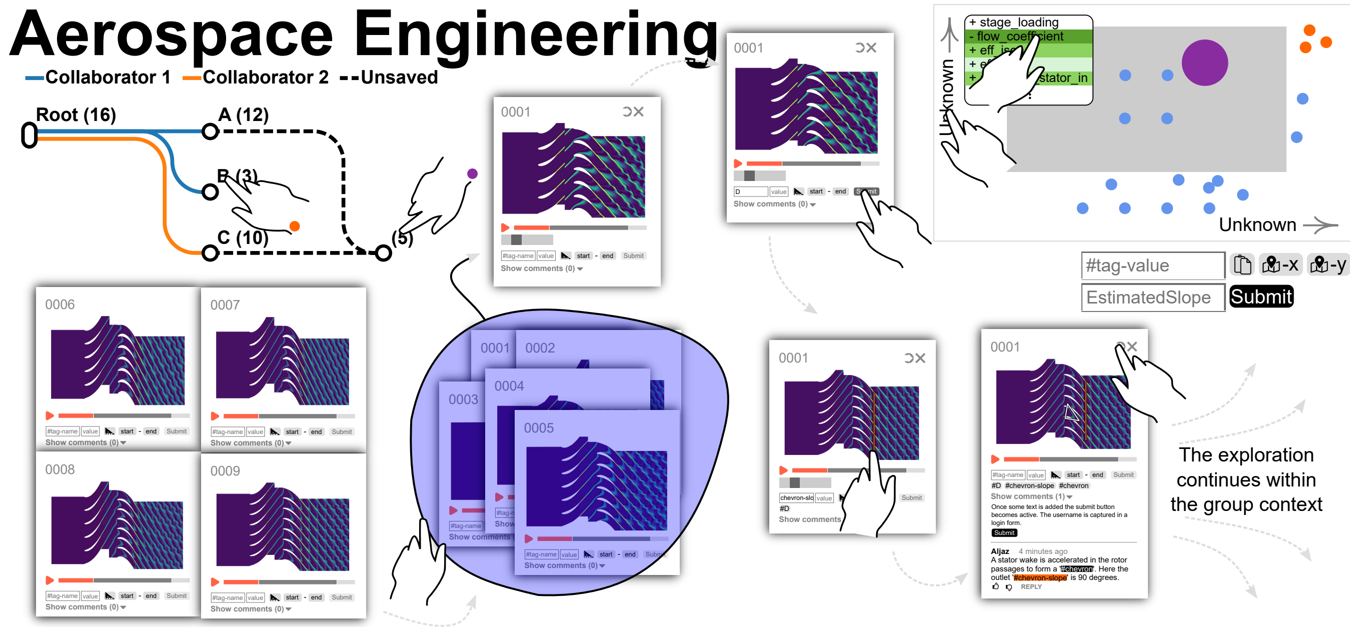

Fig. 12. Example exploration and knowledge capture of an ensemble of 2D unsteady turbine entropy scalar fields. The user can group together designs they deem similar. The observed similarity can be captured by adding a tag to the group, thus saving it. Features in the view can be annotated with geometric annotations, and events can be captured with sequence annotations. When comparing several small multiples the user can spatially encode their interpretation of the behavior, and use the spatial encoding to look for correlations with the metadata.

5.1 Aerospace Engineering

The first example is from engineering and is illustrated in Fig. 12. Each small multiple contains time-varying data (a movie of a scalar field) from simulations of different turbine designs (the flow moves from left to right, passing over the turbine airfoils). In exploring the results, the engineer's feature identification and knowledge capture process is enhanced by the following aspects of the visual analytics environment: **piling (grouping)**; **annotation**; **spatial metaphor**; **meta-data correlation**; **knowledge reuse**. We discuss each of these in turn.

Piling The user first interprets the visualisations, and creates a group of what they consider to be similar types of behaviour as displayed in the small multiple images. Once they are satisfied that the grouping is meaningful they can save a group by entering a name in the 'tag-name' input, and submitting the annotations.

Annotation In this example, the shape of the light blue/green coloured contours on the right-hand side of the small multiples is of interest. In some of the images, these contours are arranged in chevron pattern that repeats vertically from top to bottom; in others, the pattern repeats, but the alignment is no longer vertical. The user wishes to capture this observation in order to develop potential explanations. First, they type in 'chevron-slope' as the tag name, and then add a line to represent the chevron arrangement. The user also decides to capture the slope angle explicitly to support correlations with other metadata attributes, and enters the appropriate value into the value input form. After submitting the annotation the user creates a comment to hypothesise about the slope.

Spatial metaphor To capture an estimate of the 'chevron-slope' for other small multiples, the user rearranges them from left to right, left representing a shallow slope, and the right representing a steep slope. Using the batch form, the name 'chevron-slope' is added as the tag-name to all small multiples in view, and the x coordinates are used to quickly encode the approximate slope angle.

Meta-data correlation After the annotations are submitted, the user can see which metadata attributes are correlated with the slope values. Potential explanation ideas can be recorded using comments. To preserve the context in which the observation was gained, the user can group all relevant small multiples, and add a tag to them. As group small multiples aggregate comments of all members, the explanation of the behavior will appear in the comments of the group small multiple.

Knowledge capture and reuse A broad scope of knowledge has been captured during this process and stored on the database for reuse, by the same practitioner or others. The group of small multiples has been tagged, the approximate 'chevron-slope' encoded, and the important correlations to the metadata recorded.

5.2 Chess

The second example features chess game ensembles, and is illustrated in Fig. 13. Each small multiple contains a game of chess, and the analysis process follows the one outlined in Sect. 5.1.

From a visualisation standpoint the chess dataset example is fundamentally different to the aerospace engineering dataset. Chess game data is discrete, and does not require high resolution to interpret, therefore the zoom and pan interactions within the small multiples are not needed. However, chess data exploration requires other interactions: the user may wish to play through different variations to explore individual positions. The set of allowed moves changes with every ply (half-move), and thus the set of allowed interactions must change on-the-fly. Geometrical annotations must also be changed to allow convenient use with discrete data, both for area selection and movement indication. Chess game data may include more than one temporal property (e.g. the move sequence, and the time from start of the game itself), which must be accommodated by the playbar. Because all view interactions, including the geometry annotation tool, and the playbar, are prescribed by the view module of the small multiples, only that module needs to be changed to accommodate datasets with different demands.

Piling The user interprets individual games, and groups them with other games based on a similarity metric: the major themes of the game, pawn structure similarity, predominantly used pieces, for example. They can encode the similarity by submitting an appropriate tag annotation, which will also save and share the group with other analysts.

Annotation In this example, the user identified 'advanced pawns' (pawns far in the opponents side of the board) as the main theme of several games. To get a better understanding of the difficulties of such

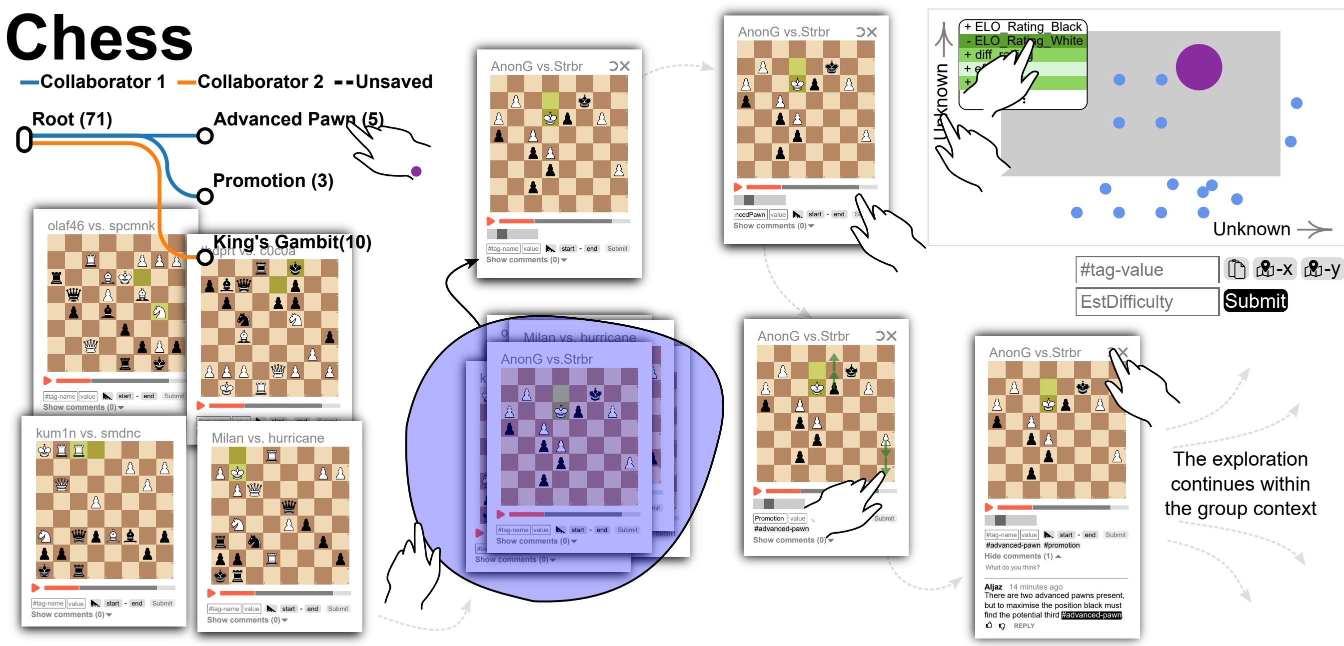

Fig. 13. Example exploration of an ensemble dataset of chess games. The user can observe themes in the chess games, and group together games that contain particular themes. To save the identified themes the user can add tags to the data. When comparing several games the user can test their hypothesis by spatially encoding relevant interpretations of the games, and then leverage the spatial encoding to find any strong correlations with the metadata. The user can capture their hypothesis, results of hypothesis testing, and their deeper understanding of the position in the comments. Other collaborators can reflect on their findings, and provide alternate views.

positions, the user decides to group the games that feature advanced pawns. To capture their observation of the advanced pawn theme, they add a tag through the group annotation form, which saves the group in the server-side database. Then the user further analyses the position to discover the threat of promotion as an important factor for both players. They use a geometrical annotation to illustrate this threat. After they are done reviewing the implications of promoting the advanced pawn they decide to capture their understanding of the position in a comment.

Spatial metaphor The user hypothesises that positions in which the player must recognise several distinct themes and connect them together in order to gain an advantage are more difficult. To test the hypothesis they arrange the positions from bottom to top, bottom representing positions with few themes, and the top representing more difficult positions. Using the batch form, the name 'EstDifficulty' is added as the tag-name to all small multiples in view, and the y coordinates are used to quickly encode the approximate number of themes observed.

Meta-data correlation. After the annotations are submitted, the user wants to see if the number of themes correlates with the players' chess expertise, as measured by their ELO rankings. They open the y axis correlation menu, and they see that although there is a strong correlation with the ELO ranking of white, there is weak correlation with the ELO ranking of black. The user may capture this contradiction in the comments, to alert other collaborators to it, and get their view on an unexpected result.

Knowledge capture and reuse. During the analysis so far the user has identified a form of similarity between a group of small multiples, and captured it for reuse. They made basic observations about a particular position in a game, and captured deeper understanding in a comment. Furthermore, they formulated a hypothesis which is captured in the comments to foster discussion with collaborators.

6 Conclusions

We provide an interactive, web-based, collaborative Visual Analytics environment for ensembles of image-based data. The twin goals are: first, to allow domain experts to rapidly explore the images and hence foster the generation of new groupings, ideas and hypotheses; and, second, to capture and visualise the knowledge created for reuse by collaborators both asynchronously and in real-time.

The proposed environment is built on a **Small Multiples Workspace** and on **Knowledge Capture and Management**:

Small Multiples Workspace

The key unit of visualisation is the small multiple. Each small multiple comprises a view, an annotation form and a comment form. The view displays the image and also facilitates interactions that are configurable to the particular data being studied. Grouping and spatial arrangement of the small multiples is supported. The latter can can be used to encode the user's knowledge prior to correlation with image metadata.

Knowledge Capture and Management

Knowledge is captured using a range of tag options (by text, value, temporal sequence or geometry) that also allow the user to create groups. Comments support real-time exchange of opinions, and can reference annotations to ease discussions of observed patterns and events.

The Knowledge Tree Visualisation supports the synthesis of different types of annotations made by several authors. It supports different points of view, either made by a single author or a team. Groups are visualised in the trees as nodes, and clicking on the node text markers allows navigation to the group. The navigation through the tree allows the user to navigate between completely separate branches.

The versatility of the proposed environment is illustrated by two contrasting examples: from aerospace engineering simulations; and from chess.

Acknowledgments

The authors wish to thank Dr Marcus Meyer for his support and advice during this work.

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/L015943/1 for the EPSRC Centre for Doctoral Training in Gas Turbine Aerodynamics.

REFERENCES

[1] E. M. Altmann, J. G. Trafton, and D. Z. Hambrick. Momentary interruptions can derail the train of thought. *Journal of Experimental Psychology: General*, 143(1):215, 2014.

[2] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec. Discovering value from community activity on focused question answering sites: a case study of stack overflow. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 850–858, 2012.

[3] L. Argote and P. Ingram. Knowledge transfer: A basis for competitive advantage in firms. *Organizational behavior and human decision processes*, 82(1):150–169, 2000.

[4] L. Argote, P. Ingram, J. M. Levine, and R. L. Moreland. Knowledge transfer in organizations: Learning from the experience of others. *Organizational behavior and human decision processes*, 82(1):1–8, 2000.

[5] S. K. Badam, S. Chandrasegaran, and N. Elmqvist. Integrating annotations into multidimensional visual dashboards. *Information Visualization*, 0(0):14738716221079591, 0. doi: 10.1177/14738716221079591

[6] J. Bernard, D. Daberkow, D. Fellner, K. Fischer, O. Koepler, J. Kohlhammer, M. Runnwerth, T. Ruppert, T. Schreck, and I. Sens. Visinfo: a digital library system for time series research data based on exploratory search—a user-centered design approach. *International Journal on Digital Libraries*, 16(1):37–59, 2015.

[7] J. Bernard, D. Sessler, T. May, T. Schlomm, D. Pehrke, and J. Kohlhammer. A visual-interactive system for prostate cancer cohort analysis. *IEEE computer graphics and applications*, 35(3):44–55, 2015.

[8] J. Bernard, M. Zeppelzauer, M. Sedlmair, and W. Aigner. Vial: a unified process for visual interactive labeling. *The Visual Computer*, 34(9):1189–1207, 2018.

[9] Y. Bian, J. Wenskovitch, and C. North. Deepva: Bridging cognition and computation through semantic interaction and deep learning. *arXiv preprint arXiv:2007.15800*, 2020.

[10] `http://www.d3js.org`. [Online; accessed 15-March-2022].

[11] M. Dahshan, N. F. Polys, R. Jayne, and R. M. Pollyea. Making sense of scientific simulation ensembles with semantic interaction. In *Computer Graphics Forum*, vol. 39, pp. 325–343. Wiley Online Library, 2020.

[12] Y. Dong, A. Fauth, M. Huang, Y. Chen, and J. Liang. Pansytree: Merging multiple hierarchies. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 131–135. IEEE, 2020.

[13] S. M. Drucker, D. Fisher, and S. Basu. Helping users sort faster with adaptive machine learning recommendations. In *IFIP Conference on Human-Computer Interaction*, pp. 187–203. Springer, 2011.

[14] A. Endert, L. Bradel, and C. North. Beyond control panels: Direct manipulation for visual analytics. *IEEE computer graphics and applications*, 33(4):6–13, 2013.

[15] A. Endert, R. Chang, C. North, and M. Zhou. Semantic interaction: Coupling cognition and computation through usable interactive analytics. *IEEE Computer Graphics and Applications*, 35(4):94–99, 2015.

[16] A. Endert, P. Fiaux, and C. North. Semantic interaction for visual text analytics. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 473–482, 2012.

[17] V. Estivill-Castro. Why so many clustering algorithms: A position paper. *SIGKDD Explor. Newsl.*, 4(1):65–75, {jun} 2002. doi: 10.1145/568574.568575

[18] C. Frappaolo. Implicit knowledge. *Knowledge Management Research & Practice*, 6(1):23–25, 2008.

[19] M. E. Gorman. Types of knowledge and their roles in technology transfer. *The Journal of Technology Transfer*, 27(3):219–231, 2002.

[20] M. Graham and J. Kennedy. A survey of multiple tree visualisation. *Information Visualization*, 9(4):235–252, 2010.

[21] D. P. Groth and K. Streefkerk. Provenance and annotation for visual exploration systems. *IEEE transactions on visualization and computer graphics*, 12(6):1500–1510, 2006.

[22] J. O. Hauglid and J. Heggland. Savanta—search, analysis, visualisation and navigation of temporal annotations. *Multimedia Tools and Applications*, 40(2):183–210, 2008.

[23] J. Heer, F. B. Viégas, and M. Wattenberg. Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 1029–1038, 2007.

[24] D. Herremans, S. Yang, C.-H. Chuan, M. Barthet, and E. Chew. Immaemo: A multimodal interface for visualising score-and audio-synchronised emotion annotations. In *Proceedings of the 12th international audio mostly conference on augmented and participatory sound and music experiences*, pp. 1–8, 2017.

[25] `http://www.jupyter.org`. [Online; accessed 15-March-2022].

[26] A. Kotnik and G. Pullan. A spatial arrangement visualisation strategy for learning from large ensembles of unsteady flow-field data. In *AIAA SCITECH 2022 Forum*, p. 0810, 2022.

[27] F. Lekschas, B. Bach, P. Kerpedjiev, N. Gehlenborg, and H. Pfister. Hipiler: Visual exploration of large genome interaction matrices with interactive small multiples. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):522–531, 2018. doi: 10.1109/TVCG.2017.2745978

[28] F. Lekschas, X. Zhou, W. Chen, N. Gehlenborg, B. Bach, and H. Pfister. A generic framework and library for exploration of small multiples through interactive piling. *IEEE Transactions on Visualization and Computer Graphics*, 10 2020. doi: 10.1109/TVCG.2020.3028948

[29] D. Leonard and S. Sensiper. The role of tacit knowledge in group innovation. *California management review*, 40(3):112–132, 1998.

[30] D. Liu, P. Xu, and L. Ren. Tpflow: Progressive partition and multidimensional pattern extraction for large-scale spatio-temporal data analysis. *IEEE transactions on visualization and computer graphics*, 25(1):1–11, 2018.

[31] N. Mahyar and M. Tory. Supporting communication and coordination in collaborative sensemaking. *IEEE transactions on visualization and computer graphics*, 20(12):1633–1642, 2014.

[32] L. M. Markus. Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success. *Journal of management information systems*, 18(1):57–93, 2001.

[33] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv*, abs/1802.03426, 2018.

[34] J. Novak and M. Wurst. Supporting knowledge creation and sharing in communities based on mapping implicit knowledge. *J. Univers. Comput. Sci.*, 10(3):235–251, 2004.

[35] `http://www.observablehq.com`. [Online; accessed 15-March-2022].

[36] `https://pixijs.com`. [Online; accessed 15-March-2022].

[37] K. M. Propp. Collective information processing in groups. *The handbook of group communication theory and research*, pp. 225–250, 1999.

[38] G. Pullan. A web-based database approach to cfd post-processing. In *55th AIAA Aerospace Sciences Meeting*, p. 0814, 2017.

[39] G. Pullan. Making use of our data. *Journal of the Global Power and Propulsion Society*, 2021(May):1–13, 2021.

[40] G. Pullan, T. Chuan, D. Wong, and F. Jasik. Enhancing web-based cfd post-processing using machine learning and augmented reality. In *AIAA Scitech 2019 Forum*, p. 2223. AIAA, 2019. doi: 10.2514/6.2019-2223

[41] `http://www.quora.com`. [Online; accessed 15-March-2022].

[42] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE transactions on visualization and computer graphics*, 20(12):1604–1613, 2014.

[43] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding, and C.-T. Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017.

[44] C. Schmidt, B. Grundel, H. Schumann, and P. Rosenthal. Annotations in different steps of visual analytics. In *VISIGRAPP (3: IVAPP)*, pp. 155–163, 2021.

[45] C. Schmidt, P. Rosenthal, and H. Schumann. Annotations as a support for knowledge generation. In *VISIGRAPP (3: IVAPP)*, pp. 264–272, 2018.

[46] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014.

[47] Y. B. Shrinivasan and J. J. Van Wijk. Supporting the analytical reasoning process in information visualization. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1237–1246, 2008.

[48] R. Simpson, K. R. Page, and D. De Roure. Zooniverse: observing the world's largest citizen science platform. In *Proceedings of the 23rd international conference on world wide web*, pp. 1049–1054, 2014.

[49] `https://stackexchange.com`. [Online; accessed 15-March-2022].

[50] H. Stitz, S. Gratzl, H. Piringer, T. Zichner, and M. Streit. Knowledgepearls: Provenance-based visualization retrieval. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):120–130, 2018.

[51] S. Van Den Elzen and J. J. Van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *2011 IEEE conference on visual analytics science and technology (VAST)*, pp. 151–160. IEEE, 2011.

[52] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of*

Machine Learning Research, 9(86), {nov} 2008.

[53] P. Vanhulst, F. Évéquoz, R. Tuor, and D. Lalanne. Designing a classification for user-authored annotations in data visualization. In *VISIGRAPP (3: IVAPP)*, pp. 85–96, 2018.

[54] B. Vasilescu, A. Serebrenik, P. Devanbu, and V. Filkov. How social q&a sites are changing knowledge sharing in open source software communities. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pp. 342–354, 2014.

[55] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.

[56] G. Wang, K. Gill, M. Mohanlal, H. Zheng, and B. Y. Zhao. Wisdom in the social crowd: an analysis of quora. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 1341–1352, 2013.

[57] J. Wenskovitch and C. North. Pollux: Interactive cluster-first projections of high-dimensional data. In *2019 IEEE Visualization in Data Science (VDS)*, pp. 38–47. IEEE, 2019.

[58] W. Willett, J. Heer, J. Hellerstein, and M. Agrawala. Commentspace: structured support for collaborative visual analysis. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 3131–3140, 2011.

[59] J. Wu, J. Wang, H. Xiao, and J. Ling. Visualization of high dimensional turbulence simulation data using t-sne. In *19th AIAA Non-Deterministic Approaches Conference*, p. 1770, 2017.

[60] D. Xu and Y. Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, 2015.

[61] J. Zahálka, M. Worring, and J. J. Van Wijk. Ii-20: Intelligent and pragmatic analytic categorization of image collections. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):422–431, 2020.

[62] J. Zhao, M. Glueck, S. Breslav, F. Chevalier, and A. Khan. Annotation graphs: A graph-based visualization for meta-analysis of data based on user-authored annotations. *IEEE transactions on visualization and computer graphics*, 23(1):261–270, 2016.

[63] J. Zhao, M. Glueck, P. Isenberg, F. Chevalier, and A. Khan. Supporting handoff in asynchronous collaborative sensemaking using knowledge-transfer graphs. *IEEE transactions on visualization and computer graphics*, 24(1):340–350, 2017.