



HAL
open science

A Survey on Progressive Visualization

Alex Ulmer, Marco Angelini, Jean-Daniel Fekete, Jörn Kohlhammer, Thorsten May

► **To cite this version:**

Alex Ulmer, Marco Angelini, Jean-Daniel Fekete, Jörn Kohlhammer, Thorsten May. A Survey on Progressive Visualization. IEEE Transactions on Visualization and Computer Graphics, inPress, 10.1109/TVCG.2023.3346641 . hal-04361344

HAL Id: hal-04361344

<https://inria.hal.science/hal-04361344>

Submitted on 22 Dec 2023

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

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



Distributed under a Creative Commons Attribution 4.0 International License

A Survey on Progressive Visualization

Alex Ulmer , Marco Angelini , Jean-Daniel Fekete , Jörn Kohlhammer , Thorsten May ,

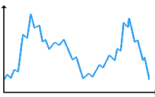




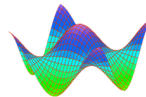
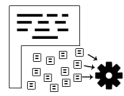


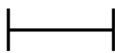

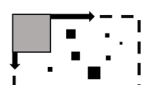
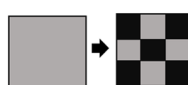
Visualization Categories	Temporal	Geo-Spatial	Hierarchical	Network	Multidimensional	Field
						
Progressive Processing Categories	Data Chunking		Process Chunking		Custom Chunking	
						
Data Domain Categories	Known End			Unknown End		
						
Visual Update Pattern	Extension			Overwrite		
						

Fig. 1: Overview of our taxonomy with its four major categories: visualization, progressive processing, data domain, and visual update pattern. The visualization categories are based on previous taxonomies [47, 65, 85]. The progressive processing categories are an extension of the progressive visual analytics characterization [5], adding a new “Custom chunking” variant. The data domain categories cover the difference in visualization design if the end of the data or process is known or unknown. The fourth category is the visual update pattern which indicates how the visualizations are updated when new partial results are produced.

Abstract—Currently, growing data sources and long-running algorithms impede user attention and interaction with visual analytics applications. Progressive visualization (PV) and visual analytics (PVA) alleviate this problem by allowing immediate feedback and interaction with large datasets and complex computations, avoiding waiting for complete results by using partial results improving with time. Yet, creating a progressive visualization requires more effort than a regular visualization but also opens up new possibilities, such as steering the computations towards more relevant parts of the data, thus saving computational resources. However, there is currently no comprehensive overview of the design space for progressive visualization systems. We surveyed the related work of PV and derived a new taxonomy for progressive visualizations by systematically categorizing all PV publications that included visualizations with progressive features. Progressive visualizations can be categorized by well-known visualization taxonomies, but we also found that progressive visualizations can be distinguished by the way they manage their data processing, data domain, and visual update. Furthermore, we identified key properties such as uncertainty, steering, visual stability, and real-time processing that are significantly different with progressive applications. We also collected evaluation methodologies reported by the publications and conclude with statistical findings, research gaps, and open challenges. A continuously updated visual browser of the survey data is available at visualsurvey.net/pva.

Index Terms—Progressive Visual Analytics, Progressive Visualization, Taxonomy, State-of-the-Art Report, Survey

1 INTRODUCTION

The research field of progressive visualization (PV) and visual analytics (PVA) have gained increasing attention in the past years to use more complex algorithms processed ever larger data sources. Although the name progressive visual analytics is relatively new, the core concept of

creating a progressively improving representation of large data or time-intensive algorithms was already introduced over 20 years ago. PVA can be split into data processing and visual-interactive parts. While data processing has been the main subject of several studies, e.g., [21, 34], visualization aspects, on the contrary, received less attention. This survey focuses on the design of progressive visualizations. Earlier research for this concept is also known as incremental visualization [23], online visualization [34], fine-grain visualization [88], progressive visualization [25], per-iteration visualization [49], optimistic visualization [64], or approximate visualization [52]. This shows that the research field is applied in many domains but is not completely defined and still has many open questions that have to be explored. One of these questions is what makes a visualization progressive compared to traditional, non-progressive (*instantaneous*, *monolithic*, or *eager* [5]) visualizations. Therefore, we decided to gather all PVA-related publications which provide a progressive visualization for data analytics and created the first survey in this research area.

As we will see in this survey, starting in the 2000s, large multi-

- Alex Ulmer was with Fraunhofer IGD and TU Darmstadt, Germany. E-mail: alex.ulmer@igd.fraunhofer.de
- Marco Angelini was with Link Campus University, Sapienza University of Rome, Italy
- Jean-Daniel Fekete was with Université Paris-Saclay, CNRS, Inria Saclay, France
- Jörn Kohlhammer was with Fraunhofer IGD and TU Darmstadt, Germany
- Thorsten May was with Fraunhofer IGD, Germany

Manuscript received 20 Mar. 2023; accepted 13 Dec. 2023. Date of Publication to appear in 2024; date of current version 21 Dec. 2023. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: [10.1109/TVCG.2023.3346641](https://doi.org/10.1109/TVCG.2023.3346641)

mensional datasets required new approaches, and the first progressive visualization systems were created. Although early solutions were limited to single visualization categories, many of the ideas in terms of data processing and progressive key properties were already sophisticated. In the 2010s, the publication rate raised significantly, and more compound PVA systems were introduced, supporting domain experts with specific data analytics tasks. A greater variety of large datasets emerged in several domains, including urban science, healthcare, IT and communication, and meteorology. The demand for tailored solutions resulted in more expert evaluations and case studies to measure the impact of contributions. This trend held on during the start of the 2020s as there are still many open research questions for PVA. However, there is currently no comprehensive summary structuring and discussing the different progressive visualization approaches.

With this survey, we intend to fill this gap. The design of progressive algorithms covers data chunking, reuse of previous partial results, and quality estimation. However, the design of progressive visualizations tackles additional complications, as for example, the visual stability between updates, the role of user interaction steering the process, and the conveyance of uncertainty. Hence, we primarily focus on those challenges that result from a user being tightly involved in the process via interactive visualizations. Therefore, the scope of this survey is the process of gaining insights from progressive visualizations. We do not cover pure analytics publications that focus on progressive algorithms i.e., [45], pipelines i.e., [17], models i.e., [97, 103], and computational steering systems i.e., [96] as their focus is primarily to improve performance.

We start by introducing the fundamentals of PVA (Section 2) and define the scope and methodology of our literature search in Section 3. The research resulted in the taxonomy illustrated in Figure 1 and explained in Section 4. In Section 5, we classify the reviewed publications, describe the specific solutions for each category. A full overview of the classified publications is shown in Table 2. In Section 6, we explain how to handle visualization with more than one type (e.g., temporal networks) and how progression can work in multiple dimensions. Section 7 covers progressive publications from other research domains that are slightly out of scope but have valuable contributions. After that, we summarize the evaluation methodologies of the surveyed publications (Section 8). We conclude by reporting statistical findings and discussing detected trends and open research gaps. We provide an accompanying website to browse all related publications and filter them according to our taxonomy at visualsurvey.net/pva.

2 FUNDAMENTALS

The fundamental idea of PVA is to increase the responsiveness of a system and improve the usability and user experience for analysis tasks. Exploration is a major task in visual analytics and requires the user to concentrate [59, 81]. Due to human cognitive constraints, the ability to focus is limited, and hampered if the system is not responsive enough. Studies have found that already at over 500 ms of *latency* (also called *response time*), significant effects on the user’s performance can be measured [67, 108]. PVA is an approach to lower and control the response times of systems by computing fast, intermediate, and approximate results, which are then improved with more time. However, in the past, the concept of PVA had different names in different communities, like online aggregation in the database community or incremental visualization in others [5, 20]. Two major contributions to the characterization and definition of PVA as well as the future challenges, were published in the past years, which we summarize briefly in the following.

Formal Definition

A traditional, non-progressive visualization (we also call it “eager”) is rendered in one pass, and each change in the data—whether due to interaction or data changes—triggers a full rendering. A progressive visualization differs in several aspects:

1. rendering is not done in one pass but in a series of passes, improving until the rendering is complete;

2. each pass limits its execution time to a requested latency and renders a *meaningful partial result*;
3. all partial results come with a meaningful error or more generally *quality measure*, as well as an estimate of the completion, e.g., by showing a progress bar;
4. a progressive visualization allows interactions at any time; it adapts as much as possible, even implementing *steering* if necessary.

Adapting definitions from [21]: Let f be a function with parameters $P = \{p_1, \dots, p_n\}$ and yielding a result value r . In mathematics, the value r calculated by f does not involve any time, and its result is perfect. Depending on the nature of the function, being perfect can mean that it has no error (e.g., when the average value of a table is computed) or it is complete (e.g., when searching for a list of relevant items):

$$f(P) \mapsto r$$

In computer science, a function takes time t to compute, and uses an amount of memory m . Also, for analytics, we distinguish the parameters P of the function from a set of *data tables* D that the function takes as input, and the ones it returns as output R . Our mathematical function f becomes a computation function F :

$$F(P, D) \xrightarrow{t, m} (R) \quad (1)$$

Note that a visualization technique is such a function. Its result R is the rendering of the visualization technique. For example, if F is a scatterplot visualization, it takes one data table d and, as parameters P , the width and height of the desired plot, the names of the data attributes used to map to the X and Y axes, and potentially the attribute used for color, texture, shape, etc. It produces as result $R = \{r\}$ either a *scene graph* of the visualization, e.g., in the SVG format, or a rendering of the visualization as an array of colored pixels. In contrast, the *progressive function* of F , denoted as F_p , is a function with three properties:

1. When called repeatedly on D_i , a set converging to D , it returns a sequence of partial results R_i , each result being computed with duration time t_i (Equation (2));
2. If q is the desired amount of time between two consecutive partial results (the *quantum* related to the maximum allowed latency), $t_i \leq q$;
3. The results R_i converges to R (see Equation (3)) or reaches a termination criterion.

$$\begin{aligned} F_p(P, D_1, q, \emptyset) &\xrightarrow{t_1, m_1} (R_1) \\ &\dots \\ F_p(P, D_z, q, R_{z-1}) &\xrightarrow{t_z, m_z} (R_z) \end{aligned} \quad (2)$$

When loading data, the growth of the data tables can be modeled with a partition $P(D)$ of D into z non-intersecting subsets, samples, or chunks: $P(D) = [d_1, \dots, d_z]$. F_p is therefore called with $D_k = \bigcup_{j=1, k} d_j$. Note that D_k does not have to grow, i.e., some d_j can be empty. For example, when computing a multidimensional projection progressively, even when all the data has been loaded, the projection algorithm keeps iterating until it reaches convergence or a termination criterion. At some point, the progressive function computing the projection is called iteratively with the whole dataset and still yields improving results. Finally, in all generality, D_k can grow and shrink during a data exploration session when the analyst performs dynamic filtering to zoom and filter. When applied to a progressive visualization technique, the quantum q should be below 10s; less when interacting with the visualization.

Convergence: If convergence can be guaranteed, the convergence criterion for the progressive computation of F_p is defined as

$$\lim_{j \rightarrow \infty} R_j = R \quad (3)$$

If a distance $\rho(R_j, R_k)$ measures the amount of changes between two results, the Cauchy convergence criterion for Equation (3) states that the partial results should verify the following property: there exists a constant N such that $\rho(R, R_{j+1}) \leq \rho(R, R_j)$ when $j > N$.

While it is desirable that the convergence happens as quickly as possible (N to be as small as possible), practically, it depends on the used function F , parameters p , data tables D , computation method, and distance function ρ .

This convergence criterion is needed to make sure that, without interaction, the results of a progressive function end up being the same as the results of the equivalent eager function. For example, a bar chart computed over a large dataset in an eager way should eventually look identical to one computed progressively. When interacting and steering a visualization, e.g., when moving a node in a node-link visualization when a force-based layout is being computed progressively, the convergence criterion still applies since the layout should eventually converge to a steady state. The type of convergence can depend on the visualization technique, but it mostly depends on the data distribution and chunking strategy that cannot always be controlled. Generally, a user may change parameters, data and results at virtually any time in the process, through interaction. Reaching convergence cannot be expected while a user keeps modifying the parameters. However, to produce a meaningful result, convergence must be guaranteed from any possible state (P, D, q, R) resulting from an interactive modification, provided the process is given enough time.

Additionally, progressive visualizations can sometimes cope with non-converging results, where the final results may be meaningless. For example, computing the *mean* of a dataset with a bimodal distribution is meaningless; still, the *mean* computation can be performed. At least, through progressive visualization, a user has a chance to notice that an algorithm is not converging, because the sequence of partial results provides information about the reliability of the result.

If convergence cannot be guaranteed, progressive visualizations can become a pivotal approach to revealing computational dynamics and opening up a black-box computation on how the path to the final result looks like. A progressive visualization can reveal unstable final results, which would be invisible if produced with an eager computation. The progression with partial results can show an alternation between two stable solution states and thus reveal that there is no convergence. This is often the case in simulations where the interest is both in the dynamics of the process and in the final result. Without convergence, a termination criterion has to be defined to prevent infinite computation, and the visualization should strive to show the alternative partial results rather than only one randomly chosen final result.

Difference between Progressive, Online, and Streaming Systems

PVA systems are sometimes confused with *online* and *streaming* systems. An online system [11] must react to input without knowing the future; for data analysis, it should process its input piece-by-piece in a serial fashion and adapt to a change in the data. When data is changed, an eager visualization system needs to recompute and redraw everything from the beginning, whereas an online visualization will do its best to take into account the changes and save time and resources to recompute and partially redraw the new result. In general, an online system is faster to update than an eager system to recompute its result from the beginning. However, it is slower to perform a whole computation. Contrary to progressive systems, online systems have no guarantee that they will return a result under a given latency, and most online algorithms are meant to accommodate new data but not changing data. Yet, progressive systems and visualizations sometimes rely on online algorithms when they know how to control their latency.

A streaming system [66] manages data produced by a *data stream*, i.e., a source of data generated over time, sometimes infinite such as the temperature measured by a sensor every second. A first issue with streaming systems is managing this data with limited resources (memory, power, latency). Additionally, time is central to streaming algorithms and systems; data are time-stamped, need to be managed in real-time with specified latency bounds, and their analysis is often performed over a time window (e.g., in the last hour). To meet these

goals, streaming systems often trade quality for resources. In contrast, with progressive systems, the processing time is an artifact of the computation and has no meaning, and the computations are not done over a time window but over a whole finite dataset. Progressive visualization systems can also use streaming algorithms when their result quality is good enough for visualization.

Characterization of Progressive Visual Analytics

Angelini et al. performed a review of PVA literature in 2018 and contributed a characterization of PVA [5]. In this article PVA is compared to Instantaneous Visual Analytics (IVA) and Monolithic Visual Analytics (MVA). PVA can bridge the time until the visualization is final at t_{complete} for MVA by producing early partial results until t_{reliable} , progressing to a stable state at t_{stable} until t_{complete} is reached. There are valid use cases when the computation is slow or indefinite by design, for example, in data streams or optimization processes. The authors defined two ways in which PVA can be used to deliver partial results: process chunking and data chunking. Finally, Angelini et al. formulated nine recommendations based on their literature review for creating PVA approaches. This characterization is an important contribution to PVA and is a part of our taxonomy.

Process Chunking

Process chunking loads the full data and then performs a complex and slow iterative algorithm on the data. In this case, the iterations are the intermediate results converging to the final result. Process chunking is usually used when the processing of data points takes too long. Even if the data is not very large, the complexity of the computations is too high to be interactive. Therefore, intermediate and uncertain results of the whole dataset are shown at first. With every iteration of the algorithm, the quality of the results is improved until the final accurate view is reached. There are large research fields where this is applied in optimization algorithms, i.e., stochastic gradient descent in machine learning [12]. While the data is fully loaded at the beginning, the algorithm takes many iterations to reach the final result. The results can be deterministic or not, resulting in one fixed solution or multiple appropriate solutions according to the optimization criteria. Progressive visualizations can show the development of the optimization and thus provide more insights into the underlying system [40]. Besides supporting progressive data analysis, this can also be used to teach students how algorithms work (e.g., [Tensorflow Playground](#)) or for developers to find bugs in optimization algorithms.

Data Chunking

In data chunking, the data is divided into partitions (chunks), and intermediate results are shown progressively after each chunk is added. In this case, the data is usually too large to be loaded at once, but the processing algorithm is fast enough to produce results. Partial results may already be accurate depending on the sampling strategy and operation performed. For example, chronological data chunking for time-oriented data already shows accurate results for early times. If the data is sampled over the whole time range, early results are uncertain, but the user can get indications of the time frames that may be of interest. Overall, data chunking allows the user to view and interact with definite partial results immediately and with no or little initial loading and processing time.

Visual Stability vs. Result Quality

Exploiting progressive visualization comes with some drawbacks: one is the trade-off between the visual consistency (also result quality) of the partially updated results—which must be truthful to what the progressive computation produces—and the overall visual stability of the incrementally rendered visualization [4]. When the first is not met, the user could be induced into an erroneous evaluation of the results (e.g., looking at some bars in a bar chart that does not present the correct height due to some smoothing during the progression). When the second is not met, the user could be distracted by continuous changes in the visualization, eventually producing more confusion than help. A final consideration of the interplay of the two is that what happens

during the progression could be considered, if strongly different from both what precedes it and what follows it, as a spike or anomaly of the progression itself. This can be caused by a not-perfectly convergent progressive algorithm or by a not-perfect adaptability of the chosen visualization technique (e.g., a progressive treemap that, even for small changes in the data, produces big changes in the treemap layout). While understanding whether both properties can be fully met is an open research challenge, the correct management of their trade-off is at the basis of effective progressive visualization.

On this topic, the frequency of updates of a progressive visualization also plays a role. On the one hand, the user should not be overwhelmed with frequent updates like bouncy bars in a bar chart. On the other hand, less frequent updates aggregate too many changes displayed at once and could destroy the user's mental map. The adaptability of visualization techniques to progressive visualization and the inclusion of analytics that, based on computed visual metrics [2], automatically or semi-automatically (where the user can force the visualization update on demand) govern the need for an update (see [3] for a refresh technique based on the estimation of the visual changes) can be potential solutions to explore.

The Human User in Progressive Visualization

Human perception and decision-making raise important issues in PVA. Micallef et al. reported on users' roles, tasks, foci, and biases in progressive analysis scenarios [63]. They explained the most relevant workflows of observers, searchers, and explorers and discussed the cognitive tasks they have to cope with. An observer for example makes less use of steering functionalities which means that the progression itself has to converge to a final accurate result in a smooth and interpretable way. Searchers and explorers, however tend to steer the progression to specific parts of the data. Searchers are looking for an answer to their question and when the progressive system delivers this answer with a high confidence the computation can be stopped. Finally, explorers are interested in the full progressive process. They are interested in the comprehensive understanding of the data and process, with all the capabilities of steering and interpreting intermediate results.

Even further in this direction is the work by Procopio et al. [75]. They performed a very extensive evaluation with 26 participants to analyze the impact of cognitive biases. The result was that four types of biases occur: uncertainty, illusion, control, and anchoring bias. A study by Zraggen et al. measured the insight discovery rate of progressive visualization compared to blocking and instantaneous visualizations [108]. While blocking visualizations were detrimental, progressive and instantaneous performed equally well. This shows that, in an exploratory setting, the cognitive load for a human is similar when all data is shown at the start compared to a progressive buildup. Another way PVA systems can support humans is by providing steering functionalities [5]. The user can change parameters, direct the sampling strategy, or queue computations while the progressive system is running. At the minimum, the functions are pausing, changing update intervals, or stopping the computation. This gives the user the possibility to handle the cognitive load.

Uncertainty

The ability to effectively communicate uncertainty is an essential component of PVA. By definition, the intermediate results that a PVA system displays during a computation are not yet complete and are an estimate of the final result [2]. Users must be able to evaluate the accuracy of intermediate results to make well-informed decisions. While visualization has long grappled with the implications of uncertainty [30,42], uncertainty is even more fundamental to PVA. One of the major dangers of ineffective communication of the PVA uncertainty is that the user makes unfounded decisions by not waiting enough. For example, the user might be tempted to stop the progression at a point when the results confirm their own beliefs but will not necessarily be supported by the data eventually. Effective visualization of the PVA uncertainty should aim at reducing or avoiding potential errors [75]. The choice to use PVA can be seen as a trade-off between time and accuracy. In an effective PVA strategy, the PVA uncertainty should continuously decrease,

approaching the same final results that would be obtained where no progressive approximation is applied. From the visualization perspective, classic visual variables like position [7], color saturation [89], or explicit means like error bars for bar charts [23,71,108] are common in many progressive systems to convey uncertainty. It remains a research challenge how to communicate its variation over time, how to make it coexists with uncertainty intrinsic to the problem (data) at hand, and how to represent both global and local uncertainty.

Challenges for Progressive Visual Analytics

A 2019 Dagstuhl report by Fekete et al. provides a definition and several future challenges for PVA [20]. With the formal definition of progressiveness, the question of good quality and progress metrics arose. Quality metrics would have to cover the quality of intermediate results based on accuracy, uncertainty, and stability. There are many use cases where PVA can deliver better results due to its continuously updating results. The authors state that there are still no practical and usable progressive systems today and that a real benefit compared to already existing solutions need more validations. By providing a comprehensive overview of the latest advancements in progressive visualization, our survey responds to some of these findings.

3 SCOPE AND METHODOLOGY

The basis of our survey are structured references that are retrieved systematically based on a precise scope. We searched in proceedings of large conferences and journals while also propagating references forward and backward. We report search terms used via Google Scholar and explain how we tagged all publications to gather our literature dataset.

3.1 Scope

The research area of this survey is PVA, as defined in Section 2. Since our focus is on the visualization part of the PVA approaches, we only included publications that present at least one progressive visualization. By definition, progressive approaches need to have a final result that the partial results converge to. Following that, streaming solutions are out of the scope of our survey as they are sometimes infinite and do not always converge. Finally, the progressive improvement should not be recomputed to give the user full freedom of interaction and parametrization.

3.2 Data Collection

We started the selection of articles by looking at the references of the PVA theory publications, which define and describe the research area. Following these references, we identified major journals and conferences where many progressive approaches are published.

- Journals:
 - *IEEE Trans. on Vis. and Computer Graphics (TVCG)*
 - *Computer Graphics Forum*
 - *IEEE Computer Graphics and Applications*
 - *Journal of Visualization*
- Conferences:
 - *IEEE Visualization Conference (VIS)*
 - *Eurographics Conference on Visualization (EuroVis)*
 - *IEEE Pacific Visualization Symposium (PacificVis)*

However, due to the fact that progressive approaches were used in many other domains and under other names in the past, we also made considerable use of *Google Scholar*. We used the following search terms combined with visualization or visual analytics to find additional articles:

Progressive, optimistic, iterative, incremental, per-iteration, fine-grain, approximate, online, and real-time.

Finally, we followed citations to find more related sources. Only full papers which were peer-reviewed were added to the survey. Exceptions

were made if the contribution had a high impact or specific quality related to progressiveness.

3.3 Data Analysis

For a systematic and structured literature list, we performed five steps of tagging. At the end of each tagging step, we unified similar tags:

1. **Visualization:** We started by tagging the literature according to the visualization categories adapted from other taxonomies.
2. **Data processing:** After that, we analyzed the data processing style of the articles and tagged them accordingly. During this, we noticed that some approaches were more than data or process chunking and extended the processing styles with custom chunking.
3. **Time and Space:** Multiple tags for the visualization style were added to differentiate between changes over time and space. These tags were unified and resulted in the categories: *visual update pattern* and *data domain*.
4. **Progressive Features:** We tagged publications that wrote about particular challenges specific to progressiveness. These tags also matched important progressive properties mentioned in the PVA theory literature.
5. **Application and Evaluation:** Finally, we added tags to provide additional information about the topics of the publications and their evaluation methodologies.

3.4 Literature Dataset

The literature dataset we assembled after the search phase with all publications related to PVA contains 96 publications from the years 1991 to 2023. We filtered out articles that did not fit our scope, which mostly were theory or algorithm articles without visualizations. Also, publications that used precomputed data structures to deliver progressive access on demand were excluded, but we mention these in [Section 7](#) as they provide related ideas. Finally, we performed the tagging on a set of 48 publications from 1999 to 2023. After the unification of similar tags, we derived the taxonomy categories that are explained in the next section. The visualization type is the main category to discriminate the publications. After the explanation of the taxonomy, all publications are classified and described individually.

4 TAXONOMY

Many different progressive visualization approaches were proposed, which we categorize according to this taxonomy to have a systematic overview of all techniques. An illustration of the taxonomy is presented in [Figure 1](#). Our taxonomy is structured based on existing techniques and does not explore all possible concepts. However, there are some challenging combinations and research gaps, which we will address in [Section 5](#) and [Section 6](#). The four categories of the taxonomy are

- **Visualization:** Temporal, Geo-spatial, Hierarchical, Network, Multidimensional, and Field
- **Progressive Processing:** Data Chunking, Process Chunking, and Custom Chunking
- **Data Domain:** Known End and Unknown End
- **Visual Update Pattern:** Extension and Overwrite

Visualization Categories

The main category of the taxonomy is the visualization type. We combined the taxonomies from Shneiderman [85], Keim [47], and Munzner [65]. Shneiderman and Keim have a similar structure but also include text/web and 3D. Munzner introduces continuous fields as a type that we adopted, as it is a good way to include the many contributions from the Scientific Visualization community to progressive visualizations. Similar to Munzner, we also found that the temporal dimension is not only limited to classical charts with a timeline but can be applied to each visualization type, e.g., large dynamic graphs that can not be visualized instantly and change over time. Additionally, by

definition, for PVA, time represents the improvement of results. We address these combined and special PVA cases in [Section 6](#). Finally, we found progressive visualizations in other research domains, namely, rendering for augmented reality, virtual reality, and video games, as well as progressive text, image, and video. Despite the fact that they did not fit in our scope, they provide related ideas; we decided to give a brief overview of these approaches in [Section 7](#).

Progressive Processing Categories

The data processing categories are based on the PVA characterizations by Angelini et al. [5] and Schultz et al. [82] and are described in [Section 2](#). The processing strategy is usually chosen according to the bottleneck to be mitigated: Data chunking for large data sources and process chunking for slow algorithms. Some publications use data and process chunking in different parts of their processing pipeline, but not in an interwoven style. A few, however, have proposed more sophisticated combinations or special forms of chunking, which we categorized under the *Custom Chunking* type. Although, there are only a few publications that fit in this category we decided to include it in our taxonomy. The first reason is that progressive visualization is still a new research field and we want to prevent excluding promising research directions. Second, there are many other ways to design the progressive pipeline than just data and process chunking, but the research is still very underdeveloped in this area. Recent works by Hogräfer et al. about tailored sampling methods for PVA [38, 39] show just the tip of the iceberg of what is still to be discovered in progressive data processing.

Data Domain Categories

The data domain describes the length of the progressive results that the visualization has to deal with. This category resulted from our third tagging stage, in which we identified preconditions for visualization design based on data. We call these types “*known end*” and “*unknown end*”.

Known End

In this type, we have a completely known dataset, which allows the up-front definition of an absolute value range, aggregation levels, and layout for the visualization. This also implies that the last computation chunks are known, and absolute progress can be shown. Depending on the visualization type, different sampling strategies can be used to progress automatically or steered by a user. The uncertainty in the visualization is progressively reduced until the final result is displayed. Another feature of the *known end* type is that the final result can be deterministic. A known end also has benefits for the design of visualizations, as visual attributes and layout can be defined at the beginning. This is in contrast to the next data domain type.

Unknown End

Since the definition of PVA demands a final result reached by convergence or a termination criterion, this category has to be described more precisely. In PVA, an *unknown end* in the data domain has two meanings. Either, the time until convergence or termination is unknown, and thus the end can not be determined until all data are transmitted, or all the processes are run. Or, the result of the progression is not deterministic. Non-deterministic results often happen in optimization-based algorithms with variable parameters and termination criteria. An unknown end in the data domain has a decisive effect on the visualization design since the visual design and layout have to be dynamic and adaptive to deal with new values. Especially, visual mappings have to be chosen carefully, because some of them e.g., *color*, are not distinctive enough when many categories are streaming in.

Visual Update Pattern

This category also resulted from the third tagging stage, where we identified differences in visualization design based on changes over time and in space. Visual update pattern types directly describe how visualizations are updated when new progressive chunks are added.

This category is more than just rendering elements. It includes the visualization design with factors like visual attribute mapping, animation design, and a controlled update frequency to prevent confusing the user. The two types of visual update pattern are *extension* and *overwrite*.

Extension

Extension denotes the process when new elements are added to the visualization or the visual domain is extended to display new data. In other words, this type uses more pixels to display more data. An advantage is that old data is not lost and can give context for new data points.

Overwrite

Overwrite can be separated into two sub-types: *partial overwrite* and *full overwrite*. In a *partial overwrite*, only the elements that were changed with new partial results are modified. In a *full overwrite*, the previous results are removed completely, and new results are displayed. There are different advantages and disadvantages for each type, dependent on the application.

Key Properties for Progressive Visualizations

The key properties partly resulted from the fourth tagging stage but are also influenced by multiple theory articles that have explained the important role of these properties for PVA. This category is different because we noticed that they are not mandatory, as many publications show impactful contributions without using certain properties. Also, other factors can prevent a meaningful integration of a certain property, which we will talk about next.

Uncertainty

Uncertainty is an important property to communicate to the user along with the intermediate results shown by the visualization. When absent or badly designed, it may lead to incorrect conclusions. However, if we take a simple temporal visualization with data chunking and chronological sampling, uncertainty does not need to be visualized because the system returns exact data chunks and the progress is directly visible as time progresses. Nevertheless, for most other cases, uncertainty visualization is needed to guide early reasoning by the user.

Interaction & Steering

Interaction and steering are core properties of PVA systems, as the idea of PV is to allow interactivity and fast response times by providing intermediate results. We tagged publications that provide interactive visualizations during the progression or allow users to steer the progression by picking more relevant parts of the data first. However, steering and interactions can be very different in their extent. The possibility of stopping the process and restarting with new parameters is the simplest form of steering. In some cases, steering is not applicable because an algorithm is using an optimal path for the progression. Hence, steering and interactivity are valuable, but not always mandatory, properties of a progressive system.

Visual Stability

Visual stability ensures that updates in progressive visualizations do not interfere with the interactive analysis by a user. If partial results are produced with a high frequency and visualization updates happen very fast, the user will be confused. We tagged publications that mentioned the topics: preserving the mental map, visual stability, and visual consistency. Again, this property does not pose a problem for some progressive systems where the update frequency is low by default, or the visualization can update new partial results with small visual changes.

Real-time Processing

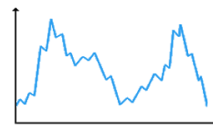
Real-time processing is necessary if the speed of computation is most important. Progressive approaches can be used as functions in larger systems that demand fast results, but also need accurate results in a reasonable time frame. We tagged publications that mentioned processing time as a major factor when designing the approach. Most of these

publications performed a technical evaluation and measured time and computational cost.

5 CLASSIFICATION OF PROGRESSIVE VISUALIZATION TECHNIQUES

In this section, we describe how approaches handled specific progressive challenges for visualization. We identify open research gaps and give insights and hypotheses on why these are demanding challenges. We refer the reader to our visual survey browser, where all solutions are shown and linked to the publications (visalsurvey.net/pva).

A TEMPORAL VISUALIZATION



Time-oriented data is used in a wide range of applications with a long research history [1]. In this category, we focus on progressive visualizations that represent time in one dimension. However, time is often one among multiple dimensions, for example, in dynamic graphs. These multi-category combinations will be discussed in Section 6. Overall, 14 out of 47 articles include a temporal visualization. We found that the temporal dimension is visualized in two general ways: continuous or aggregated. Continuous representations are shown with line charts and show a data point for each time step [41, 48, 55, 73]. If the data is too large, a sampling method is applied [29] (see Section A.1 for more on sampling strategies). The aggregated presentation with bar charts is also commonly used [78, 93, 94]. In this case, the temporal data is aggregated in buckets, and the bars show, for instance, the average or sum of the data for a time frame. Finally, we found one approach that aggregates multidimensional data for specific dates and visualizes the changes with a mixture of a Sankey diagram and a streamgraph [57].

A.1 Progressive Processing

Here, we focus on describing the sampling strategies used, which have a direct influence on the visual update pattern.

Constraints to Data Chunking

Temporal data has a natural order which made most of the approaches with data chunking use chronological sampling [41, 48, 84, 93]. Additionally, if the data is generated on the fly, for example, in progressive optimization approaches, chronological order is always used [73, 109]. Some approaches did not sample chronologically because of other constraints. For example, the order of the data chunks could not be controlled as the data was received from different data services [57, 94], or multidimensional data were sorted according to other dimensions [55, 78, 90]. The majority of temporal visualizations had a data chunking approach (13 out of 14).

Constraints to Process Chunking

Process chunking for temporal visualizations was only used in two approaches. One approach provided a coarse representation over the whole time frame by sampling evenly until the final result was computed [29]. The other approach uses adaptive sampling by choosing lesser sampled areas first [77]. More sampling strategies can be applied here that were used for other data types, and more research is necessary to evaluate how beneficial they are for temporal visualizations. Also, we think there is more space for future research as optimization scenarios, like curve fitting and trend computation, are widely used analytics features.

Custom Chunking

For temporal visualization, we have found four publications that use data and process chunking [29, 73, 77, 109]. However, the chunking methods were used in different parts of the systems. There were no interwoven forms or special chunking conditions comparable to the other custom chunking solutions we have found.

A.2 Data Domain

Here, we focus on how temporal visualizations differ based on a known or unknown data domain. Temporal visualizations are a special case in this category. Based on the aggregation method or query parameters, the approach can have an unknown end, but the time frame is already defined, making it a known end for the temporal dimension. This is why some of the publications with temporal visualization are categorized as both known or unknown ends [90, 94, 109].

Known End

The most common fact for a known end is that the time axis in the visualization can already be defined. It shows the user the full domain in which the chunks will appear during the progression [77, 78, 94]. For example, which time of day showed the most flight delays [41] or when an iterative optimization will terminate [73]. An additional benefit of temporal visualizations is that if the chunk sampling is performed chronologically, the visualization also represents a progress bar [73, 109].

Unknown End

If the end is unknown, the visualization has to be more adaptive. The temporal axis has to be extended until the maximal screen space is reached [57]. Then aggregations have to be performed to fit the data into the available space [93], or older data has to drop out [48].

A.3 Visual Update Pattern

The visual update pattern for temporal visualizations is highly dependent on the processing type.

Extension

Approaches that use data chunking with chronological sampling strategy have a defined temporal dimension and expand the visualization along this axis [55, 57, 73, 109]. However, as mentioned above, if the end is unknown, the display space might run out, and aggregations are required, which results in an overwriting update [93].

Overwrite

Overwriting updates are common with process chunking approaches. When temporal data is sampled equally, it spreads out to show a rough estimate of the data at first and is constantly overwritten and enhanced until the progression finalizes [29, 77].

A.4 Key Properties

Uncertainty Visualization

We have observed that uncertainty was rarely visualized in progressive temporal visualizations [41]. If data chunks are chronologically sampled, the data is always accurate, and no uncertainty can be displayed [73, 84, 93]. If the sampling is not chronological, there is no direct uncertainty indicator besides the progressive process itself [57, 94]. For process chunking, uncertainty visualization is very important because the user needs to know how certain the current result is before an update overwrites the current state [29].

Interaction & Steering

In most of the surveyed publications, the temporal visualization was a supporting component without interactive features. Approaches with interactivity usually provide a mechanism to focus on relevant time frames and thus steer the system. This is performed via panning [29], brushing [78, 93, 94], and selection [55].

Visual Stability

Visual stability was not covered in most of the approaches because, as mentioned before, most approaches use data chunking and update the visualization by extension. This preserves the mental map of the user while also giving a progress indicator. The cases where visual stability is important are when overwriting update patterns are used. For example, when coarse data is replaced with more accurate data in process chunking approaches [29] or when the aggregation level has to be changed [93].

Real-time Processing

Real-time processing was only necessary for four out of the 14 temporal progressive visualizations. The requirement was highly dependent on the use cases in which fast decisions had to be made in performance monitoring [48, 84], or analysis of current discussion topics in news or social media [57].

A.5 Challenges

Temporal progressive visualizations are mainly used with data chunking because the temporal dimension has a unique chronological order which is easy to understand for users. Most of the surveyed approaches used this sampling order as it preserves the user's mental map. Only a few articles [29, 77] used another sampling strategy with a focus on perceptual importance. Progressive temporal visualizations with perceptually important points (PIPs) is a challenging research direction that may possibly lead to earlier reliable results. Chronological sampling often helps with multidimensional data where the temporal dimension is used as the progression direction. However, this is not the case if data needs to be aggregated over time intervals. In an unknown end scenario the display space may run out and the aggregation has to be switched which produces a strong visual instability. This challenge is connected to the overwrite update pattern in general, which can also be found in process chunking approaches. A sophisticated transition is necessary to prevent confusing the user when updating the intermediate results. Process chunking approaches cannot exploit the chronological order for conveying progress and must rely on other means to display the uncertainty in a comprehensible way. Finally, there is no research on progressive temporal visualizations with radial axis, which may be due to niche application for the analysis of periodic features.

B GEO-SPATIAL VISUALIZATION



Geo-spatial visualizations are mainly used for cartography but also in three-dimensional scenarios [51]. Certain data dimensions are implicitly geo-related. This includes coordinates like latitude and longitude to map the data on the world map or an id referring to a predefined region.

The coordinates can also be three-dimensional to describe the height above sea level or even define a position in a local coordinate system for augmented or virtual reality. In progressive visualization systems, this category benefits from its independent location data, making it possible to deliver early accurate results for specific regions. In our survey, 11 out of 47 publications include geo-spatial visualizations, which cover point maps [14, 52, 82, 92, 102], symbol maps [56], choropleths [3], heatmaps [34, 109] and flow maps [90, 94]. All of them use geo-spatial visualization as a supporting component without a direct influence on the progression or sampling strategy.

B.1 Progressive Processing

Here, we focus on describing the sampling strategies used, which have a direct influence on the visual update pattern.

Constraints to Data Chunking

Most of the publications use data chunking in conjunction with geo-spatial visualizations (8 out of 10). But the sampling of the data chunks is almost always not affected by the geo-spatial dimension. Only two articles use advanced sampling approaches. Chen et al. [14] pre-compute a pyramid-based structure to provide a more density-oriented sampling strategy, while Kwon et al. [52] explore how the user could interactively steer the sampling. A progressive sampling approach directly dependent on the geo-spatial dimension remains to be investigated.

Constraints to Process Chunking

Four publications use process chunking with geo-spatial visualizations. As mentioned above, the sampling strategy was not based on the geo-spatial dimension. However, these approaches use the results from early partial results to influence the next chunks [34, 56, 82]. Another example of this is PEviz [109], where each process iteration shows the development of ocean water flows.

Custom Chunking

For geo-spatial visualization, we have found two publications that use both data and process chunking [34, 109]. However, they do not qualify for a custom chunking approach because the chunking methods were used in different parts of the system. There are no interwoven forms or special chunking conditions comparable to the other custom chunking solutions we have found. A possible research direction would be to explore geographical quality metrics combined with regional sampling.

B.2 Data Domain

In this section, we focus on how geo-spatial visualizations differ based on a known or unknown data domain. In both cases, we have found that a boundary for the geo-spatial dimension can be defined. For example, there may be indefinite points (unknown end) or a set of regions (known end), but all of them are geographically in one country. This means the visualization can already be adapted to the preset boundary and only show the country. If this boundary can not be defined, the visualization has to show the whole space (e.g., the world map) and be interactive or provide automatic focus features.

Known End

Approaches with a known geo-spatial data domain used a wider variety of visualization styles. In two publications, the authors combined choropleth with symbol maps because the number of regions was known, and the coloring and symbols could be adjusted to this [3, 56]. In another article, the authors overlaid a grid heatmap over a point map to show uncertainty, which is only possible if the end of the data domain is known [35]. Three approaches that use data with a known end use no aggregations and visualize the data with a point map [82, 92, 102].

Unknown End

All publications in this category use either point or flow maps, as no aggregations can be made initially. Two articles show the whole world map with interactive focus features, while new chunks enter the visualization [94, 109]. Another two publications pre-define boundary regions, and the number of points or flows that will be displayed at the end are unknown [14, 90].

B.3 Visual Update Pattern

The visual update pattern is highly dependent on the data domain and, thus, of the geo-spatial visualization type.

Extension

All four articles that have an unknown end in the data domain use point [14, 109] or flow maps [90, 94] as extensions for visual updates to represent data points, as these can be added independently to the visualization for each chunk. The challenge for geo-spatial visualizations is that the geographic space is limited, and if many data points share a location, visual clutter becomes a problem [94].

Overwrite

The main geo-spatial visualization types that use overwrites as update pattern are aggregation visualizations like heatmaps and choropleth maps. All five publications with a known end for the data domain make use of this and provide early aggregation visualizations that are overwritten when new chunks arrive.

B.4 Key Properties

Uncertainty Visualization

Uncertainty for geo-spatial visualizations is also dependent on the data domain. As mentioned before, if the end is unknown, all approaches use point and flow maps but without a representation of uncertainty. This may be due to unavailable information for geographic accuracy or because adding uncertainty to a location or path can make the visualization more difficult to understand. Future research is necessary to elaborate on uncertainty in progressive geo-spatial visualizations with an unknown end. When the data domain end is known, however, uncertainty indicators are used via heatmaps [34] or choropleth maps with symbols [3].

Interaction & Steering

In general, approaches without a defined boundary region provide free navigation in the geographic space, while those with a defined boundary are mostly static. In some articles, it is possible to interactively filter data in the geo-spatial visualization [56, 90, 92], but none of the approaches provide a way to steer the progression via the geo-spatial visualization. Although this idea is already used in multidimensional approaches, direct interactive steering of the progression by selecting regions or points of interest still remains an unexplored area of PV research.

Visual Stability

For geo-spatial visualization, visual stability is not only a problem of update frequency but also covers visual clutter and overplotting. Since the geographical space is limited, new results can make the visualization more difficult to read. None of the publications use approaches like dynamic aggregation techniques, though one article makes use of side-by-side snapshots of point maps to show the progress instead of adding all data points to one map [109].

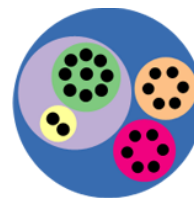
Real-time Processing

Real-time processing is less prominent in progressive geo-spatial visualizations, as only two articles address this topic. One focuses on enhancing the sampling strategy of geo-spatial data [14], and since this is an essential pre-computation step for other approaches; it has to be optimized at run-time. The other article proposes a fast visualization of field data from hurricanes, which is displayed on a world map [102].

B.5 Challenges

A very important finding is that all publications used the geo-spatial visualizations as a supporting visualization and not as the main interactive structure to steer the algorithm or apply a regional sampling strategy based on user input. Regional resampling would require special pre-processing of the data to allow sampling based on the geo-spatial dimensions. This is similar to hierarchical visualizations, which we cover in the next section, because geographical regions can be clustered in multiple zoom levels. This gives the user better steering options, which needs to be evaluated to show the benefit over non-progressive visualization designs. Another difficulty is to keep the users mental map intact when using overwriting visual update pattern, especially when uncertainty is visualized. Uncertainty can be misunderstood easily in geographical visualizations and progressive updates may result in frequently changing and temporary high uncertainty values which may disturb the user. Overall, the challenges for geographical visualizations will only be made clear when approaches start to use the geo-spatial dimension as the primary progression direction. A first step in this direction has been done by Hogräfer et al. [38].

C HIERARCHICAL VISUALIZATION



Hierarchical visualizations represent an ordered structure of relations or data that is aggregated on multiple levels to improve scalability and readability [19]. In this visualization category, multiple visualizations of the same data can be shown on different aggregation levels. These levels create a hierarchy from the highest level of detail, the raw data points, to the lower levels of detail, where data points are summarized by an aggregation function. Such functions range from simple arithmetic operations to complex similarity clusterings. The hierarchy levels can be visualized together or separately.

Such hierarchies can be an advantage for progressive approaches because the different levels can be progressively computed and displayed in a bottom-up or top-down style. Nonetheless, there are only five publications that use progressive hierarchical visualizations. The types of visualizations used in those approaches are treemaps [79], parallel coordinates [80], Sankey [56, 57], and sunburst diagrams [90]. The three publications on parallel coordinates and Sankey diagrams are listed here because they use a hierarchical progressive approach.

C.1 Progressive Processing

Here, we focus on describing the sampling strategies used, which have a direct influence on the visual update pattern. For hierarchical visualizations, we found that all approaches create the hierarchy in a pre-processing step with process chunking and later update the visualization by using data chunking. In general, sampling for hierarchical data follows four strategies: bottom-up or top-down with a depth-first or breadth-first selection. The possibility to let users steer the sampling is specific to the progressive implementation and introduces new concepts.

Constraints to Data Chunking

Three of the five publications use data chunking to update the visualization progressively. Rosenbaum et al. use a top-down approach with a mixture of depth- and breadth-first [79], depending on the number of subtree levels. In another work, they use parallel coordinates where a similar top-down approach is called the cluster-of-interest concept [80]. But they also provide sampling based on the dimensions of the parallel coordinates (dimension-of-interest). Both concepts are driven by user interaction, which shows how progressive processing creates new opportunities. The third approach by Liu et al. uses bottom-up sampling [57] to create a dynamic Sankey diagram. As new data chunks arrive, they are categorized into existing or new hierarchy levels based on the similarity distance to the previous data.

Constraints to Process Chunking

For the two publications mentioned above, the creation of the hierarchies is done in a process chunking approach. The treemap is constructed with a top-down sampling approach while (depending on the available computation resources) nodes are ordered in depth-first or breadth-first style [79]. The parallel coordinates hierarchy is constructed with a top-down sampling approach with a recursive interval subdivision (RISD) [80]. The other two approaches with process chunking use hierarchical visualization to support progressiveness by displaying the interaction path of the user [90] or a filtered subset of the data [56].

Custom Chunking

For hierarchical visualization, we have found two publications that use both data and process chunking [79, 80]. However, they do not qualify for a custom chunking approach because the chunking methods were used in different parts of the system. There are no interwoven forms or special chunking conditions comparable to the other custom chunking solutions we have found. A possible research direction would be to explore custom chunking methods based on the nodes' hierarchy level and the number of children.

C.2 Data Domain

Known End

For progressive hierarchical visualizations, there is a strong bias towards a known data domain because then it is possible to pre-compute the hierarchy. Four out of five publications in this category have a known end in the data domain. If the end of the data is known, a top-down approach can be used, making it easier to handle new data points from progressive results.

Unknown End

One publication related to progressive hierarchical visualization used data with an unknown end [57]. They receive a stream of data chunks and construct a dynamic hierarchical structure over time. However, this is the only approach in this research direction, and in our opinion, it remains a challenge to adapt this idea to other hierarchical visualizations. Dynamic data requires resource-heavy re-computations of hierarchies which demand more sophisticated progressive approaches.

C.3 Visual Update Pattern

A characteristic of hierarchical visualizations is that the top hierarchy level represents the highest data aggregation.

Overwrite

Because of that, the two progressive hierarchy approaches in our survey use the overwrite pattern to update their visualization [79, 80]. This is an intuitive way to update the visualization because it maintains the mental map of the user and shows the completion status indirectly.

Extension

There are two other publications where the hierarchy visualization has a supporting role, both with extension as the primary update pattern. The Sankey diagrams are extended based on the user interaction history or only show filtered data [56, 90]. These approaches only use an overwriting update when the analysis is restarted to reset to the initial state. The approach by Liu et al. deals with the unknown data domain by visualizing new data as points and progressively packing them to already existing hierarchy levels or constructing new ones if necessary [57]. Overall, the Sankey diagram is extended along a time axis and shows the evolution of the hierarchical topic structure.

C.4 Key Properties

Uncertainty Visualization

The representation of uncertainty for hierarchical visualization is, in general, difficult and a topic of ongoing research [87]. All five articles in our survey do not use uncertainty indicators because the data is loaded in chunks that are precise for the current level of aggregation. It would be interesting to investigate uncertainty visualization for future approaches with dynamic hierarchies.

Interaction & Steering

The approach with progressive treemaps allows direct steering of the progression by selecting rectangles in the treemap, which forces the algorithm to make a depth-first sampling from this node onwards [79]. The progressive parallel coordinates approach allows the selection of clusters and the selection of dimensions to steer the algorithm. The approach with the Sankey diagram [57] does not provide steering but allows the user to compare new data points interactively. The other two publications use hierarchical diagrams to support the workflow, allowing the user to navigate back in the progression by selecting previous states [56, 90].

Visual Stability

None of the surveyed publications emphasized visual stability. This may be because most of them deal with a known end in the data domain and, thus, are able to visualize the high-level hierarchies first to preserve the mental map of the user. Only one approach has an unknown end [57] but shows the full evolution from the start of the computation, which preserves visual stability. However, it is challenging to scale this approach for large time frames.

Real-time Processing

None of the publications with hierarchical visualizations had a focus on real-time processing.

C.5 Challenges

Hierarchical visualizations are rarely used in progressive approaches. A weak hypothesis might be that, already non-progressive hierarchical visualizations suffer from the high visual literacy necessary to understand this visualization type. Adding progression makes the visualization even more complex and difficult to design. Currently most approaches use data chunking with a known end and extension as the visual update pattern. This is a relatively straight forward approach similar to a recursive descent. Other combinations are not researched yet. For example, the re-computation of hierarchies in an unknown end scenario. Especially, towards the end of the progression changes in the hierarchy can be a challenge. Updates applied to hierarchy levels close to the root are demanding because multiple overwriting visual updates have to be made. Not only it is a challenge to have a fast running algorithm but the design of the visual update pattern is even more demanding as the user can easily be confused. An open research direction is to use

process chunking in this case to have uncertain but more stable hierarchies throughout the progression with less distracting visual update patterns. This however, leads to possible non-deterministic hierarchies depending on the sampling order. Therefore, the open research space for progressive hierarchical visualizations is larger than for all other visualization categories.

D NETWORK VISUALIZATION



Network visualizations represent connections between data points. The goal is to understand global and local relations between entities [100]. Each entity can be a multidimensional data point, but the key feature in this category is the linking information between the entities. The most common techniques to visualize these relations are node-link diagrams or adjacency matrices [9, 68]. Progressive network

visualizations are similar to dynamic networks [8] in the sense that data changes over time. Although progressive approaches can use established solutions from dynamic networks, there are also new challenges. This category has only five approaches and all of them use a node-link representation. The exploration of progressive adjacency matrices remains an open research direction.

D.1 Progressive Processing

Here, we focus on describing the sampling strategies used to progressively visualize a large network. Two publications in this category handle the sampling progression for the network chronologically, as for dynamic graphs [26, 93]. One article introduces a hierarchical support structure to guide the sampling process [6] while another lets the user select nodes to expand the network [95]. The fifth approach samples the dataset based on user selections in other views [90].

Constraints to Data Chunking

For network visualization, data chunking is used to update the topological structure over time in all cases. This is similar to dynamic graphs, and the surveyed articles follow established approaches from this domain. Three out of four articles use data chunking to update the graph visualization with new data, while the layout is computed on-the-fly with a process chunking approach [6, 26, 93].

Constraints to Process Chunking

Frishman et al. [26] take a deeper look into the layout computation and propose a fast layout algorithm with a process chunking design. They try to maintain a global structure of the graph after new data chunks are added to preserve the mental map of the user. Another approach uses a node-link diagram to show the similarity between data in a progressive context [90]. For a given user selection, the diagram shows the entities and their similarity. The similarity measure can be edited by the user, and the node-link diagram is updated accordingly.

Custom Chunking

None of the publications with network visualizations uses custom chunking. However, the separate usage of data and process chunking is very established in this category. Dynamic layout computation with process chunking and topological progression with data chunks are utilized by most of the publications.

D.2 Data Domain

Known End

Two approaches have a known end in the data domain. One pre-computes a hierarchical data structure to separate large graphs into sub-graphs [6] while the other stores the neighborhoods of nodes [95]. Then, the graph can be explored progressively by loading selected data chunks or neighborhoods of nodes on demand.

Unknown End

The other publications have an unknown end in the data domain. Thus, the layout of the graph has to be dynamic to adjust itself to the new data. The challenge, in this case, is to preserve the layout when each new data chunk is loaded [26]. The already existing nodes should stay as close as possible to their original position to support visual stability. However, this might lead to a sub-optimal layout after many new data chunks have been loaded. Some approaches use force-directed layouts, i.e. non-deterministic simulations with a termination criterion [90, 93].

D.3 Visual Update Pattern

The visual update pattern for network visualization is very consistent across all publications.

Extension

Extension is used to add new links or nodes to the graph or add new rows and columns to an adjacency matrix. However, in our survey, we did not find any approach using matrices to visualize networks.

Overwrite

The positions and visual attributes of the nodes and links are overwritten as new progressive chunks are processed. In the dynamic graph literature, there are also cases using extension to show multiple snapshots of the network over time. This takes a considerable amount of display space but offers different analysis approaches. For progressive network visualization, this case was not explored and remains an open research avenue.

D.4 Key Properties

Uncertainty Visualization

Only one of the surveyed publications used uncertainty indicators in their network visualization by showing fading lines from nodes [95]. This indicates that there are more connections that can be sampled on demand. In theory, uncertainty can be visualized by adding a visual indicator to nodes, links, or matrix cells. Examples are fuzzy nodes or an uncertainty color map for matrix cells.

Interaction & Steering

One of the approaches used network visualization to steer the progression [95]. For the others the interactivity is limited to the selection of nodes and links and the manual change of the layout by dragging nodes.

Visual Stability

Layout stability over time is important for progressive network visualization [26, 32]. A major difficulty is preserving the visual stability when updating the layout of the graph or the order inside matrices, especially if the data domain is unknown and layouts cannot be pre-computed. It is necessary to prevent an overwhelming number of animations on updates to stabilize the mental map of the user.

Real-time Processing

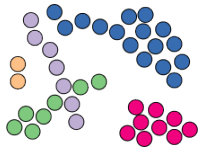
Because the layout computation is often running in parallel to new incoming data chunks, the computation has to be fast. The focus on real-time processing is important, as shown by Frishman et al. [26]. That is also why many other approaches use the fast force-directed layout algorithm. But this algorithm is not capable of generating readable layouts for very large graphs, often resulting in hairball layouts.

D.5 Challenges

Network visualizations are rarely explored in progressive visualization research. This may be due to the already thoroughly researched area of dynamic network visualizations, which is close to the progressive use case. Often, process chunking approaches are used in this context, where a layout is improved over time. Data chunking however is a different challenge. Adding new nodes and links to a network with an extension update pattern is not complicated at the first glance but the overwriting update that has to be made to the complete layout of

the network has to be designed carefully. There has to be a balance between preserving the initial layout and the mental map of the user versus the optimal layout for the current data. Especially, if we have an unknown end, the layout and the incremental changes can not be pre-computed. We think that sophisticated evaluations are needed to determine an optimal balance. Another remark is that uncertainty is almost not used in progressive network visualization but important to highlight the ongoing progression for the user. This is an open research direction, in which it would be interesting to analyze the usage of known approaches [83] transfer to progressive approaches. Finally, there is a considerable challenge if networks are dynamic in time and large in space. Then, a progressive approach has to be applied to the time dimension and the topological dimension (see Section 6).

E MULTIDIMENSIONAL VISUALIZATION



Multidimensional data can be found in many domains. Due to the high number of dimensions, it is difficult to identify patterns or correlations directly. A core technique to visualize this data type is dimensionality reduction [18]. The reduced data is commonly visualized with 2D or 3D scatterplots or parallel coordinate plots. Many progressive approaches focus on computationally intensive dimension-reduction algorithms to show early results.

Multidimensional data can also be visualized in multiple views, each focusing on a different dimension. In our survey, 33 out of 47 publications deal with multidimensional data, and 18 out of them are splitting the dimensions into different views using multiple visualization types. In this section, we want to focus on the remaining 15 publications, which primarily use one visualization to display multidimensional data because the other 18 are already mentioned in their respective visualization category. Most of these publications use scatterplots, and only a few employ bar charts or box plots.

E.1 Progressive Processing

Here, we focus on describing the sampling strategies used, which have a direct influence on the other categories.

Constraints to Data Chunking

In a multidimensional data chunking scenario, data is usually too large to be displayed at once. Therefore, sampling is used in all publications we surveyed. However, it is interesting that up to 2018, mainly random sampling is used [23, 43]. Exceptions are cases when the sampling was determined by streamed data chunks [36] or by user-defined parameters [110]. Only one approach by Jo et al. [44] addressed the sampling challenge. They implemented safeguards that are tested when the user formulates intermediate hypotheses in the early stages of the progression. Thereby, the user knows if the early assumptions still apply after more data is processed. Another approach by Hografer et al. [37] proposes a combination of default uniform sampling enhanced with steering-by-example. In this case, data selections by the user formulate an example that steers the sampling strategy. Overall, the research for multidimensional sampling strategies intensifies, while many challenges remain.

Constraints to Process Chunking

All of the publications with multidimensional process chunking are working with dimensionality reduction. Many of the most common dimensionality reduction algorithms have been modified to run progressively, including MDS [104], tSNE [45, 74], and UMAP [50]. Each of these algorithms has its specific sampling strategy that was adapted to be progressive. Steerable, progressive MDS is the oldest of these publications and uses random sampling but allows the user to select regions of interest to refine first [104]. Progressive tSNE samples the neighborhood and allows the user to set a range parameter to steer the progression [74]. Progressive UMAP uses a negative sampling approach from Word2Vec and solves the long initial delay of the algorithm by making it possible to add new data points progressively [50].

Finally, one approach enhanced the progressive MDS, tSNE, and k-means approaches with an interactive way to modify the data during the progression [49]. It is possible to move data points to other locations inside the visualization, and the algorithm will adapt to changes. The user can also fix a class label during a k-means clustering, and the algorithm will react to this input.

Custom Chunking

In the multidimensional category, only one of the publications uses data and process chunking together [7], which shows that the approaches have specialized in either direction. This approach, however, uses chunking methods for different parts of the system. Data chunking is used to load new Twitter data chunks, and then tSNE runs on a set amount of process chunks. Three remarkable ideas go beyond a hybrid of data and process chunking and motivate us to add the broader category of custom chunking.

Wong et al. present a solution to handle large data streams with MDS [105]. They look at the rate of incoming data to decide based on how the chunking process is performed. If the influx is low, MDS is used to re-process the entire dataset when new information arrives. If the influx is higher than the processing rate of the system, the MDS process is interrupted, and a custom sliding window approach is executed. The sliding window updates the visualization with the new information, and then an accumulated error is computed. The sliding window is applied until the accumulated error reaches a threshold. Then, the influx rate is checked again.

Stolper et al. propose a progressive Sequential Pattern Mining (SPAM) algorithm that has a custom queue [89]. The algorithm searches data for frequent event sequences with a depth-first traversal. The authors changed this to a breadth-first traversal for their progressive implementation. The algorithm reports patterns from shortest to longest, giving an overview of short but frequent patterns. Another feature they introduce is a queue with all the sub-patterns found during the breadth-first traversal. The user can interact with the queue and decide where the sampling should continue. It is also possible to prune parts of the queue, reducing the computation load. This promising feature could also be adopted in other domains in the future.

The third custom chunking approach by Giachelle et al. proposes a dual computation system [27]. A dynamic system progressively computes data chunks and synchronizes with a stable system at certain stages. If such a stage is reached, the user gets a notification to load the new data or continue with the current analysis. This is especially useful in use cases with large data sources where frequent progressive updates would slow down the workflow of the user. The combination of a stable system that grants the user smooth interactivity and fast response times with a dynamic system that handles the heavy computational load is a promising idea for future research.

E.2 Data Domain

Most approaches have a known data domain end, except for the non-deterministic dimension reduction algorithms.

Known End

Many publications make use of the known data domain and pre-compute data structures for sampling strategies [89] or steering recommendations [15]. Also, the straightforward feature of showing a progress bar is possible but only utilized by a few approaches [23, 37, 44].

Unknown End

The non-deterministic dimension reduction algorithms are highly dependent on the initialization and different parameters that can be set by the user. Also, the orientation of the 2D space varies for different executions, which makes it difficult to compare results [49].

E.3 Visual Update Pattern

The variety of visual update patterns is larger than for other categories. From the 15 multidimensional visualizations, 6 only use an overwriting pattern and 2 only an extension pattern, while 7 use both patterns.

Extension

The most common visual extension pattern is the addition of new data points to a scatterplot and the extension of the visual domain if the data is beyond the current view [37, 89].

Overwrite

Overwriting update patterns are more common as it is not feasible to show snapshots of progressive dimension reduction algorithms on one screen [50]. Usually, the updates are added with an animation effect to make it easier for the user to follow [44].

E.4 Key Properties

Uncertainty Visualization

Uncertainty is only shown in approaches with a known end in the data domain. The most common indicators are box plots [23] and bar charts with an opacity gradient [44]. Heatmaps are another indicator underlying scatterplots to show a trend of processed or unprocessed data [7, 89].

Interaction & Steering

Most of the multidimensional approaches use steering and interaction. We highlight a few novel ideas which stand out. One publication shows a processing queue for the progression, and the user is able to rearrange the order or prune computations [89]. Another article computes different metrics in the background and then gives user analysis recommendations and steering suggestions [15]. Finally, one approach presents a general way to extract information from selected data items to use them as an example to steer the sampling progress [37].

Visual Stability

Visual stability is not mentioned by many of the multidimensional approaches. Especially for dimension reduction algorithms, the visual changes at the start of the progression are large [49, 50]. For scatter and box plots, it is much easier to update the visualization and preserve the user's mental map.

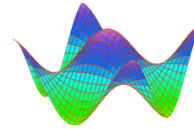
Real-time Processing

A focus on real-time processing is almost non-existent. Only three approaches that deal with data streams have requirements and recommendations on computation speed [36, 43, 105]. All other publications deal with large multidimensional data sources with the goal of reducing the long computation times from several minutes to below one minute.

E.5 Challenges

Overall, the most used category for progressive visualizations is multidimensional, with 33 out of 47 publications. In 18 cases, the multiple dimensions are broken down into separate, linked visualizations to enhance the view of the data. The benefit of introducing progression to multidimensional data is shown by the number of articles. The challenge to process large multidimensional datasets is tackled by many contributions and they show how progressive visualizations benefit the user in the form of steering capabilities and computation time. While process chunking supported many algorithms in delivering fast early results, the now progressive algorithms suffer from being unstable in the beginning, resulting in frequent overwriting update patterns which can confuse the user. It is a challenge to find the optimal balance between update frequency and the amount of visual change that needs to be made. If the visual change is minimal, the update frequency might be set higher. Data chunking requires a sampling strategy that somehow covers the relevant information in the first chunks to ensure that meaningful partial results are delivered early. There is a large potential for machine learning applications as many ML techniques are based on stochastic gradient descent algorithms that are progressive in essence. Watching an ML algorithm unfolding progressively is informative [40] and could play a role comparable to explainable AI [91].

F FIELD VISUALIZATION



Field visualization has a long research history [10] and is found in many important domains, such as medical imaging and climate simulation. Depending on the dimensions of the data points, fields are also called scalar fields, vector fields, or tensor fields. There are many ways to visualize this data in 2D and 3D, for example, with geometric glyphs, flow glyphs, isolines, isosurfaces, and streamlines.

In our literature research, we found 8 publications related to progressive visualization, which make use of field visualizations. They include velocity field, streamlet, scatter plot, isosurface visualizations, and also persistence diagrams. However, all of the approaches focus on a specific research or theory topic; none of them provides a multi-view visualization and only a few an interactive prototype. A related form of field visualization in 3D is direct volume rendering. However, this is out of the scope of this report. Nonetheless, we give a brief overview in Section 7.

F.1 Progressive Processing

Here, we focus on describing the sampling strategies used to deal with large multidimensional field data. The sampling strategy varies strongly because approaches heavily optimize special formats of the input data.

Constraints to Data Chunking

Two approaches use data chunking with a client-server infrastructure to process the large data sources on multiple workstations independently [61, 102]. The partial results are sent in data chunks to be combined in the visualization. One approach encodes the partial results with a significance map to include uncertainty information, which would later be overwritten by more significant chunks [61]. Another approach even uses coverage sampling, like Halton or Hammersley sequences, to generate separate renderings of scatterplots [33]. The separate scatterplots are then overlaid during the progression. The fourth approach with data chunking performs a grid-based sub-sampling on multi-hierarchy data [99]. The data chunks are sampled in hierarchical order until a convergence threshold is reached or the user interrupts the process.

Constraints to Process Chunking

All five publications in this category use a simulation or optimization approach. The first approach progressively computes a 3D hierarchy on a scalar or vector field [62]. The partitioning progresses until an approximation error threshold is reached. A second article also makes use of quality metrics to steer the progression. The authors progressively compute Wasserstein barycenters by using accuracy and persistence metrics [98]. Another approach simulates hurricanes and writes that sampling is not a crucial part of the process but rather the positioning of the seed points [102]. The fourth has a similar seed point approach for finding fibers in medical data and aggregating them to streamlines [86]. The fifth article uses a 2D planar sampling grid with Kriging interpolation for scattered data [16]. The Kriging interpolation is implemented in a progressive way so that the user can change weights interactively via a supporting parallel coordinates visualization.

Custom Chunking

In the field category, only one publication uses both data and process chunking [102]. However, the chunking methods are used in different parts of the system and do not qualify for a custom chunking approach. There are no interweaved forms or special chunking conditions comparable to the other custom chunking solutions we have found.

F.2 Data Domain

The data domain for field visualizations is directly coupled to the result being deterministic or non-deterministic. In all cases, the initial dataset is fully known, and different algorithms are applied.

Known End

The approaches with a deterministic result produce the same outputs with different sampling strategies. They parallelize the progression or define the quality metrics so that the same final result is achieved [33, 61, 62, 102].

Unknown End

For non-deterministic results, the sampling order or initial state is important. A simulation is dependent on the initial seed [102], or an optimization with gradient descent techniques is capable of falling into local optima due to the sampling strategy [98, 99].

F.3 Visual Update Pattern

The visual update pattern is very consistent across all publications in this category, with all of them using overwriting updates with a few exceptions of extension.

Extension

The exceptions are when new visual elements are added during an optimization process, like new markers in persistence diagrams [98, 99]. Or when separate data chunks are processed in parallel and then put together via superposition [33, 62].

Overwrite

All field visualizations we surveyed use overwriting update patterns. Each approach with optimization or parallel data processing stacks the new results over the other. Previous results are fully replaced [16] with a transition to their new position [98, 99, 102].

F.4 Key Properties

Uncertainty Visualization

Uncertainty is visualized in different ways for fields. Some of the approaches have clear quality metrics and can visualize with shape or color [61, 62]. Others can not provide a quality metric and visualize the convergence of the results with supporting scatterplots [98, 99]. One approach uses superposed scatterplots and applies blending and splatting to approximate the continuous density function [33]. Finally, Siddiqui et al. provide interactive features to explore the uncertainty of their streamline visualization for medical data [86].

Interaction & Steering

Interaction and steering are not present in older publications. Recent approaches add interactive visualizations to steer the sampling and refine regions of interest [16]. Others achieve performance speed-ups to approximate barycenters in interactive times and allow the user to interrupt the algorithm without losing previous results.

Visual Stability

The surveyed publications do not explicitly mention visual stability in their visualizations. However, the idea of superposition is an intuitive approach to preserving visual stability [33]. Also, all of the articles use an overwriting update pattern that preserves the mental map, e.g., by animation in an optimization scenario [98, 102].

Real-time Processing

Real-time processing or responses in interactive times are important for all publications. For each article, it is one of the main contributions to improve the computation speed or present faster results to users.

F.5 Challenges

The publications in this field have a different focus compared to the other visualization categories. There are more contributions that tackle the challenge of improving computation speed and creating interruptible algorithms. However, there are far fewer interactive prototypes to be used by domain experts, an open research challenge that is due to the often very large datasets used in field visualization. And with that comes the task to design interaction and steering for the visualization and evaluating the benefit of the progressiveness. A major challenge

is also the application of data chunking approaches with an unknown end. Currently, all approaches work with complete data sets and can estimate the progress of the computation. New solutions will have to be found for the design of visualizations and visual update patterns to be adaptive for an unknown end in the data domain.

6 MULTI-CATEGORY COMBINATIONS AND PROGRESSIVE DIMENSIONS

Above, we have categorized all the related work on progressive visualizations. The approaches had specific visualizations, progressive processing methods, data domains, and visual update patterns. However, most of them had a focus on one data dimension, e.g., time or geo-spatial, and the progression was oriented along this dimension. But it is also possible to have a visualization that has two progressive dimensions, like a large dynamic graph. The progression has to cover changes over time and also changes in topology while progressively loading the large graph for one point in time. There are many other use cases since the combination of any visualization category with temporal data is possible. This is also in line with the taxonomy by Munzner [65] that shows how temporal dynamics can be applied to all other data types. It is also possible to have combinations of other categories besides temporal, but we did not find any related research. However, in many combination cases, one of the visualization types or progression dimension plays the leading role in the analysis goal and thus can make use of the findings from the dominant visualization category.

The research in this direction is limited, but we think it is important as more multidimensional data with multiple progressive directions emerge in the future. We found only one publication that explores this, though it does not combine the dimensions in one progression. Badam et al. process a Twitter data stream in chronological order and visualize the data with a tSNE approach [7]. The temporal progression from the text data dimension and the topological progression from the similarity dimension are not merged. The authors introduce a waiting time for the twitter data until the tSNE has computed a layout based on interactive parameters. This idea is a first attempt at a multi-category progression, but more research in this direction is needed.

7 PROGRESSIVE VISUALIZATION IN OTHER RESEARCH DOMAINS

In this section, we highlight progressive approaches that do not fit in the scope of this survey but provide interesting ideas that could be adopted for data analysis workflows. The progression in the following articles is mostly prepared in a preprocessing step to streamline the execution. This also means that the progression is fixed and will always play out in the same way. Progressive visualization or presentation has a long research history in multimedia. 3D rendering in general, but also for video games, augmented reality, and virtual reality that need progressive data structures to save computation resources. Early approaches show progressive refinements for volume rendering to have a fast but coarse representation [54]. Rendering frameworks were extended with progressive ideas to enhance the process [70]. Many video games make use of this by creating level-of-detail (LOD) hierarchies to show far-away objects with less detail and only near objects with the highest level of detail [60].

Different image formats like JPEG were developed to allow progressive loading of the information [28], and multiple loading schemes were evaluated to find the best way to present the image on load [31]. Similar ideas were applied to video encoding [107] and the generation of progressive text descriptions for videos [106]. Finally, the research domain for rendering is still very actively making use of progressive approaches for volume rendering [13, 24, 25], isosurface propagation [58], and virtual reality approaches for surgery [46].

8 EVALUATION

Evaluation is important to verify and validate the contributions in publications. Lam et al. [53] mention seven scenarios of evaluations used in visualization. Table 1 summarizes the articles on PV that relate to each scenario.

Table 1: Evaluations related to the Seven Scenarios of Lam et al. [53] and evaluations on the impact of progressive visualization.

Scenario	References
1 Understanding Environments and Work Practices	—
2 Evaluating Visual Data Analysis and Reasoning	[7, 23, 29, 56, 71, 77, 89, 108]
3 Evaluating Communication through Visualization	—
4 Evaluating Collaborative Data Analysis	—
5 Evaluating User Performance	[7, 14, 15, 23, 29, 36, 37, 44, 48, 49, 56, 57, 61, 63, 71, 74, 75, 77, 78, 80, 82, 84, 86, 89, 90, 92–95, 109, 110]
6 Evaluating User Experience	[7, 23, 29, 36, 56, 64, 89, 90, 93]
7 Evaluating Visualization Algorithms	[6, 14, 16, 26, 33, 41, 43, 45, 48, 50, 55, 57, 72, 73, 76, 78, 86, 92, 98, 99, 102, 104, 105]
Evaluating the impact of progressive visualization	[7, 44, 71, 75, 108]

Evaluation of Progressive Features

Progressive approaches demand more evaluation metrics compared to non-progressive ones. These metrics relate to algorithms and humans. For algorithms, they are about controlling the trade-off between time and quality while maintaining low latency. They also relate to providing a useful assessment of progressive uncertainty. For humans, they are about maintaining perceptive, cognitive, and decision-making capabilities, in addition to the capability to assess progressive uncertainty (see Section 2).

Algorithm Evaluation

For quantitative evaluation, competitive analysis can be used, as with online algorithms [11], assuming the algorithm runs to the end. Competitive analysis measures how much slower an online algorithm is compared to its eager counterpart. Similarly, a progressive algorithm can be compared to its eager counterpart. It is used by Chen et al. [14], who, like other articles, acknowledge that this measure is misleading for progressive systems.

One important assumption of progressive systems is that a useful result can be obtained before the algorithm runs to completion. In that case, a good measure is when the quality of the progressive algorithm is good enough to make a decision, and what is the chance of making a bad decision then? Answering these questions requires statistics that belong to the family of sequential analysis [101] used by [71]. A few articles also use traditional tests but acknowledge they are inaccurate [44, 71].

User Performance and Experience

For qualitative evaluation, new methods have to be developed. The metric should consider the trade-off between time and quality, rating how fast and accurate the decision-making process is, but also how confident users are making the decision. The goal is to evaluate the cognitive process of the user to document the qualitative value and possible pitfalls of the progressive approach. Most of the approaches in our survey resort to established evaluation methods.

Users also need to deal with the visual instability inherent to some progressive algorithms; they need to detect important changes during the progression but avoid being distracted by unimportant changes. Jo et al. [45] mention a strategy where their progressive tSNE algorithm notifies the user that triggering a new “exaggeration” phase would improve the overall quality but change the layout and potentially disrupt the user’s mental map. More work is needed to find strategies, interactive or not, to address progressive instability issues.

Two articles also mention *speculative exploration* [44, 64] when users assume (speculate) that what the progressive visualization shows at some point will remain valid; users continue exploring with that assumption in mind. This saves time if the assumption is true, but needs potentially expensive counter-measures to detect that the assumption does not hold and to forget the insights found with the false assumption in mind. This issue is specific to progressive visualization and needs more research to understand how humans can forget some of their insights.

Progressive uncertainty visualization is important [35, 89] but has only been tackled for simple visualizations such as bar charts [22, 35, 71] or using an aggregated “quality” measure [7]. More work is needed to support it for general progressive visualization techniques; classical uncertainty visualization techniques [69] might work but are not applicable to all the visualization techniques yet and have not been evaluated in a progressive setting when the uncertainty changes dynamically.

Categorization

We documented which types of evaluations were performed in the publications. After an aggregation step, we categorized the evaluations into the following seven types: The most used evaluation method is the user performance evaluation (31). This covers use cases and non-expert user evaluations with a focus on interviews, task completion tests, or perception and cognition tasks. The second most used evaluation type is algorithm evaluation, with 23 publications. Many approaches report a speed-up in computation and validate this with a technical evaluation compared to other state-of-the-art approaches. Nine publications made user experience evaluations by going deeper into the application domain and testing their tool with experts in their workflow. The evaluation of data analysis and decision-making was part of seven publications. They analyzed how progressive visualization can improve the early decision-making process. The evaluation types “Understanding Environments and Work Practices”, “Evaluating Communication through Visualization”, and “Evaluating Collaborative Data Analysis” are not addressed in the reviewed literature and remain one of the research gaps for PVA. The categorization for each article can be accessed in our visual survey browser at visualsevery.net/pva.

9 DISCUSSION AND CONCLUSION

9.1 Statistics

In this section, we summarize the most evident trends based on our taxonomy and the categorization of publications seen in Table 2. Overall, the publication rate for progressive visualizations is on an upwards trend, with 8 publications from 1999 to 2009, 29 from 2010 to 2019, and already 11 from 2020 to 2023. Most of the articles use multidimensional data and visualizations (33 out of 48). However, more than half of those (18) split the dimensions and use different linked visualizations. The least used progressive visualization types are hierarchical and network visualizations. For progressive processing, more approaches use data chunking than process chunking (37 vs. 22). Similarly, but more interesting, 36 approaches use datasets with a known end compared to 17 approaches with an unknown end in the data domain. For the other categories, the differences are not significant, except that steering is most often added as a key property (35 out of 48), while uncertainty and real-time processing are rarely used (20 or 10 out of 48).

9.2 Discussion

Our literature review showed that there are many open research questions in progressive visualization. In general, for progressive approaches, new ideas and methods for the presentation of the research

Table 2: Categorization of Related Works with Progressive Visualizations from 1999 to 2023

Publication	Year	Visualization					Processing			Data		Update		Key Properties				
		Temporal	Geo-Spatial	Hierarchical	Network	Multidim.	Field	Data Chunk	Process Chunk	Custom Chunk	Known End	Unknown End	Extension	Overwrite	Uncertainty	Steering	Visual Stability	Real-time
Interactive Data Analysis [34]	1999		•					•	•			•	•	•	•	•		
EVITA [61]	2001							•					•			•		
Dynamic data streams [105]	2003					•		•	•		•		•				•	
Fast Scalar- and Vectorfield [62]	2003								•			•	•					
Multidimensional Scaling [104]	2004					•			•				•		•	•		
Bucket of text into a pipe [36]	2005					•		•				•	•		•	•	•	
Dynamic Graph Drawing [26]	2008							•	•			•	•			•		
Large Hierarchies [79]	2009			•				•	•				•		•	•		
Large Graph Exploration [95]	2009							•				•		•	•	•		
Hurricane Visualization [102]	2011		•					•	•			•	•		•			
Progressive Splatting [33]	2011					•	•	•				•	•	•		•		
Trust Me, I'm Partially Right [23]	2012					•		•				•	•	•	•			
Parallel coordinates [80]	2012			•		•		•	•				•		•	•		
VisReduce [41]	2013	•				•		•					•		•			
Model for Incremental Vis [3]	2013		•					•					•		•			
Response Surfaces [16]	2013							•				•	•		•		•	
User-driven visual exploration [89]	2014					•		•		•		•	•	•	•	•		
Dive in! [29]	2014	•						•	•			•	•	•	•	•		
Online text streams [57]	2015	•		•		•		•				•	•		•		•	
Process Model Incremental Vis [82]	2016		•						•				•		•	•	•	
Interactive Analytics Processes [92]	2016		•			•		•				•	•	•	•	•		
Steerable tSNE [74]	2017					•			•				•		•			
Steering the craft [7]	2017					•		•	•			•	•	•	•	•		
I've Seen "Enough" [77]	2017	•				•		•	•			•	•				•	
Pattern Discovery [110]	2017					•		•				•	•		•			
Deepeyes [73]	2017	•				•		•	•			•	•	•		•		
Swifttuna [43]	2017					•		•					•		•	•	•	
PIVE [49]	2017					•			•				•		•	•		
Sampling for scalable VA [52]	2017		•			•		•				•	•	•	•			
TPFlow [56]	2018	•	•	•		•			•			•	•	•	•	•	•	
Pattern Mining [78]	2018	•				•		•				•	•		•			
Sub-Graph Explorations [6]	2019					•		•	•			•	•				•	
Incremental Evaluation [27]	2019					•			•			•			•			
Netcapvis [93]	2019	•				•		•	•			•	•		•	•		
Proreveal [44]	2019					•		•					•		•	•	•	
P5 [55]	2019	•				•		•				•	•		•	•		
Datasite [15]	2019					•		•				•	•		•			
Wasserstein Barycenters [98]	2020					•			•			•	•		•		•	
UMAP [50]	2020					•			•				•			•		
Streaming performance [48]	2020	•				•		•				•	•	•	•	•	•	
ProBGP [94]	2021	•	•					•				•	•		•	•		
Pyramid-based Scatterplots [14]	2021		•			•		•				•	•				•	
Diffusion Tensor Imaging [86]	2021					•	•		•			•		•	•			
Scalar Field Topology [99]	2021					•			•			•	•		•			
Hardware System Monitoring [84]	2022	•				•		•				•	•				•	
Steering-by-Example [37]	2022					•		•				•			•			
Hierarchical Topic Refinement [90]	2023	•	•	•	•	•		•				•	•		•			
PEViz [109]	2023	•	•			•		•	•			•	•		•	•		
Sum: 48		14	11	5	5	33	8	37	22	3	36	17	33	43	20	35	24	10

results would be helpful. For the reader of the publications, it is difficult to imagine the progressive behavior of the approach if there is no video, online prototype, or animation. Some articles provide links, but it would be beneficial to allow the inclusion of new types of media in PDFs, like animated images or videos. We identified multiple research gaps on different levels of detail. While we described the individual research challenges from each visualization type in Section 5, there are some general challenges that we want to highlight here:

- Update and overview of current libraries: Most of the popular visualization libraries are not designed for progressive updates; they are difficult to configure or provide inappropriate mechanisms to deal with the progressive updates.
- Visual stability and accuracy: several visualization techniques can dramatically change their appearance when updated progressively, e.g., tree maps and sorted bar charts. There is currently no general approach to mitigate this issue.
- Visualization design: Designing progressive visualization is different for each use case. Every combination of our taxonomy categories has visual mappings that are more appropriate. For example, if the end is unknown mapping color on a category is not recommended as it may have too many values at the end of the progression. A exhaustive study is necessary to evaluate visual mappings for progressive visualization.
- Evaluation: As outlined above, many visualization techniques have not been used in a progressive setting and require design adjustments backed by evaluations. The evaluation of progressive approaches would also benefit from research on new metrics and streamlining sequential testing methods. Additionally, it would be interesting to evaluate the difference of insights a user gets by using data chunking vs. process chunking.

9.3 Conclusion

We surveyed literature on progressive visualization and derived a new taxonomy. Our taxonomy is based on findings from other taxonomies and extended by features specific to progressive approaches. We provided detailed descriptions of the categorization of the surveyed publications and highlight open research questions. We also reported on the application areas of the articles and discussed present and future evaluation methods for progressive applications. With our survey, the current research state for progressive visualizations is condensely summarized and hopefully supports the research community to investigate the various open challenges.

Progressive visualization is one of the few paradigms that can scale visualization in terms of data size, and visual analytics in terms of analytics complexity and power. More research should be devoted to making it usable for the many application domains requiring scalability. There are many opportunities in both research and application to transfer existing methods to different data types and to more visualization techniques. This survey describes most of the inspiration to start from. Finally, we have not addressed the progressive analytics aspect, which is another wide world that needs more research and experiments too. When more progressive visual analytics methods are developed, their results will be progressive visualizations that will remain a fundamental component of scalable data exploration.

ACKNOWLEDGMENTS

The authors wish to thank the reviewers for their valuable feedback that improved the survey in multiple areas and made it more inclusive for future findings in progressive visualization and analytics. This project was supported in part by the MUR PRIN 2022 Project No. 202248FWFS “Discount quality for responsible data science: Human-in-the-Loop for quality data” within the NextGenerationEU Programme.

REFERENCES

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, 2011. doi: 10.1007/978-0-85729-079-3 6

[2] M. Angelini, T. May, G. Santucci, and H. Schulz. On quality indicators for progressive visual analytics. In *EuroVA@EuroVis*, pp. 25–29, 2019. doi: 10.2312/eurova.20191120 4

[3] M. Angelini and G. Santucci. Modeling incremental visualizations. In *EuroVA@EuroVis*, 2013. doi: 10.2312/PE.EuroVAST.EuroVA13.013-017 4, 7, 8, 15

[4] M. Angelini and G. Santucci. On visual stability and visual consistency for progressive visual analytics. In *(VISIGRAPP IVAPP)*, pp. 335–341, 2017. doi: 10.5220/0006269703350341 3

[5] M. Angelini, G. Santucci, H. Schumann, and H. Schulz. A review and characterization of progressive visual analytics. *Informatics*, 5(3):31, 2018. doi: 10.3390/informatics5030031 1, 2, 3, 4, 5

[6] S. Armstrong, K. Bruhwiler, and S. L. Pallickara. Rapid, progressive sub-graph explorations for interactive visual analytics over large-scale graph datasets. In *IEEE/ACM BDCAT*, pp. 1–10, 2019. doi: 10.1145/3365109.3368793 10, 14, 15

[7] S. K. Badam, N. Elmqvist, and J. Fekete. Steering the craft: UI elements and visualizations for supporting progressive visual analytics. *CGF*, 36(3):491–502, 2017. doi: 10.1111/cgf.13205 4, 11, 12, 13, 14, 15

[8] F. Beck, M. Burch, S. Diehl, and D. Weiskopf. A taxonomy and survey of dynamic graph visualization. *CGF*, 36(1):133–159, 2017. doi: 10.1111/cgf.12791 10

[9] M. Behrisch, B. Bach, N. H. Riche, T. Schreck, and J. Fekete. Matrix reordering methods for table and network visualization. *CGF*, 35(3):693–716, 2016. doi: 10.1111/cgf.12935 10

[10] C. Bi, L. Yang, Y. Duan, and Y. Shi. A survey on visualization of tensor field. *J. Vis.*, 22(3):641–660, 2019. doi: 10.1007/s12650-019-00555-8 12

[11] A. Borodin and R. El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998. 3, 14

[12] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. *SIAM Rev.*, 60(2):223–311, 2018. doi: 10.1137/16M1080173 3

[13] S. P. Callahan, L. Bavoil, V. Pascucci, and C. T. Silva. Progressive volume rendering of large unstructured grids. *IEEE TVCG*, 12(5):1307–1314, 2006. doi: 10.1109/TVCG.2006.171 13

[14] X. Chen, J. Zhang, C. Fu, J. Fekete, and Y. Wang. Pyramid-based scatterplots sampling for progressive and streaming data visualization. *IEEE TVCG*, 28(1):593–603, 2022. doi: 10.1109/TVCG.2021.3114880 7, 8, 14, 15

[15] Z. Cui, S. K. Badam, M. A. Yalçın, and N. Elmqvist. Datasite: Proactive visual data exploration with computation of insight-based recommendations. *Inf. Vis.*, 18(2), 2019. doi: 10.1177/1473871618806555 11, 12, 14, 15

[16] I. Demir and R. Westermann. Progressive high-quality response surfaces for visually guided sensitivity analysis. *CGF*, 32(3):21–30, 2013. doi: 10.1111/cgf.12089 12, 13, 14, 15

[17] D. J. Duke, M. Wallace, R. Borgo, and C. Runciman. Fine-grained visualization pipelines and lazy functional languages. *IEEE TVCG*, 12(5):973–980, 2006. doi: 10.1109/TVCG.2006.145 2

[18] G. Dzemyda, O. Kurasova, and J. Zilinskas. Multidimensional data visualization. *Optimization and its applications*, 75(122):10–5555, 2013. doi: 10.1007/978-1-4419-0236-8 11

[19] N. Elmqvist and J. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE TVCG*, 16(3):439–454, 2010. doi: 10.1109/TVCG.2009.84 8

[20] J. Fekete, D. Fisher, A. Nandi, and M. Sedlmair. Progressive data analysis and visualization (dagstuhl seminar 18411), 2018. doi: 10.4230/DagRep.8.10.1 2, 4

[21] J.-D. Fekete and R. Primet. Progressive analytics: A computation paradigm for exploratory data analysis. *CoRR*, abs/1607.05162, 2016. 1, 2

[22] D. Fisher, S. M. Drucker, and A. C. König. Exploratory visualization involving incremental, approximate database queries and uncertainty. *IEEE Computer Graphics and Applications*, 32(4):55–62, 2012. doi: 10.1109/MCG.2012.48 14

[23] D. Fisher, I. O. Popov, S. M. Drucker, and m. c. schraefel. Trust me, i’m partially right: incremental visualization lets analysts explore large datasets faster. In *CHI*, pp. 1673–1682, 2012. doi: 10.1145/2207676.2208294 1, 4, 11, 12, 14, 15

[24] S. Frey and T. Ertl. Progressive direct volume-to-volume transformation. *IEEE TVCG*, 23(1):921–930, 2017. doi: 10.1109/TVCG.2016.2599042 13

- [25] S. Frey, F. Sadlo, K. Ma, and T. Ertl. Interactive progressive visualization with space-time error control. *IEEE TVCG*, 20(12):2397–2406, 2014. doi: [10.1109/TVCG.2014.2346319](https://doi.org/10.1109/TVCG.2014.2346319) 1, 13
- [26] Y. Frishman and A. Tal. Online dynamic graph drawing. *IEEE TVCG*, 14(4):727–740, 2008. doi: [10.1109/TVCG.2008.11101415](https://doi.org/10.1109/TVCG.2008.11101415)
- [27] F. Giachelle and G. Silvello. A progressive visual analytics tool for incremental experimental evaluation. In *Italian Information Retrieval Workshop*, pp. 2–5, 2019. 11, 15
- [28] J. M. Gilbert and R. W. Brodersen. Globally progressive interactive web delivery. In *INFOCOM*, pp. 1291–1299. IEEE Computer Society, 1999. doi: [10.1109/INFCOM.1999.752147](https://doi.org/10.1109/INFCOM.1999.752147) 13
- [29] M. Glueck, A. Khan, and D. J. Wigdor. Dive in!: enabling progressive loading for real-time navigation of data visualizations. In *CHI*, pp. 561–570, 2014. doi: [10.1145/2556288.2557195](https://doi.org/10.1145/2556288.2557195) 6, 7, 14, 15
- [30] H. Griethe and H. Schumann. The visualization of uncertain data: Methods and problems. In *SimVis '06*, pp. 143–156. SCS Publishing House, 2006. 4
- [31] C. Harrison, A. K. Dey, and S. E. Hudson. Evaluation of progressive image loading schemes. In *CHI*, pp. 1549–1552. ACM, 2010. doi: [10.1145/1753326.1753557](https://doi.org/10.1145/1753326.1753557) 13
- [32] A. Hayashi, T. Matsubayashi, T. Hoshida, and T. Uchiyama. Initial positioning method for online and real-time dynamic graph drawing of time varying data. In *IV*, pp. 435–444. IEEE Computer Society, 2013. doi: [10.1109/IV.2013.5710](https://doi.org/10.1109/IV.2013.5710)
- [33] J. Heinrich, S. Bachthaler, and D. Weiskopf. Progressive splatting of continuous scatterplots and parallel coordinates. *CGF*, 30(3):653–662, 2011. doi: [10.1111/j.1467-8659.2011.01914.x](https://doi.org/10.1111/j.1467-8659.2011.01914.x) 12, 13, 14, 15
- [34] J. M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, and P. J. Haas. Interactive data analysis: The control project. *Computer*, 32(8):51–59, 1999. doi: [10.1109/2.781635](https://doi.org/10.1109/2.781635) 1, 7, 8, 15
- [35] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. In *SIGMOD*, pp. 171–182, 1997. doi: [10.1145/253260.253291](https://doi.org/10.1145/253260.253291) 8, 14
- [36] E. G. Hetzler, V. L. Crow, D. A. Payne, and A. Turner. Turning the bucket of text into a pipe. In *InfoVis 2005*, pp. 89–94, 2005. doi: [10.1109/INFVIS.2005.1532133](https://doi.org/10.1109/INFVIS.2005.1532133) 11, 12, 14, 15
- [37] M. Hogräfer, M. Angelini, G. Santucci, and H. Schulz. Steering-by-example for progressive visual analytics. *ACM Trans. Intell. Syst. Technol.*, 13(6):96:1–96:26, 2022. doi: [10.1145/3531229](https://doi.org/10.1145/3531229) 11, 12, 14, 15
- [38] M. Hogräfer and H.-J. Schulz. Tailorable sampling for progressive visual analytics. *TVCG*, 2023. 5, 8
- [39] M. Hogräfer, J. Burkhardt, and H.-J. Schulz. A Pipeline for Tailored Sampling for Progressive Visual Analytics. In *EuroVA@EuroVis*, 2022. doi: [10.2312/eurova.20221079](https://doi.org/10.2312/eurova.20221079) 5
- [40] T. Höllt, N. Pezzotti, V. van Unen, F. Koning, E. Eiseemann, B. P. F. Lelieveldt, and A. Vilanova. Cytosplore: Interactive immune cell phenotyping for large single-cell datasets. *CGF*, 35(3):171–180, 2016. doi: [10.1111/cgf.12893](https://doi.org/10.1111/cgf.12893) 3, 12
- [41] J. Im, F. G. Villegas, and M. J. McGuffin. Visreduce: Fast and responsive incremental information visualization of large datasets. In *(IEEE BigData 2013)*, pp. 25–32, 2013. doi: [10.1109/BigData.2013.6691710](https://doi.org/10.1109/BigData.2013.6691710) 6, 7, 14, 15
- [42] A. Jena, U. Engelke, T. Dwyer, V. Raiamanickam, and C. Paris. Uncertainty visualisation: an interactive visual survey. In *PacificVis*, pp. 201–205. IEEE, Institute of Electrical and Electronics Engineers, 2020. doi: [10.1109/PacificVis48177.2020.10144](https://doi.org/10.1109/PacificVis48177.2020.10144)
- [43] J. Jo, W. Kim, S. Yoo, B. H. Kim, and J. Seo. Swifttuna: Responsive and incremental visual exploration of large-scale multidimensional data. In *IEEE PacificVis*, pp. 131–140, 2017. doi: [10.1109/PACIFICVIS.2017.8031587](https://doi.org/10.1109/PACIFICVIS.2017.8031587) 11, 12, 14, 15
- [44] J. Jo, S. L'Yi, B. Lee, and J. Seo. Proreveal: Progressive visual analytics with safeguards. *IEEE TVCG*, 27(7):3109–3122, 2021. doi: [10.1109/TVCG.2019.2962404](https://doi.org/10.1109/TVCG.2019.2962404) 11, 12, 14, 15
- [45] J. Jo, J. Seo, and J. Fekete. PANENE: A progressive algorithm for indexing and querying approximate k-nearest neighbors. *IEEE TVCG*, 26(2):1347–1360, 2020. doi: [10.1109/TVCG.2018.2869149](https://doi.org/10.1109/TVCG.2018.2869149) 2, 11, 14
- [46] M. Kamarianakis, A. Protopsaltis, D. Angelis, M. Tamiolakis, and G. Papiannakis. Progressive tearing and cutting of soft-bodies in high-performance virtual reality. *CoRR*, abs/2209.08531, 2022. doi: [10.48550/arXiv.2209.08531](https://doi.org/10.48550/arXiv.2209.08531) 13
- [47] D. A. Keim. Information visualization and visual data mining. *IEEE TVCG*, 8(1):1–8, 2002. doi: [10.1109/2945.981847](https://doi.org/10.1109/2945.981847) 1, 5
- [48] S. P. Kesavan, T. Fujiwara, J. K. Li, C. Ross, M. Mubarak, C. D. Carothers, R. B. Ross, and K. Ma. A visual analytics framework for reviewing streaming performance data. In *PacificVis*, pp. 206–215, 2020. doi: [10.1109/PacificVis48177.2020.9280674](https://doi.org/10.1109/PacificVis48177.2020.9280674) 6, 7, 14, 15
- [49] H. Kim, J. Choo, C. Lee, H. Lee, C. K. Reddy, and H. Park. PIVE: per-iteration visualization environment for real-time interactions with dimension reduction and clustering. In *AAAI Conference on Artificial Intelligence*, pp. 1001–1009, 2017. doi: [10.1609/aaai.v31i1.10628](https://doi.org/10.1609/aaai.v31i1.10628) 1, 11, 12, 14, 15
- [50] H. Ko, J. Jo, and J. Seo. Progressive uniform manifold approximation and projection. In *EuroVis 2020 - Short*, pp. 133–137, 2020. doi: [10.2312/evs.20201061](https://doi.org/10.2312/evs.20201061) 11, 12, 14, 15
- [51] M.-J. Kraak and F. Ormeling. *Cartography: visualization of geospatial data*. CRC Press, 2020. doi: [10.1201/9780429464195](https://doi.org/10.1201/9780429464195) 7
- [52] B. C. Kwon, J. Verma, P. J. Haas, and Ç. Demiralp. Sampling for scalable visual analytics. *IEEE Computer Graphics and Applications*, 37(1):100–108, 2017. doi: [10.1109/MCG.2017.611715](https://doi.org/10.1109/MCG.2017.611715)
- [53] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *IEEE TVCG*, 18(9):1520–1536, 2012. doi: [10.1109/TVCG.2011.279](https://doi.org/10.1109/TVCG.2011.279) 13, 14
- [54] D. Laur and P. Hanrahan. Hierarchical splatting: a progressive refinement algorithm for volume rendering. In *SIGGRAPH*, pp. 285–288. ACM, 1991. doi: [10.1145/122718.122748](https://doi.org/10.1145/122718.122748) 13
- [55] J. K. Li and K. Ma. P5: portable progressive parallel processing pipelines for interactive data analysis and visualization. *IEEE TVCG*, 26(1):1151–1160, 2020. doi: [10.1109/TVCG.2019.2934537](https://doi.org/10.1109/TVCG.2019.2934537) 6, 7, 14, 15
- [56] D. Liu, P. Xu, and L. Ren. Tpflow: Progressive partition and multidimensional pattern extraction for large-scale spatio-temporal data analysis. *IEEE TVCG*, 25(1):1–11, 2019. doi: [10.1109/TVCG.2018.2865018](https://doi.org/10.1109/TVCG.2018.2865018) 7, 8, 9, 14, 15
- [57] S. Liu, J. Yin, X. Wang, W. Cui, K. Cao, and J. Pei. Online visual analytics of text streams. *IEEE TVCG*, 22(11):2451–2466, 2016. doi: [10.1109/TVCG.2015.2509990](https://doi.org/10.1109/TVCG.2015.2509990) 6, 7, 8, 9, 14, 15
- [58] Z. Liu, A. Finkelstein, and K. Li. Improving progressive view-dependent isosurface propagation. *Comput. Graph.*, 26(2):209–218, 2002. doi: [10.1016/S0097-8493\(02\)00052-3](https://doi.org/10.1016/S0097-8493(02)00052-3) 13
- [59] Z. Liu and J. Heer. The effects of interactive latency on exploratory visual analysis. *IEEE TVCG*, 20(12):2122–2131, 2014. doi: [10.1109/TVCG.2014.2346452](https://doi.org/10.1109/TVCG.2014.2346452) 2
- [60] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson, and R. Huebner. *Level of detail for 3D graphics*. Morgan Kaufmann, 2003. doi: [10.1016/B978-1-55860838-2/50000-5](https://doi.org/10.1016/B978-1-55860838-2/50000-5) 13
- [61] R. Machiraju, J. E. Fowler, D. Thompson, B. Soni, and W. Schroeder. Evita—efficient visualization and interrogation of tera-scale data. In *Data mining for scientific and engineering applications*, pp. 257–279. Springer, 2001. doi: [10.1007/978-1-4615-1733-7_15](https://doi.org/10.1007/978-1-4615-1733-7_15) 12, 13, 14, 15
- [62] T. May, S. Schneider, M. Schmidt, and V. Luckas. Fast scalar- & vectorfield visualization using a new progressive grid class. In *SimVis*, pp. 89–102, 2003. 12, 13, 15
- [63] L. Micallef, H. Schulz, M. Angelini, M. Aupetit, R. Chang, J. Kohlhammer, A. Perer, and G. Santucci. The human user in progressive visual analytics. In *EuroVis 2019 - Short Papers*, pp. 19–23, 2019. doi: [10.2312/evs.20191164](https://doi.org/10.2312/evs.20191164) 4, 14
- [64] D. Moritz, D. Fisher, B. Ding, and C. Wang. Trust, but verify: Optimistic visualizations of approximate queries for exploring big data. In *CHI*, pp. 2904–2915. ACM, 2017. doi: [10.1145/3025453.3025456](https://doi.org/10.1145/3025453.3025456) 1, 14
- [65] T. Munzner. *Visualization Analysis and Design*. A K Peters, 2014. 1, 5, 13
- [66] S. Muthukrishnan. Data streams: Algorithms and applications. *Found. Trends Theor. Comput. Sci.*, 1(2):117–236, 2005. doi: [10.1561/04000000023](https://doi.org/10.1561/04000000023)
- [67] J. Nielsen. Powers of 10: Time scales in user experience, Oct 2009. 2
- [68] C. Nobre, M. D. Meyer, M. Streit, and A. Lex. The state of the art in visualizing multivariate networks. *CGF*, 38(3):807–832, 2019. doi: [10.1111/cgf.13728](https://doi.org/10.1111/cgf.13728) 10
- [69] L. Padilla, M. Kay, and J. Hullman. *Uncertainty Visualization*, pp. 1–18. John Wiley & Sons, Ltd, 2021. doi: [10.1002/9781118445112.stat08296](https://doi.org/10.1002/9781118445112.stat08296) 14
- [70] A. Pang and N. Alper. Mix&match: A construction kit for visualization. In *Proceedings Visualization '94*, pp. 302–309. IEEE, 1994. 13
- [71] A. Patil, G. Richer, C. Jermaine, D. Moritz, and J. Fekete. Studying early decision making with progressive bar charts. *IEEE TVCG*, 29(1):407–417, 2022. doi: [10.1109/TVCG.2022.3209426](https://doi.org/10.1109/TVCG.2022.3209426) 4, 14
- [72] N. Pezzotti, T. Höllt, B. P. F. Lelieveldt, E. Eiseemann, and A. Vilanova. Hierarchical stochastic neighbor embedding. *CGF*, 35(3):21–30, 2016.

- doi: 10.1111/cgf.12878 14
- [73] N. Pezzotti, T. Höllt, J. C. van Gemert, B. P. F. Lelieveldt, E. Eisemann, and A. Vilanova. Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE TVCG*, 24(1):98–108, 2018. doi: 10.1109/TVCG.2017.2744358 6, 7, 14, 15
- [74] N. Pezzotti, B. P. F. Lelieveldt, L. van der Maaten, T. Höllt, E. Eisemann, and A. Vilanova. Approximated and user steerable tsne for progressive visual analytics. *IEEE TVCG*, 23(7):1739–1752, 2017. doi: 10.1109/TVCG.2016.2570755 11, 14, 15
- [75] M. Procopio, A. Mosca, C. Scheidegger, E. Wu, and R. Chang. Impact of cognitive biases on progressive visualization. *IEEE TVCG*, 28(9):3093–3112, 2022. doi: 10.1109/TVCG.2021.3051013 4, 14
- [76] M. Procopio, C. Scheidegger, E. Wu, and R. Chang. Selective wander join: Fast progressive visualizations for data joins. *Informatics*, 6(1):14, 2019. doi: 10.3390/informatics6010014 14
- [77] S. Rahman, M. Aliakbarpour, H. Kong, E. Blais, K. Karahalios, A. G. Parameswaran, and R. Rubinfeld. I’ve seen “enough”: Incrementally improving visualizations to support rapid decision making. *Proc. VLDB Endow.*, 10(11):1262–1273, 2017. doi: 10.14778/3137628.3137637 6, 7, 14, 15
- [78] V. Raveneau, J. Blanchard, and Y. Prié. Progressive sequential pattern mining: steerable visual exploration of patterns with PPMT. In *Visualization in Data Science (VDS at IEEE VIS 2018)*, Oct. 2018. 6, 7, 14, 15
- [79] R. Rosenbaum and B. Hamann. Progressive presentation of large hierarchies using treemaps. In *ISVC*, pp. 71–80, 2009. doi: 10.1007/978-3-642-10520-3_7 8, 9, 15
- [80] R. Rosenbaum, J. Zhi, and B. Hamann. Progressive parallel coordinates. In *PacificVis 2012*, pp. 25–32, 2012. doi: 10.1109/PacificVis.2012.6183570 8, 9, 14, 15
- [81] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. P. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE TVCG*, 20(12):1604–1613, 2014. doi: 10.1109/TVCG.2014.2346481 2
- [82] H. Schulz, M. Angelini, G. Santucci, and H. Schumann. An enhanced visualization process model for incremental visualization. *IEEE TVCG*, 22(7):1830–1842, 2016. doi: 10.1109/TVCG.2015.2462356 5, 7, 8, 14, 15
- [83] J. Schwank and S. Schöffel. Visualizing uncertainty in node-link diagrams - a user study. In *Advances in Usability and User Experience*, pp. 495–507, 2017. doi: 10.1007/978-3-319-60492-3_47 11
- [84] Shilpika, T. Fujiwara, N. Sakamoto, J. Nonaka, and K. Ma. A visual analytics approach for hardware system monitoring with streaming functional data analysis. *IEEE TVCG*, 28(6):2338–2349, 2022. doi: 10.1109/TVCG.2022.3165348 6, 7, 14, 15
- [85] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Symposium on Visual Languages*, pp. 336–343. IEEE Computer Society, 1996. doi: 10.1109/VL.1996.545307 1, 5
- [86] F. Siddiqui, T. Höllt, and A. Vilanova. A progressive approach for uncertainty visualization in diffusion tensor imaging. *CGF*, 40(3):411–422, 2021. doi: 10.1111/cgf.14317 12, 13, 14, 15
- [87] M. Sondag, W. Meulemans, C. Schulz, K. Verbeek, D. Weiskopf, and B. Speckmann. Uncertainty treemaps. In *PacificVis 2020*, pp. 111–120. IEEE, 2020. doi: 10.1109/PacificVis48177.2020.7614 9
- [88] D. Song and E. J. Golin. Fine-grain visualization algorithms in dataflow environments. In *IEEE Vis 1993*, pp. 126–134. IEEE Computer Society, 1993. doi: 10.1109/VISUAL.1993.398860 1
- [89] C. D. Stolper, A. Perer, and D. Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE TVCG*, 20(12):1653–1662, 2014. doi: 10.1109/TVCG.2014.2346574 4, 11, 12, 14, 15
- [90] Y. Tao and Y. Tang. Progressive visual analysis of traffic data based on hierarchical topic refinement and detail analysis. *Journal of Visualization*, 26(2):367–384, 2023. doi: 10.1007/s12650-022-00879-y 6, 7, 8, 9, 10, 14, 15
- [91] Tensorflow playground. Github, 2016. 12
- [92] C. Turkay, E. Kaya, S. Balcişoy, and H. Hauser. Designing progressive and interactive analytics processes for high-dimensional data analysis. *IEEE TVCG*, 23(1):131–140, 2017. doi: 10.1109/TVCG.2016.2598470 7, 8, 14, 15
- [93] A. Ulmer, D. Sessler, and J. Kohlhammer. Netcapvis: Web-based progressive visual analytics for network packet captures. In *IEEE VizSec*, pp. 1–10, 2019. doi: 10.1109/VizSec48167.2019.9161633 6, 7, 10, 14, 15
- [94] A. Ulmer, D. Sessler, and J. Kohlhammer. ProBGP: Progressive visual analytics of live BGP updates. *CGF*, 40(3):37–48, 2021. doi: 10.1111/cgf.14287 6, 7, 8, 14, 15
- [95] F. van Ham and A. Perer. “search, show context, expand on demand”: Supporting large graph exploration with degree-of-interest. *IEEE TVCG*, 15(6):953–960, 2009. doi: 10.1109/TVCG.2009.108 10, 14, 15
- [96] R. van Liere, J. D. Mulder, and J. J. van Wijk. Computational steering. *Future Generation Computer Systems*, 12(5):441–450, 1997. HPCN96. doi: 10.1016/S0167-739X(96)00029-5 2
- [97] E. Ventocilla and M. Riveiro. A model for the progressive visualization of multidimensional data structure. In *VISGRAPP*, pp. 203–226, 2019. doi: 10.1007/978-3-030-41590-7_9 2
- [98] J. Vidal, J. Budin, and J. Tierny. Progressive wasserstein barycenters of persistence diagrams. *IEEE TVCG*, 26(1):151–161, 2020. doi: 10.1109/TVCG.2019.2934256 12, 13, 14, 15
- [99] J. Vidal, P. Guillou, and J. Tierny. A progressive approach to scalar field topology. *IEEE TVCG*, 27(6):2833–2850, 2021. doi: 10.1109/TVCG.2021.3060500 12, 13, 14, 15
- [100] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J. Fekete, and D. W. Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. *CGF*, 30(6):1719–1749, 2011. doi: 10.1111/j.1467-8659.2011.01898.x 10
- [101] A. Wald. *Sequential analysis*. Courier Corporation, 2004. 14
- [102] S. R. Wang, R. You, S. Li, Y. S. Chen, and G. P. Wang. Realtime visualization of hurricane in distributed environment. In *Applied Mechanics and Materials*, vol. 40, pp. 948–954. Trans Tech Publ, 2011. doi: 10.4028/www.scientific.net/AMM.40-41.948 7, 8, 12, 13, 14, 15
- [103] Y. Wang, Y. Wang, X. Chen, Y. Zhao, F. Zhang, E. Wu, C. Fu, and X. Yu. OM3: an ordered multi-level min-max representation for interactive progressive visualization of time series. *Proc. ACM Manag. Data*, 1(2):145:1–145:24, 2023. doi: 10.1145/3589290 2
- [104] M. Williams and T. Munzner. Steerable, progressive multidimensional scaling. In *InfoVis 2004*, pp. 57–64, 2004. doi: 10.1109/INFVIS.2004.60 11, 14, 15
- [105] P. C. Wong, H. Foote, D. Adams, W. Cowley, and J. J. Thomas. Dynamic visualization of transient data streams. In *InfoVis 2003*, pp. 97–104, 2003. doi: 10.1109/INFVIS.2003.1249014 11, 12, 14, 15
- [106] Y. Xiong, B. Dai, and D. Lin. Move forward and tell: A progressive generator of video descriptions. In *ECCV*, vol. 11215, pp. 489–505. Springer, 2018. doi: 10.1007/978-3-030-01252-6_29 13
- [107] P. Yi, Z. Wang, K. Jiang, J. Jiang, and J. Ma. Progressive fusion video super-resolution network via exploiting non-localspatio-temporal correlations. In *ICCV*, pp. 3106–3115. IEEE, 2019. doi: 10.1109/ICCV.2019.00320 13
- [108] E. Zraggen, A. Galakatos, A. Crotty, J. Fekete, and T. Kraska. How progressive visualizations affect exploratory analysis. *IEEE TVCG*, 23(8):1977–1987, 2017. doi: 10.1109/TVCG.2016.2607714 2, 4, 14
- [109] Y. Zhang, G. Li, R. Yue, J. Liu, and G. Shan. Peviz: an in situ progressive visual analytics system for ocean ensemble data. *J. Vis.*, 26(2):423–440, 2023. doi: 10.1007/s12650-022-00883-2 6, 7, 8, 14, 15
- [110] H. Zhao, H. Zhang, Y. Liu, Y. Zhang, and X. L. Zhang. Pattern discovery: A progressive visual analytic design to support categorical data analysis. *J. Vis. Lang. Comput.*, 43:42–49, 2017. doi: 10.1016/j.jvlc.2017.05.004 11, 14, 15



Alex Ulmer received his master’s degree in computer science and visual computing from Technische Universität Darmstadt, Germany. He is currently a PhD candidate at Fraunhofer Institute for Computer Graphics Research and TU Darmstadt. His research interests are focused on visual-interactive analysis of cybersecurity data, progressive visual analytics and explainable AI.



Marco Angelini is an Associate Professor in computer science at Link Campus University Rome, and a researcher at Sapienza University of Rome. He is a member and coordinates research projects of A.W.A.RE group. His main research interests include Visual Analytics, Progressive Visual Analytics, and Human-centered AI. More about him at sites.google.com/dis.uniroma1.it/angelini



Jean-Daniel Fekete (Senior Member, IEEE) received the PhD degree in computer science from the University of Paris-Sud, France, in 1996. He is the scientific leader of the Inria Project Team Aviz that he founded in 2007. His main research areas are visual analytics, information visualization and human-computer interaction.



Jörn Kohlhammer is Head of the Competence Center for Information Visualization and Visual Analytics at Fraunhofer IGD, and Honorary Professor of user-centered visual analytics at TU Darmstadt, Germany. His research interests are focused on decision-centered visual analytics in healthcare and cybersecurity. He is a member of IEEE.



Thorsten May is currently a researcher with the Fraunhofer Institute for Computer Graphics Research (IGD), Germany with a background of mathematics and computer science. His research interests include the systematization of options to combine visualization with machine learning, with a particular focus on multivariate data and progressive visual analytics.