# Comparative Analysis of Timeline-based Visualizations for Dynamic Overlapping Sets

Mariana Pron, Shivam Agarwal, Madhav Poddar, and Fabian Beck †

University of Bamberg, Germany

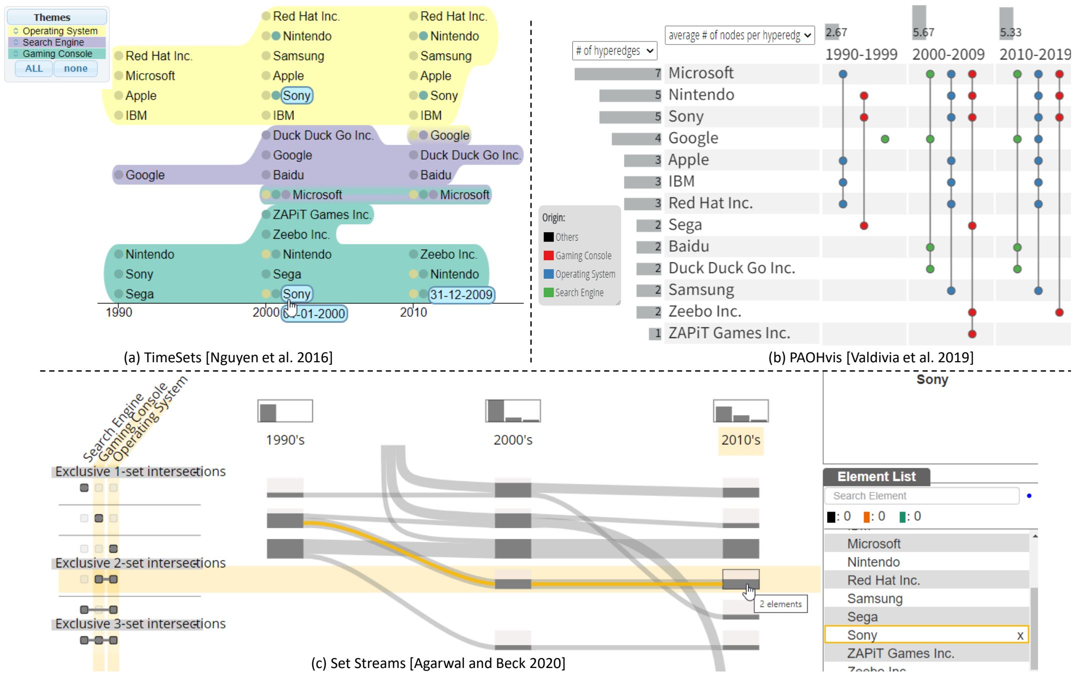


**Figure 1:** *The data of 13 companies offering products in three overlapping categories over three decades is being visualized by: (a) TimeSets, where an element Sony is highlighted on hovering, (b) PAOHvis, which we slightly extended to represent a set with colored dots in a hyperedge (vertical lines connecting rows), and (c) Set Streams, where an element Sony is selected and shown as yellow-colored stream.*

**Abstract**

*Timeline-based set visualizations provide an overview of how overlapping categorical data evolves. We study three different visualization techniques of such type and made minor modifications to visualize the same data in a two-fold comparison. First, we contrast their encodings and interactions through a conceptual analysis. Second, in a user study with 28 participants, we evaluate their performance regarding different analysis tasks for dynamic sets and record user feedback along various dimensions.*

## 1. Introduction

Analysis of a dataset often involves grouping the data items into categories. When the data items belong to multiple categories (or sets), the categories overlap. Various visualizations have been proposed to analyze overlapping sets [AMA*16]. However, most approaches do not include the temporal dimension and hence do not visualize dynamic overlapping sets. But such dynamics are common and found in almost all examples of set-typed data; for instance, companies expanding their portfolio of products, researchers broadening their expertise, and developers contributing to different modules in a repository over time. Being an active research topic, until now,

† email: {shivam.agarwal, madhav.poddar, fabian.beck}@uni-bamberg.de



DOI: 10.2312/evs.20241074

only a limited number of dynamic overlapping set visualizations have been proposed, as evident in a short survey [FFKS21].

To map temporal changes, timeline-based visualizations are used frequently (e.g., [GGJ*21, BBDW17]). Among the limited number of existing dynamic set visualizations, timeline-based designs are most common [NXWW16, VBP*20, AB20]. Timelines provide an intuitive and integrated view of overall dynamics, while accommodating different types of entities (i.e., elements, sets, and their overlaps). To build such approaches, we first need to understand the advantages and limitations of existing visualizations, and assess their comparative performance in analysis tasks. In this work, we lay the groundwork by comparing the three existing timeline-based dynamic overlapping set visualizations. We do not include dynamic set visualizations based on other encodings of time (e.g., animation [MWTI19] or difference-based [ATWB20]), and exclude those that limit the element membership to only one set (e.g., [vLBA*12]).

The contribution of this work is a two-fold comparative analysis of the three timeline-based visualizations [NXWW16, VBP*20, AB20]. Initially, we compare their designs conceptually, focusing on *encodings* and *interactions*. Subsequently, we assess these visualizations through a user study involving 28 participants, aiming to determine their efficacy in performing typical dynamic set analysis tasks. Our findings reveal distinct conceptual and empirical differences among the approaches.

Figure 1 illustrates how three visualizations can depict the same dataset of companies offering products in different categories. We assume that sets are named. *TimeSets* (Figure 1a) displays elements (companies) on a timeline, where the background of each element and adjacent filled colored dots (representing product categories) identify the sets. Elements appearing in multiple timesteps have duplicated labels to show recurring participation. *PAOHvis* (Figure 1b) visualizes sets as hyperedges (edges that can link two or more nodes, here rows) with vertical lines, indicating element relationships over time. To ensure comparability, we opted to make minimal changes and adapted *PAOHvis* by coloring nodes within a hyperedge (e.g., blue-colored nodes indicate *Operating System* hyperedge). *Set Streams* (Figure 1c) employs a multi-view interface to demonstrate exclusive set intersections in rows and timesteps in columns, with streams illustrating changes in set membership over time. Interactive features allow for the selection of specific elements. For transparency, we point out that there is some overlap between the authors of the current paper and *Set Streams* [AB20].

## 2. Conceptual Analysis: Design-based Comparison

We compare the visualizations on two design criteria: visual encodings and interactive data exploration, employing a four-point scale (not, partially, mostly, and fully supported) summarized in Table 1 (an extended version of the table with comments and interaction videos is part of the supplementary material [Aga24]).

**Table 1:** *A design-based comparison of three dynamic overlapping set visualizations. The ratings range from not supported (□ □ □), to supported partly (■ □ □), mostly (■ ■ □), and fully (■ ■ ■).*

| | | *TimeSets* | *PAOHvis* | *Set Streams* |
|---|---|---|---|---|
| **Encodings** | *Time* | ■■■ | ■■□ | ■■□ |
| | *Elements* | ■■■ | ■■■ | ■■□ |
| | *Sets* | ■■■ | ■■■ | ■■□ |
| | *Overlaps* | ■□□ | ■□□ | ■■■ |
| | *Set dynamics* | ■■□ | ■■□ | ■■■ |
| **Interactions** | *Search* | □□□ | ■■■ | ■■■ |
| | *Select* | ■□□ | ■■□ | ■■■ |
| | *Filter* | ■□□ | ■■■ | □□□ |
| | *Reorder* | ■□□ | ■■■ | ■■■ |
| | *Summarize* | ■□□ | ■□□ | ■■□ |

**Encodings.** All visualizations use timelines on the x-axis, but somewhat differently. *TimeSets* presents set memberships as events with label width indicating duration, fully encoding *time*. For comparison, we aggregated individual memberships' timestamps, leading to uniform label widths in Figure 1a. *PAOHvis* and *Set Streams*, lacking duration encoding for memberships and employing discrete timesteps, still mostly support *time* encoding. *TimeSets*, representing *elements* as text labels sorted by set order, can duplicate labels for multiple memberships, as seen with *Sony* in Figure 1a. *PAOHvis* uses rows for *elements*, while *Set Streams*, aggregating elements, requires interaction for detailed exploration, mostly supporting *elements* encoding. For *sets* encoding, *TimeSets* and *PAOHvis* fully support this through colored backgrounds/dots and nodes, respectively. *Set Streams*, displaying only exclusive intersections, mostly supports *sets* encoding, necessitating interactions for exploration. *Overlap* encoding is partially supported by *TimeSets* and *PAOHvis* as they rely on tracking colors and rows. *Set Streams* fully supports *overlap* encoding by representing exclusive intersections as rows, facilitating set identification (e.g., exclusive intersection of *Gaming Console* and *Operating System* in the *2010s*, shown in Figure 1c). Regarding *set dynamics*, the encoding of changes in element's memberships over time, *TimeSets* and *PAOHvis* mostly support this through inference from repeated labels or hyperedges across timesteps. *Set Streams* fully supports *set dynamics* by visualizing changes with streams connecting cells across timesteps, clearly tracking temporal transitions.

**Interactions.** While *TimeSets* lacks interaction for *searching* elements, both *PAOHvis* and *Set Streams* incorporate a search box. The three techniques allow the *selection* of only elements (*TimeSets*), elements and sets (*PAOHvis*), and groups of elements, sets, or overlaps (*Set Streams*), providing partial, most, and full support respectively. The selection results are displayed in a pop-up (*TimeSets*), via highlighted rows and hyperedges (*PAOHvis*), and colored streams (*Set Streams*, as seen with the selected element *Sony* in Figure 1c). *TimeSets* partially supports *filtering* sets (via checkboxes in the legend), while *PAOHvis* fully supports *filtering* by various criteria, such as the minimum number of connecting hyperedges, selected elements, and timesteps. *Set Streams* does not support *filtering*. *Reordering* for more insightful data views is partially supported by *TimeSets* (the vertical order of sets can be changed via the legend), while *PAOHvis* and *Set Streams* provide full support. *PAOHvis* allows sorting by hyperedges, distance measures, and chronology, whereas *Set Streams* sorts by stability, similarity, cardinality, and set priority. *Summarization* is mostly supported by *Set Streams*, which collapses intersection rows for a more concise view, while *Time-*

*Sets* and *PAOHvis* offer partial support through zoom-out features, reducing whitespace or aggregating labels.

This analysis indicates that *TimeSets* and *PAOHvis* are designed to showcase individual elements and sets, whereas *Set Streams* focuses on aggregating elements to display overlaps and temporal changes. *TimeSets* has the least built-in interaction support. *Set Streams*, as a linked-view visualization, offers rich interaction capabilities, and *PAOHvis* stands out for its diverse selection, filtering, and reordering capabilities. Consequently, *TimeSets* may be more suited for scenarios with elements or *events* rarely repeated across timesteps, while *PAOHvis* and *Set Streams* are better equipped for visualizing recurring elements. Moreover, the use of straightforward representations in *TimeSets* and *PAOHvis*, avoiding domain-specific jargon, might facilitate non-expert users.

## 3. User Study: Task-based Comparison

We conducted a user study to answer the research question: *what are the differences among the three visualizations in facilitating the analysis of dynamic overlapping sets?* We collected quantitative data on time and accuracy to measure performance on the analysis tasks, while the qualitative feedback provided insights about the comparison. User study artifacts (questionnaires, responses, and tutorial videos) are included in the supplementary material [Aga24].

**Tasks.** To compare set visualizations in the survey, Alsallakh et al. [AMA*16] introduced static set analysis tasks (element-, set-, and attribute-related). For consistency, we use a similar structure and propose nine dynamic set analysis tasks (Figure 2) and group them in three categories: *element*, *set*, and *time*. Since no dynamic set visualizations involve other attributes in data items, we did not extend tasks related to element attributes in the survey.

**Setup.** Three datasets from varying domains—tech companies (13 elements, 3 sets, 3 timesteps; Figure 1), author-conference contributions (48, 3, 9), and software evolution (111, 5, 10)—were used [AB20]. The study conducted as offline one-on-one sessions in English or German as per participant preference, gathering information on age, education, and familiarity with set theory, information visualization, and the datasets. They were then introduced to the visualizations via short videos and independently explored them. Participants completed nine tasks grouped into three blocks. Each participant explored one dataset and all three visualizations, with the order of visualizations randomized for each block. Response time was logged, concluding with participant feedback on each visualization. Participants were observed by the instructor while performing the tasks. To promote the user study, we advertised and distributed gift cards of 25 Euros to two randomly selected participants.

**Participants.** The study was conducted with 28 participants. They were aged 18–45, most in the 25–29 range, and had no color-vision deficiencies. In terms of education, 14 had Bachelor's degrees, 10 Master's, 2 doctoral, and 2 other qualifications. For data visualization familiarity, 8 were unfamiliar, and 20 had slight to moderate knowledge. In set theory, 14 were unfamiliar, 1 was extremely familiar, and the remaining had slight to moderate familiarity.

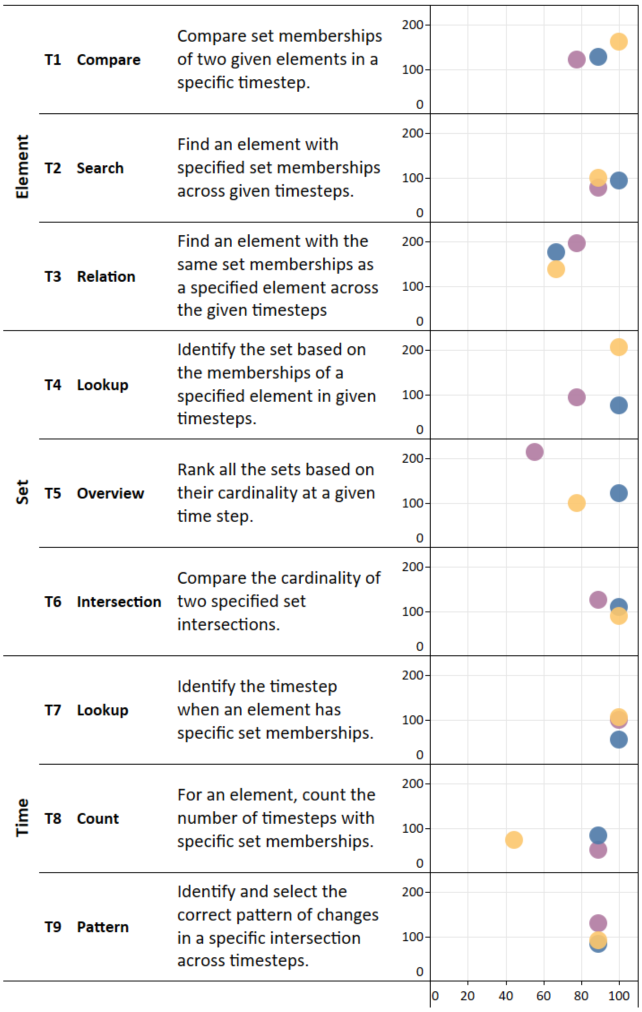


**Figure 2:** *Accuracy (x-axis) vs. completion time in seconds (y-axis) across tasks T1–T9, each discerning the three compared visualization techniques (TimeSets, PAOHvis, and Set Streams).*

**Task Performance.** To assess the performance, we look at the accuracy and completion times of the tasks. Overall, participants had the best accuracy with *PAOHvis* and needed the shortest time, averaged over all nine tasks, as shown in Figure 3. Regarding individual tasks, *PAOHvis* performed well in all tasks (Figure 2). Moreover, time-related tasks (T7, T8, T9) were solved with similar accuracy for all three visualizations, except for task T8 where *TimeSets* had considerably lower accuracy. Participants also struggled with *TimeSets* in task T4, where, although answers were accurate, they took longer. In the case of *Set Streams*, it performed worst for task T5 both in terms of accuracy and completion time. Additionally, participants had lower accuracy with *Set Streams* for task T4, whereas the usage of the other two visualizations for the same task led to accurate results. Regarding the element tasks (T1, T2 T3) and the set-centered task T6, the three visualizations performed similarly.

**Feedback.** Next, we evaluated how participants perceived the ease of using each technique, their overall experience, and interactions in

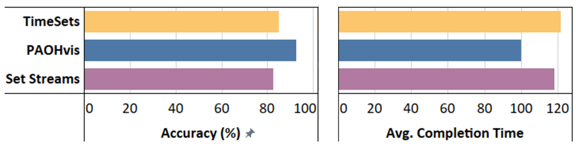


**Figure 3:** *Overall accuracy and completion time of the three compared visualization techniques.*

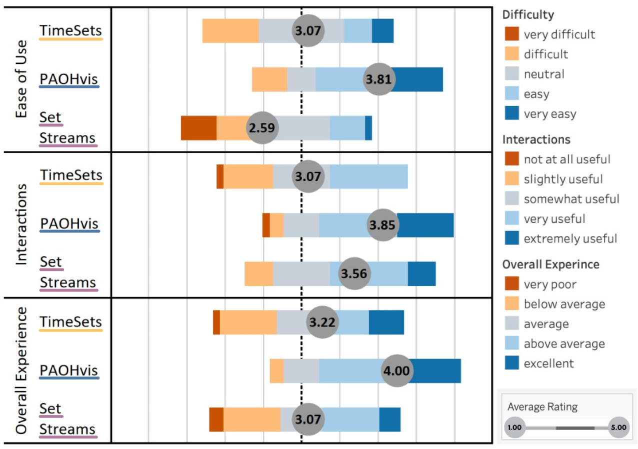


**Figure 4:** *Participant feedback on Likert scales (1–5) regarding different usability aspects, shown as diverging answer distributions with average ratings.*

the visualizations. The participants were asked (each question had five rating scores to choose from): (I) *How difficult was it to understand and use each visualization?* (II) *How useful were the interactions?* (III) *How would you assess your overall experience with each of the visualizations?* Results showed *PAOHvis* receiving the most favorable feedback, while *Set Streams* was found challenging to use, and *TimeSets* had lagging interactivity (Figure 4).

Participants clarified their ratings with additional comments. Regarding difficulty, participants appreciated the individual text labels, use of color for set memberships in *TimeSets*, but were confused when the number of colors increased with more sets. They found *PAOHvis* intuitive with good color usage. Cluttered lines and limited colors in *Set Streams* initially posed challenges, but participants found it useful after some practice. Concerning interaction, *TimeSets* lacked essential features like search, highlighting, and filtering, affecting its usage. Basic interactions in *PAOHvis* (e.g., search, hover) were received well by the participants. However, advanced interactions (e.g., reordering) in both *PAOHvis* and *Set Streams* needed further simplification or explanation.

**Limitations.** We had to make minimal extensions in *PAOHvis* to ensure comparability and use common datasets. While this is a viable approach, the unique strengths of individual techniques (e.g., encoding individual membership events with different time intervals in *TimeSets*) or advanced features (e.g., comparison of two selected element groups in *Set Streams*) could not be included. Most participants had low to medium data analysis expertise, which may not align with the targeted visualization expert audience by *Set Streams*. However, the study still adds value by highlighting the specific design aspects to be simplified for a non-expert audience. We used real-world datasets, but did not systematically vary their complexity (in terms of element and set counts, overlap sizes, or timesteps). Finally, our statistical analysis is restricted to descriptive statistics; however, we made sure to only interpret clear and practically relevant differences in the quantitative data.

## 4. Discussion and Conclusion

We have conducted two complementary analyses of three dynamic set visualization techniques and gained relevant insights.

**Self-explanatory Design.** Text labels for individual elements in *TimeSets* and *PAOHvis* were well-received, while aggregated representation in *Set Streams* required more intuitive interaction methods. While designing generalizable visualizations is desirable, it may not always be possible, demanding alternate solutions. Hence, domain-specific adaptations like using clear, contextual text for explanations and dynamic captions for complex interactions [MLBW20] could enhance understandability. For instance, we could use *Companies offering only 1 product* instead of *Exclusive 1-set intersections* for tech companies dataset in *Set Streams*. This could help improve the visualization performance on some analysis tasks, e.g., regarding set lookup and overview (T4, T5).

**Interactions.** Although interaction support by *TimeSets* was lower in comparison, it could be improved easily, e.g., by integrating *searching* an element that could improve the performance in some tasks (T8). However, integrating other interactions might be more complex due to the impact on the visualization design and layout. For instance, a *summarize* interaction requires visual abstractions, which impacts the encoding of individual elements and sets in *TimeSets* and *PAOHvis*. Hence, when developing new timeline-based approaches, integrating such interactions with the visual encodings and layout should be considered early in the design process.

**Visualization Scenario.** Our results indicate that *TimeSets* is optimal for analyzing non-recurring elements with text labels, *PAOHvis* excels in showing relationships over time through dedicated rows, and *Set Streams* focuses on set dynamics with stream encoding. While set overlap analysis is common, other set relations (e.g., union, difference) could also be relevant. This suggests a need for visualizations that can accommodate other analysis scenarios, for instance, focusing on other element attributes (e.g., location) while showing temporal changes in overlapping sets on a timeline. Additionally, the compared visualizations do not scale well. For instance, representing more than six sets is challenging due to the requirement of unique colors (*TimeSets* and *PAOHvis*) or accommodating the high number of exclusive intersections mapped as rows (*Set Streams*). For better scalability we need novel designs [FMW19]. Currently, we lack such solutions.


### Acknowledgments

We thank the authors of TimeSets and PAOHVis techniques for sharing the code and for making it available for extension. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 517412225.

## References


[AB20] AGARWAL S., BECK F.: Set Streams: Visual exploration of dynamic overlapping sets. *Computer Graphics Forum 39*, 3 (2020), 383–391. doi:10.1111/cgf.13988. 2, 3

[Aga24] AGARWAL S.: Supplementary material: Comparative analysis of timeline-based overlapping dynamic set visualizations, 2024. doi:10.17605/OSF.IO/2GHBM. 2, 3

[AMA*16] ALSALLAKH B., MICALLEF L., AIGNER W., HAUSER H., MIKSCH S., RODGERS P.: The state-of-the-art of set visualization. *Computer Graphics Forum 35*, 1 (2016), 234–260. doi:10.1111/cgf.12722. 1, 3

[ATWB20] AGARWAL S., TKACHEV G., WERMELINGER M., BECK F.: Visualizing sets and changes in membership using layered set intersection graphs. In *Vision, Modeling, and Visualization* (2020). doi:10.2312/vmv.20201189. 2

[BBDW17] BECK F., BURCH M., DIEHL S., WEISKOPF D.: A taxonomy and survey of dynamic graph visualization. *Computer Graphics Forum 36*, 1 (2017), 133–159. doi:10.1111/cgf.12791. 2

[FFKS21] FISCHER M. T., FRINGS A., KEIM D. A., SEEBACHER D.: Towards a survey on static and dynamic hypergraph visualizations. In *IEEE Visualization Conference (VIS)* (2021), pp. 81–85. doi:10.1109/VIS49827.2021.9623305. 2

[FMW19] FABRIKANT S. I., MIKSCH S., WOLFF A.: Visual analytics for sets over time and space (Dagstuhl Seminar 19192). *Dagstuhl Reports 9*, 5 (2019), 31–56. doi:10.4230/DagRep.9.5.31. 4

[GGJ*21] GUO Y., GUO S., JIN Z., KAUL S., GOTZ D., CAO N.: Survey on visual analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics* (2021), 1–1. doi:10.1109/TVCG.2021.3100413. 2

[MLBW20] MUMTAZ H., LATIF S., BECK F., WEISKOPF D.: Exploranative code quality documents. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 1129–1139. doi:10.1109/TVCG.2019.2934669. 4

[MWTI19] MIZUNO K., WU H.-Y., TAKAHASHI S., IGARASHI T.: Optimizing stepwise animation in dynamic set diagrams. *Computer Graphics Forum 38*, 3 (2019), 13–24. doi:10.1111/cgf.13668. 2

[NXWW16] NGUYEN P. H., XU K., WALKER R., WONG B. W.: TimeSets: Timeline visualization with set relations. *Information Visualization 15*, 3 (2016), 253–269. doi:10.1177/1473871615605347. 2

[VBP*20] VALDIVIA P., BUONO P., PLAISANT C., DUFOURNAUD N., FEKETE J.-D.: Analyzing dynamic hypergraphs with Parallel Aggregated Ordered Hypergraph Visualization. *IEEE Transactions on Visualization and Computer Graphics 27*, 1 (2020), 1–13. doi:10.1109/TVCG.2019.2933196. 2

[vLBA*12] VON LANDESBERGER T., BREMM S., ANDRIENKO N., ANDRIENKO G., TEKUSOVA M.: Visual analytics methods for categoric spatio-temporal data. In *Proceedings of the 2012 IEEE Conference on Visual Analytics Science and Technology* (2012), pp. 183–192. doi:10.1109/VAST.2012.6400553. 2



Proceedings published by Eurographics - The European Association for Computer Graphics.