Van Goethem, A., Staals, F., Loffler, M., Dykes, J. & Speckmann, B. (2017). Multi-Granular Trend Detection for Time-Series Analysis. IEEE Transactions on Visualization and Computer Graphics, 23(1), pp. 661-670. doi: 10.1109/TVCG.2016.2598619

Multi-Granular Trend Detection for Time-Series Analysis

Arthur van Goethem, Frank Staals, Maarten Löffler, Jason Dykes, and Bettina Speckmann

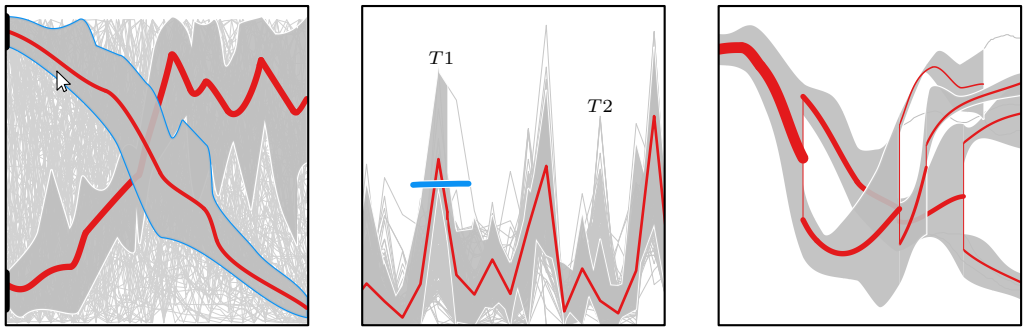

Fig. 1. Left: automated trend detection in combination with manual interaction finds two opposing key trends in the data (CO2 intensity of economic output: kg CO2 per 2005 PPP $ of GDP [13]). Middle: selecting only the subset that is part of the peak at $T1$, directly shows that this subset is not part of the peak at $T2$ (homogenized daily amount of precipitation for 240 weather stations [21]). Right: different visual styles help (de-)emphasize different properties of the detected trends.

Abstract— Time series (such as stock prices) and ensembles (such as model runs for weather forecasts) are two important types of one-dimensional time-varying data. Such data is readily available in large quantities but visual analysis of the raw data quickly becomes infeasible, even for moderately sized data sets. Trend detection is an effective way to simplify time-varying data and to summarize salient information for visual display and interactive analysis. We propose a geometric model for trend-detection in one-dimensional time-varying data, inspired by topological grouping structures for moving objects in two- or higher-dimensional space. Our model gives provable guarantees on the trends detected and uses three natural parameters: granularity, support-size, and duration. These parameters can be changed on-demand. Our system also supports a variety of selection brushes and a time-sweep to facilitate refined searches and interactive visualization of (sub-)trends. We explore different visual styles and interactions through which trends, their persistence, and evolution can be explored.

Index Terms—Interactive Exploration, Trend Detection, Time Series.

1 Introduction

Due to our increased ability to monitor processes and record detailed information, a large body of data is readily available that describes the behavior of complex processes over time. Sensors in our mobile phones and cars, or more conventional sources such as weather stations, supply continuous information on the state of different variables over time. In this paper we focus on one-dimensional time-varying data, that is, *time-series data*. An important type of time-series data are so-called *ensembles*: the set of simulation results of intricate models for varying starting conditions and parameter settings. Ensembles give an indication of the possible spread of a model over time and contain exact traces of the different inputs, making them highly valuable for domain experts [26]. They are commonly used in weather prediction [8, 16], climate change [20, 38] and hurricane prediction [12, 35].

The visual analysis of time-series data quickly becomes infeasible if the raw data is used directly. The high complexity of the data sets, combined with their inherent uncertainty, makes them difficult to analyze. Hence there is a growing body of visualization literature [3, 26] that attempts to visualize uncertainty in data.

- *A. van Goethem and B. Speckmann are with TU Eindhoven. E-mail: a.i.v.goethem@tue.nl and b.speckmann@tue.nl.*
- *F. Staals is with MADALGO, Aarhus University. E-mail: f.staals@cs.au.dk.*
- *M. Löffler is with Utrecht University. E-mail: m.loffler@uu.nl.*
- *J. Dykes is with City University London. E-mail: J.Dykes@city.ac.uk.*

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014; date of publication xx xxx 2014; date of current version xx xxx 2014. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

One common approach to visualize uncertainty, while also decreasing complexity, is to reduce the data to its main statistics. Heat-maps and fan-charts, for example, give a statistical overview of the distribution of the data. The abstraction of these maps is their main strength, but also their main weakness. The strong reduction of the data creates structure but at the same time runs the risk of overemphasizing the extremes and "hot spots". Without knowledge of the individual entities that comprise the data such representations may be misleading. Other options to display uncertainty include the display of regular distribution plots [31], and uncertainty isolines [30].

An alternative approach to displaying uncertain data is to enforce more structure on the data. Agglomeration via *trend detection* reduces complexity while maintaining a significant part of the information contained in the original input data. Trend-detection groups entities that behave similarly over an extended period of time into *trends*. The extent to which time series "behave similarly" and the notion of an "extended period of time" are not predetermined. However, results and interpretation are dependent upon these analytical parameters. We think of a trend as a set of entities such that all entities in the trend are a small distance away from their neighbors. See Fig. 2 for an illustration. This criteria of closeness results in trends that reasonably match with expectations. There may be more aspects that affect human perception of trends, though, such as the distribution of elements.

A more abstracted view of the data reduces the cognitive load for the user and may help detect structure that is hidden when all information is shown. In this sense trend-detection serves a similar goal as edge-bundling [18, 19, 28]. Trend detection, and the subsequent visualization of the detected trends, also performs well for multi-modal time series or ensembles.

Trend detection is not without challenges. The visualization of

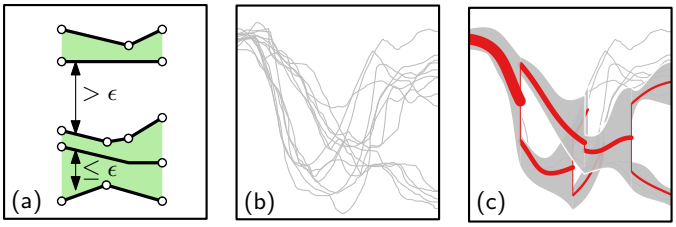

Fig. 2. (a) Entities that form chains of close neighbors define a trend. (b) A simple set of time series. (c) The trends found by our algorithm. A trend is a set of entities that are in close proximity of each other for an "extended period of time". Note that there are relations between the trends, that is, the initial trend splits into two subtrends, both of which again have subtrends.

trends has an inherent effect on the perceived support, variability, and distribution of different trends. It is unclear how this affects user perception. Furthermore, agglomeration requires a predefined granularity to detect trends. Since the trends, and hence the necessary granularity, are not known a piori, trend detection should be performed at different granularities. However, many current methods for trend detection do not use models that easily support a change of granularity.

Contribution. We propose a geometric model for trend-detection in one-dimensional time-varying data, inspired by topological grouping structures for moving objects in two- or higher-dimensional space. Specifically, our model builds on the work by Buchin et al. [5] and Van Kreveld et al. [40]. Trends are detected using three natural parameters: granularity, support-size, and duration. The model is firmly rooted in geometrical methods for trend detection and trends have a clean mathematical definition. This mathematical soundness allows for provable guarantees on the detection of trends.

We define a series of tasks and visualization criteria for trend detection systems. Based on these tasks we then explore different visual styles and show how they may affect the task-appropriateness of the system. Distinct styles affect the displayed information differently and we discuss their trade-offs and complications.

Finally, we discuss several tools that help to interactively explore the data via a computer-assisted model. Different selection brushes, fixed starting conditions, and a time-sweep allow the user to decompose and construct trends on-demand.

Related Work. The visualization of uncertainty is an important current area of visualization research [26]. In uncertainty visualization [3, 32] a general stochastic uncertainty in the data is assumed. This uncertainty may be based in inexactness of the measurements or of the model. To represent this uncertainty the probability density function (PDF) is visualized. Overview statistics (such as the mean and standard deviation) are used to display the variability in the PDF over time or position. Different glyphs have been proposed to visualize this variability, for example, box plots, violin plots [17], and summary plots [31]. Higher-dimensional data may be visualized using (probabilistic) isolines to display uncertainty [2, 29, 30, 41]. Local surface distortion [15] and animation [4] have also been applied to display uncertainty in higher-dimensional data.

Ensemble visualization differs from uncertainty visualization, as the different data series are not stochastic samples, but correspond to exact instances of the input. The ability to analyze and correlate data series, and thus different inputs, make ensembles very valuable for domain experts [26]. Previous work on ensemble visualization has mainly focussed on representing two-dimensional and three-dimensional ensemble data. Sanyal et al. [37] proposed a spaghetti-like representation of the ensemble, highlighting different features in the data set. Mirzargar, Whitaker, and Kirby [24] recently proposed a curved box plot to display ensemble information. They use the model of data depth to determine a multi-dimensional distribution function. However, similar to PDF-based visualizations they are able to handle only uni-modal density functions. Ferstl, Burger, and Westermann [11] use trend detection to agglomerate the different ensemble members. They use principle component analysis (PCA) to detect trends in the ensemble. PCA, however, is not intuitive and changing the dimension for PCA may not give clear changes in trend detection. Dykes and Brunsdon [10] use ensembles to investigate the distribution of a single data variable across space and scale. Potter et al. [33] argue that presenting all information in a single display is not required. They propose a multiple linked displays framework to study ensembles. Finally, multi-dimensional ensembles may also be projected down to one-dimensional functions. Demir, Dick, and Westermann [9] combine histograms with local one-dimensional ensembles and brushing techniques to interactively explore a three-dimensional ensemble.

Time-series data, similar to ensembles, consist of a set of time-varying series. In contrast to ensembles though, the series need not stem from a single model, but may be uncorrelated. Time series are often used as a part of larger visualization frameworks, for example, in small multiples. The comparison of different time series and trend detection within groups of them has received some attention. Stacked graphs [6] can be useful to visualize several functions at the same time, but comparisons between different series may be difficult. Recently Horizon Graphs [27, 34] have have been introduced as a tool to display multiple time series while allowing easy and quick comparisons.

Our work on trend detection is based on the trajectory grouping structure, introduced by Buchin et al. [5]. Intuitively, a group is a sufficiently large set of entities (for example, animals, people, or cars) that move together during a sufficiently long time interval. Buchin et al. present an efficient algorithm that, given the trajectory data, can compute all maximal groups, and the relations between these groups (that is, groups G_1 and G_2 merge into group G at time s, and G splits into groups H_1 and H_2 at time t). Recently, van Goethem et al. [39] extended their work to efficiently change the group parameters, that is, the required minimum group size, the minimum group duration, and the maximum distance between the group members. Their work allows a user to interact with the data set to explore the effect of the different parameters. Kostitsyna et al. [22] and Löffler et al. [23] extend the work of Buchin et al. to environments with obstacles. Van Kreveld et al. [40] recently observed that in dense environments the original definition by Buchin et al. may not capture the nature of a group well. They propose a slight variation on the definition for this setting. This adapted definition, however, does lead to a higher computational cost.

Organization. We formally define the problem we study in Section 2. There we discuss the concepts involved in trend detection and identify primitive tasks. Section 3 presents the algorithmic foundations of our work. We first describe the work by Buchin et al. and then explain a simpler and faster algorithm for trend detection. Section 4 discusses a variety of visual encodings that may be used to represent trends and explains how these visual encodings relate to the primitive tasks. In Section 5 we explain our tools for interactive exploration. Using the results of all previous sections, in Section 6 we show how to detect correlations in two real-world examples with our system. Section 7 discusses the limitations of our approach and Section 8 concludes with a general discussion and possible directions for future work.

2 Problem Definition

In this section we first define the concepts of time series, ensembles, and trends. We then analyse the problem of trend detection and define the tasks that should be supported by any trend-detection system. We also define several aesthetic criteria that should be taken into account to efficiently visual the detected trends.

2.1 Preliminaries

Time series. A *time series* is a time-ordered sequence of data points that describes how a variable changes over time. Each data point d can be described by a pair $d = (t, v)$, where t is the time-stamp at which the data has value v. We consider only time series that have exactly one data-variable, hence v is assumed to be a single real number. The *length* of a time series is the number of data points. We assume that the data changes linearly in between consecutive data points. Thus, we can interpret a time series as a piece-wise linear function describing the

change in a variable over time. We do not require that the data points are uniformly spaced in time.

A collection of time series is called *time-series data* and consists of n different time series. Each of these time series has at most τ data-points. For ease of description we assume that all time series have exactly the same time span $[t_s, t_e]$.

Ensemble. An *ensemble* is a specific form of time-series data where all time series originate from (a perturbed version of) the same model. All time series describe the change of the same variable over time but have slightly perturbed starting conditions or models. Consequently, these time series are strongly correlated at the initial time-stamp and as time increases become more diverse. Very well known examples include weather predictions and runs of climate change models.

Trends. In trend-detection we are interested in detecting the hidden structure in an input data set. That is, we want to detect whether there are several time series that behave similarly over an extended period of time. We say that such a set of similar time series form a *trend*, and that the time series comprising it form the *trend-members*. When time series "behave similarly" and what an "extended period of time" is, depends on the model used. We use the grouping structure as the base for trend detection. Intuitively this means that all entities comprising a trend should be at most a small distance ε away from their neighbors in the trend. A more formal definition is given below. See Fig. 2(b).

Important properties of a trend are its support, duration, range, and its distribution. The *support* of a trend is the number of time series that are part of the trend, the *duration* is the length of the time interval during which the time series behave similarly. The *range* of a trend is the interval between the highest valued and lowest valued time series in the trend, and the *distribution* of a trend indicates where exactly the time series making up the trend lie within the range of the trend. For a given trend T, a subset of the time series in T may form a trend T' that has a longer duration (but a lower support) than T. We refer to such a trend as a *sub-trend* of T. See Fig. 2(c) for an illustration.

We focus on finding and displaying only those trends that are present initially, that is, at time t_s. There are several advantages to focussing only on initial trends. First, the detection of trends starting at a fixed point in time naturally captures the uncertainty in the ensembles that we consider (for example, weather predictions). As a result, the trends are naturally diverging. Note that by changing the starting position users can still browse all parts of all different trends. Second, the grouping algorithm runs significantly faster than in the general case, allowing for better user interaction.

By considering only trends that start at t_s, the trends form a forest [7]. We may have several trends with large support and a short duration that form the roots of the trees in our forest. Each of these trends may have several children that represent sub-trends of slightly lower support and larger duration. In turn, these trends may have sub-trends of their own, thus resulting in a tree structure.

2.2 Problem Analysis

An input instance consists of a set of n time series. Each time series is a piece-wise linear function defined over the time-interval $[t_s, t_e]$. The objective is to detect patterns, at arbitrary granularities, to find correlations between different input time series. We assume a computer-assisted model in which the user interacts with automated trend detection to analyze the data. Within the model, we specify a series of primitive tasks that should be supported by the system to enable exploration and correlation of the data. We also specify different visual encodings that affect the usability of the system.

Aigner et al. [1] describe a set of tasks that are involved in the analysis of time-oriented data. These tasks include classification, clustering, search and retrieval, pattern discovery, and prediction. Our approach is focussed on the clustering task and as such all primitive tasks described below are focussed around the effectiveness of the clustering. In a clustering task the data is grouped based on similarity, in our case inter-entity distance. Clustering abstracts the data and therefore it is important that it does not falsely represent the underlying data. Manual pattern discovery (automatic detection of interesting patterns), and prediction (infer the evolution of data in the future) may be supported through the use of our clustering approach, however we do not support the automated computational support of these tasks.

Primitive Tasks. To detect coherence in the data several primitive tasks need to be performed. These tasks are centered on two core concepts: estimating trend statistics and determining connectivity within the detected trends. The tasks support a more abstract level of data processing where similarly behaving items are clustered in trends.

- **T1a** Estimate support of a trend.
- **T1b** Estimate range of a trend.
- **T1c** Estimate distribution inside a trend.
- **T2a** Determine behavior of (a subset of) time series over time.
- **T2b** Determine how granularity affects the trend formation.
- **T2c** Determine connectivity of the trend structure.
- **T2d** Correlate different trends.

The combination of different primitive tasks allows more complex questions to be answered. What is the strongest supported trend in the data and how large is the range? (T1a, T1b). How does the range of a group of time series change over time? (T1b, T2c) . Are there time series that are included in two or more specific trends? (T2a, T2b, T2d). Do the time series included in a given trend behave consistently, or do they form finer trends at other granularities? (T2a, T2b). How do the initial conditions affect future behavior? (T2c, T2d).

An effective visualization should support the majority (if not all) of the primitive tasks. Not all tasks are enabled equally by different visualizations though. We focus more on the effects of visual encoding on the primitive tasks in Section 4.

Visual Encoding. Besides the support of primitive tasks, an effective visualization should also aim to reduce the effort involved. Clear visual cues help reduce the cognitive load by pre-structuring the presented information. Reducing the cognitive load improves task-efficiency and maintains clarity for complex data sets. We distinguish several aspects that may positively affect the task-effectiveness.

- **V1** Firm grounding in input data.
- **V2** Clear trend separation.
- **V3** Low visual complexity of trends.
- **V4** Continuity of shapes in trend-structure.
- **V5** Visual distinction between uni-modality and multi-modality.

To properly judge relationships within the data, the visualization should display only information that is firmly grounded in the input data (V1). This (nearly) trivial observation can be difficult to ensure in practice and has often been overlooked. The display of range (T1b), affects the perceived distribution (T1c) of a trend, unless explicitly visualized. Support (T1a) and range (T1b) together define the local density at each point in time, but either may overshadow the other, affecting the perceived density. Furthermore, a visualization should not introduce salient features that are not present in the input data. This is problematic for automated trend detection since the (arguably artificial) begin- and end-point of a trend are highly visible.

As trends may overlap, clear trend separation (V2) ensures that different trends can easily be kept apart. This allows different trends to be related and emphasizes the structure between trends (T2a, T2c, T2d). Reducing the visual complexity of the represented trends (V3) helps reinforce this structure even further and shifts focus from exact representations to a higher-level structure. We note that this is in stark contrast with the first criterion (V1) as simplifying the data reduces the strength of the relation to the original input data. The low visual complexity of trends is improved further by the (smooth) continuation of trends (V4). We ensure that trends split into sub-trends continuously and so further emphasize the structure in the trends (T2c, T2d).

When the distribution inside a trend is not explicitly encoded, visualizations may incorrectly imply distributions with are not in fact

present in the data. For example, by solely encoding the range of a trend, a uniform or uni-modal distribution may be implied. In many cases this is, however, not a fair implication. By making a visual distinction between uni-modal or multi-modal distributions (V5) we can help parse the underlying structure of trends (T2a, T2b).

3 Computing Trends

In this section we describe how the work on trajectory grouping by Buchin et al. [5] and Van Kreveld et al. [40] can be used to detect trends. We first revisit their definitions, and see how they relate to trends in time-series data in Section 3.1. The algorithms for computing trajectory grouping structures may be used to detect maximal groups (and as a result, trends) that begin and end at any arbitrary position in time. However, in our setting, we are interested only in maximal groups that start at the initial time t_s. In Section 3.2 we present a simple algorithm, based on the one by van Kreveld et al., to compute all maximal groups that start at t_s.

3.1 Trajectory Grouping Structure

Buchin et al. [5] consider the problem of detecting maximal groups in trajectory data. They define a (m,ε,δ)-*group* to be a set of at least m entities (e.g. animals, people, or objects) that travel together, with respect to distance parameter ε, during a time interval of length at least δ. A set of entities G is together if and only if it is ε-*connected*: that is, if for every pair of entities $a,b \in G$, there is a chain of entities $a = c_1,..,c_k = b$ such that for any pair of consecutive entities c_i and c_{i+1} the distance is at most ε. See Fig. 3 and Fig. 5(a).

Consider each time series as the trajectory of an entity that moves in $\mathbb{R}^1$. We then observe that an (m,ε,δ)-group G corresponds to a set of (at least m) time series that behave similarly, that is, they are ε-connected at any time, during a time interval of length δ. Indeed, the notion of a group naturally corresponds to a trend.

Let G be an (m,ε,δ)-group on time interval I of size $|G| > m$. All $\binom{|G|}{m}$ subsets of size m also form (m,ε,δ)-groups on interval I. We are interested only in *maximal* groups G: groups that are maximal in size and duration (that is, we cannot add any entities to G, nor can we grow the interval on which G is ε-connected). In the remainder of this paper, we refer to mean maximal groups simply as groups. Buchin et al. [5] show that, in a collection of n trajectories of length τ, there are at most $O(\tau n^3)$ maximal groups, and they can be computed in $O(\tau n^3)$ time.

In the definition of groups as given above, the entities in a (maximal) group G may be ε-connected through entities that are not in G. If the set of entities is very dense this may yield somewhat unintuitive results, see Fig. 4. Therefore, van Kreveld et al. [40] give a refined definition of group in which the entities in group G have to be ε-connected using *only* entities that are also in G. The number of maximal groups according to this definition is still $O(\tau n^3)$. Computing all of them takes $O(\tau^2 n^4)$ time. Since our ensembles are likely to be dense, we consider this refined version of groups.

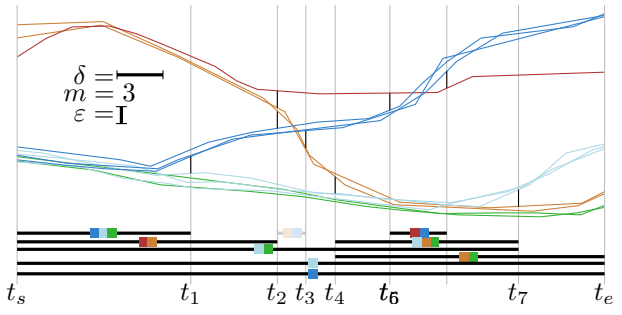

Fig. 3. A set of trajectories and maximal groups as defined by Buchin et al. [5]. The two sets of blue entities also form a group on $[t_s,t_1]$, however this group is not maximal, as all blue entities together with the green entities also form a group on $[t_s,t_1]$. The orange and blue entities do not form a group on $[t_2,t_3]$ as their time interval is too short.

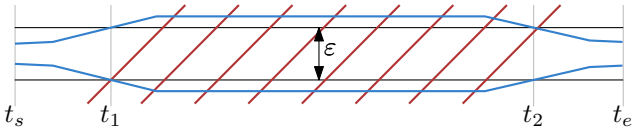

Fig. 4. An (artificial) example showing the difference between the group definitions of Buchin et al. [5] and van Kreveld et al. [40]. The blue entities by themselves are too far apart during $[t_1,t_2]$, but they remain ε-connected through the red entities. Hence, by the former definition they are a maximal group on $[t_s,t_e]$, whereas by the latter definition they are a group only on $[t_s,t_1]$ and on $[t_2,t_e]$.

3.2 Algorithm for Trend Detection

The algorithm by van Kreveld et al. [40] can find all groups in a set of trajectories (or in an ensemble) in $O(\tau^2 n^4)$ time. We are interested in a more restricted setting where all groups start at the initial time t_s. As a result, groups can split, but will never merge. The algorithm by van Kreveld et al. can be adapted to our setting to run in $O(\tau n^3 \log n)$ time. However, in this restricted setting a much simpler solution is possible. Next, we describe a simple sweep line algorithm that achieves the same running time, and uses only elementary data structures. In particular, there is no need for a data structure storing the arrangement (ensemble) in which we can delete lines (time series). Consequently, the new algorithm is significantly simpler to implement than the technically rather involved algorithm by van Kreveld et al. An overview of the algorithm is given in Algorithm 1.

We start by sorting all time series by increasing order of their values at time t_s. To detect the maximal groups that are present at t_s, we perform a linear scan over the time series. We maintain the maximal groups such that for each consecutive pair in a group they are at most ε apart at t_s. As the groups are maximal at the start we can do this greedily (see Fig. 5(a)). Hence, in $O(n \log n)$ time we find all ε-connected maximal groups at t_s. As groups cannot merge, we process each of the initial ε-connected groups separately.

To detect when a maximal group G stops being ε-connected, we use a vertical sweepline and sweep from t_s to t_e. There are three types of events we may encounter: (i) *vertex events*, at which we encounter a new data-point of a time series (Fig. 5(b)), (ii) *intersection events*, at which two time series intersect (Fig. 5(c)), and (iii) *split events*, at which the distance between two vertically consecutive, and diverging time series becomes exactly ε (Fig. 5(d)). We process all events based on increasing time using a priority queue.

At a vertex event belonging to a time series T, the computed split and intersection events of T with neighboring time series become invalid as T follows a new linear segment. We update the intersection and split events with both neighbors in the priority queue.

At an intersection event the order of two consecutive time series is

Algorithm 1 `ComputeMaximalGroups`(T)

Input: a non-empty set T of n time series in the interval $[t_s,t_e]$.
Output: the set $\mathcal{G}$ of maximal groups starting at t_s.

{*Find initial maximal ε-connected sets:*}
Sort T by increasing their value at t_s.
Scan through T to find all maximal ε-connected sets at time t_s.
for all initial maximal ε-connected sets G **do**
  Let E be the ensemble induced by all the time series in G.
  Sweep E to find the first ε-split at time t', at which G splits into G_1 and G_2. Let $t' = t_e$ if no such split exists.
  {G *is a maximal group on time-interval* $[t_s,t']$.}
  Add G, with time interval $[t_s,t']$, to $\mathcal{G}$.
  if $t' \neq t_e$ **then**
    $\mathcal{G}_1 =$ `ComputeMaximalGroups`(G_1)
    $\mathcal{G}_2 =$ `ComputeMaximalGroups`(G_2)
    Add sub-groups from $\mathcal{G}_1$ and $\mathcal{G}_2$ lasting longer than t' to $\mathcal{G}$
return $\mathcal{G}$

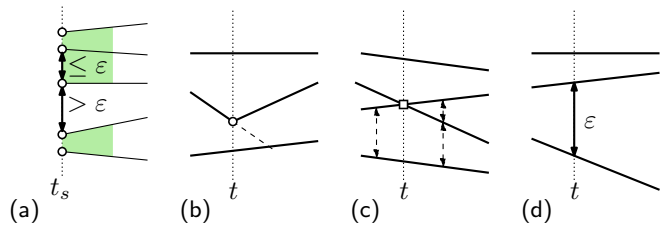

Fig. 5. (a) Finding maximal ε-connected groups at t_s (in green) can be done greedily. Groups are drawn wider for illustrational purposes only. (b) Vertex events change the relation with neighboring entities. Future intersection events and split events must be recomputed. (c) Intersect events change neighbor relations. (d) Split events occur only when neighboring entities are not ε-connected anymore.

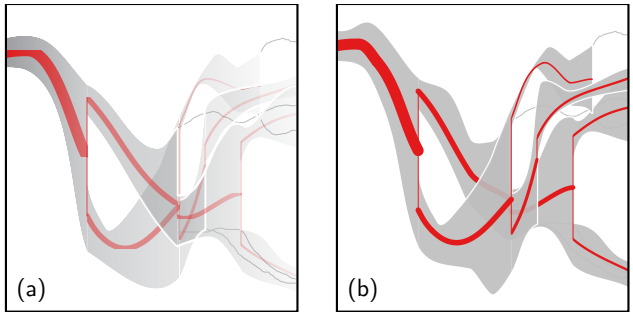

Fig. 6. (a) Linear relationship between opacity and the support of a trend. (b) Smoothed extreme values for each trend.

inverted, changing which pairs of time series are consecutive. This again affects the future intersect- and split-events of the time series involved. We remove the old intersect- and split-events from the priority queue, and add the new events to the queue.

At a split event the current maximal group G stops forming an ε-connected group. We report G, and split the group in two ε-connected groups G_1 and G_2. As subgroups may stop earlier than the group they split from, we restart the computation at t_s for both subgroups.

Analysis. Each of the n time series has at most τ data points. It follows that there are at most $O(\tau n)$ vertex events, and $O(\tau n^2)$ intersection events. At each split-event at least one new maximal group is found. There are at most n maximal groups that end at t_e, each containing a single element. As the groups can only split into sub-groups, the number of split events is at most $n-1$. Handling a vertex event or an intersection event takes $O(\log n)$ time. Thus, in between two split-events we spend at most $O(\tau n^2 \log n)$ time. Since there are at most $O(n)$ split events the total running time is $O(\tau n^3 \log n)$.

If we consider yet another variation on the definition of group, in which we require a (sub)group G to be ε-connected only from the time t it comes into existence (that is, because a larger group split into G and G' at time t), then we do not need to restart from t_s for each split. In this case the algorithm runs in $O(\tau n^2 \log n)$ time.

4 Visual Styles

Automated trend detection allows a flexible high-level overview of time series and ensembles. The way in which trends are encoded should involve visual design that reveals and emphasizes the structure in the data. Distinct visualizations present the information differently. In this section we explore the different visual styles possible and discuss trade-offs. We focus on several types of information we want to maintain in our visualization. The behavior of trends over time is captured in the structure and connectivity of trends and their splits (T2c, T2d). The main statistics of the trends give an overview of the local coherence. We focus on support, range, and distribution (T1a, T1b, T1c). We purposefully do not visualize quartiles since this would imply that the underlying data is uni-modal (T1c, V5). To ensure trends are clearly separated we add a white casing to all trends (V2). See, for example, Fig. 2(c), in which the white casing around the top child of the root allows us to more easily distinguish it from the bottom child.

4.1 Main Statistics

Support. To encode the support (T1a) of the different trends and sub-trends in our data, we add a centerline to the trends. The centerline is positioned along the function describing the mean value of all trend-members to prevent outliers from affecting it overly much. The width of the centerline is used as a natural cue to encode the support of the trend. Besides encoding support, the centerline serves two more purposes. First, it is a concrete and localized item that the user can relate with the general trend that is occurring (T2a). Second, it re-enforces the top-level structure of trends (T2c, T2d).

The encoding of trend structure and support using a centerline is a plausible approach. We do, however, note several issues that might occur under boundary conditions. Strongly supported trends that are highly correlated and have a small range may be overshadowed by the centerline (¬V1). This can partially be resolved by globally scaling the width of the centerlines. In the exact opposite case, where the centerline is minimal compared to the range of the group, the centerline need not be the most visually salient feature. This might lead to an overestimation of the support of a group as the range is visually dominant. Finally, using the mean position for the centerline implicitly encodes a uni-model distribution (¬V5). This is not always justified, specifically when trends split. We discuss possible solutions in Section 4.2.

Besides encoding support on the centerline, we may also encode it through the use of transparency (see Fig. 6(a)). Here the transparency of a trend is directly correlated with the support of a trend. To ensure a consistent smooth change we linearly interpolate the gradient in each trend to match at trend splits. The main disadvantage of transparency is that future trends may become less discernable, especially when many trends overlap (¬T2a, ¬T2c). Furthermore, the strength of a trend is not solely defined by the support, but also by the range. A trend with similar support, but smaller range is more firmly present in the data. However, a trend with a smaller range (and similar support) will be less visible in the visualization (¬V1).

Distribution. The distribution within a trend supports analysis of trends at other granularities than the current (T1c, T2a, T2b). The extremes, quartiles, and mean are an indication of trends at different granularities. Re-introducing the statistical information for each trend increases information density, however, that is at odds with our goal to simplify the trend structure (V3, V4). We mainly focus on displaying the range of possibilities using the extremes of the trend. A similar approach might be applied using the quartiles as boundaries. We purposefully do not include both as this visually reinforces a uni-modal distribution, whereas a significant part of our data is likely not uni-modal. We investigate several options to visualize the range and compare their weaknesses and strengths.

The most basic visual style simply displays the complete range (see Fig. 6(b)). That is, we display the area in between the maximum and minimum extreme. We simplify the outliers to remove any (meaningless) noise (V3). Displaying the extremes does not give any information on the distribution of time series within the group. All elements are at least ε-connected, but the distribution is not guaranteed to be uniform or uni-modal. There may be large sections that are enclosed in between the extremes, but that are not supported by any time series in the trends. This is particularly the case when trends split into several sub-trends. This "illusion of presence" is also a problem for fan charts when the distribution of the data is multi-modal (¬V5).

We present two possible approaches to counteract this illusion. First, we may increase the information encoded in a trend. For example, we might encode the (complete) distribution by using a heatmap (see Fig. 7(a)). Introducing more information reduces the illusion of presence by increasing the accuracy of the representation (V3, V5). Of course there is a trade-off between simplification and information.

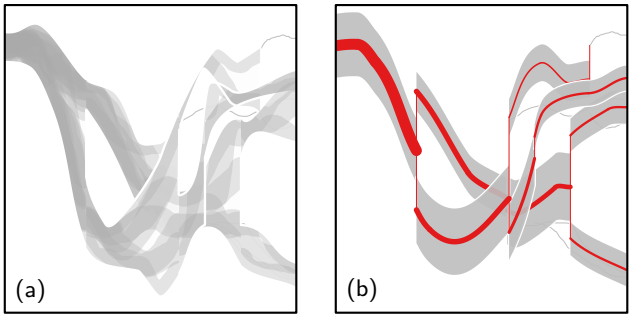

Fig. 7. (a) A heatmap approach displays more information dispelling the illusion of presence. (b) A uniform range per trend emphasizes inaccuracy, but connectivity with sub-trends is lost.

An alternative visual style is to reduce the presented information. By visually reinforcing the inaccuracy in the data, we reduce the illusion of presence. One way to achieve this is to display a representative range per trend. As representative range we pick the average range integrated over time. This range is presented for the complete trend independent of the actual variability in the trend. It is possible to compare the average range between trends, but within a trend all information is abstracted. The result is a harmonious display of the data as the group and centerline are in complete agreement (see Fig. 7(b)) (V3).

There are some visual artifacts associated with this style. First, a consistent range may be achieved by offsetting the centerline along the vertical axis. Curvature, however, will result in an appearance of non-uniform width. Hence the range might be interpreted as accurate by the user after all. A perpendicular offset of the centerline prevents this, but causes the range to be non-uniform along the trend. Consequently, it is hard to compare the range of different groups (¬V1). Second, trends and sub-trends become non-continuous. A discontinuity occurs at the common boundary as the mean range of trends and sub-trends do not match up (¬T2c, ¬V4). This visual discontinuity may be harmful in situations with many overlapping groups. Furthermore, it is a strong visual cue that has no firm grounding in the input data (¬V1).

To prevent discontinuities at the boundaries we simplify the boundary instead. We sample the range along the trend at regularly fixed intervals. These sample include at least the start and end of each trend. Using a low resolution sampling (up to three per group) we simplify the input but remove any visual discontinuity (see Fig. 8(a)).

4.2 High-level Structure

Our geometrical model clearly defines when a group (or trend) comes into existence and when it ceases to exist. However, the resulting trend boundaries are artificial in a sense, since they are the results of our need to group different items and as such only a second-order product of the input. Hence, we should take care to not overly emphasize the discontinuities at trend splits (V1, V4).

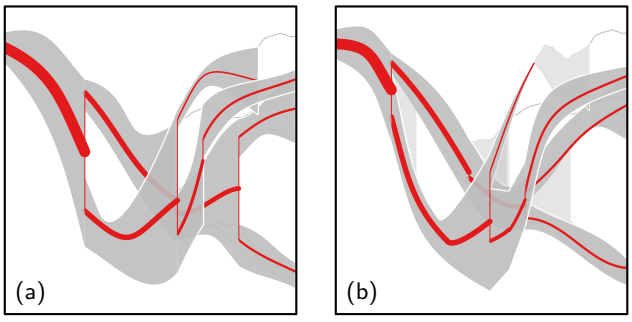

Fig. 8. (a) Simplifying the input smoothes out discontinuity while maintaining low visual complexity. (b) By de-emphasizing the area just before a split, we can reduce the visual saliency of the split itself.

Splits. Trend splits have high visual impact: just before a trend splits there must be an area of height ε that does not contain any time series and after the split this area does not exist anymore. If we display trends uniformly in their full range, there will hence be a strong visual break at a split (¬V1). The data, clearly, is neither uniform nor uni-modal around a split (after all, there is an ε-split between at least two of the items). We reduce the saliency of ε-splits by visualizing the area devoid of time-series occuring before an ε-split.

Let T be the trend that splits into two sub-trends. Let S be the time series that are part of the newly formed upper trend and S' the time series that are part of the lower trend. We compute the lower envelope of S and the upper envelope of S' (see Fig. 9(a)). Let i be the last intersection of these two envelopes that falls inside T. The two envelopes from i onwards define an area A inside T that is devoid of any supporting time series (see Fig. 9(b)). We draw A with reduced opacity, thus reducing the saliency of the trend split itself (see Fig. 8(b)).

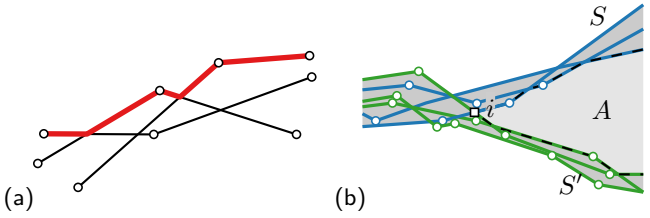

Fig. 9. (a) The upper-envelope is the pointwise maximum of a set of poly-lines (red). (b) The lower envelope of S and the upper envelope of S' define the area A devoid of items.

Centerline. While the centerlines reinforce the structure of the trends, the vertical connections between centerlines emphasize the visual discontinuities in the detected trends at splits (¬V1). We explore different visual styles to reduce the disruptive influence of these *connectors*.

The connectors between centerlines of trends are not part of the original input data. Hence, we aim to reduce their saliency. One option is to color the connectors differently to emphasize that they are not part of the data (see Fig. 10(a)) (V1). The different color connectors, however, make the visualization more complex and do not form a logical connection with the centerlines. Alternatively, dotting or dashing may help to de-emphasize the connectors (see Fig. 10(b)).

We may decide to not draw connectors at all. As discussed before, placing the centerlines along the mean implicitly encodes a uni-modal distribution. This is not always justified, specifically at splits. Hence, we can decide to reposition the centerline to prevent a visually salient split. Let T be the main trend and S the sub-trend that has the largest support among all sub-trends. We compute the mean centerline of T as well as the mean centerline of the subset of time series in T that are also part of S. The resulting centerline for T is the linear interpolation of both centerlines. Let x_s, x_e be the x-coordinate of the start, respectively end, of T. The final centerline $C(x)$ is positioned at $(x_e-x)/(x_e-x_s)\cdot T(x)+(x-x_s)/(x_e-x_s)\cdot S(x)$. For other sub-trends a similar connecting centerline is computed, that is cut off as soon as it intersects C. Connections to sub-trends may be displayed either dotted or regular (see Fig. 10(c)). Repositioning the centerline reinforces the trend structure, but reduces the grounding in the input data (¬V1, V4).

5 Exploring Trends

Our method supports three parameters to steer the automated trend detection: granularity, duration, and support (see Fig. 11). Varying the *granularity*, controlled by the ε-parameter, allows detection of coarser or finer trends. Filtering on *duration* prevents the detection of too short trends, and filtering on the *support* prevents trends with low support.

Trend Granularity. The detection of trends is based on a singular parameter that determines a threshold on the maximum allowed distance between consecutive elements. By changing the ε-parameter, the different granularities of trends may be investigated (T2a, T2b, T2c). A smaller value of ε results in strongly coherent trends and allows only minor deviations from the trend. While this may capture the most

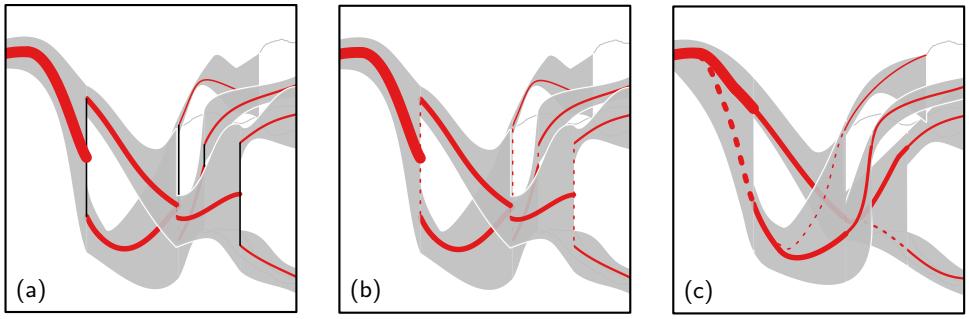

Fig. 10. (a) Encoding the connectors as artificial by using a different color. (b) Using dashing to encode the connectors. (c) Removing the need for connectors by interpolating the different centerlines.

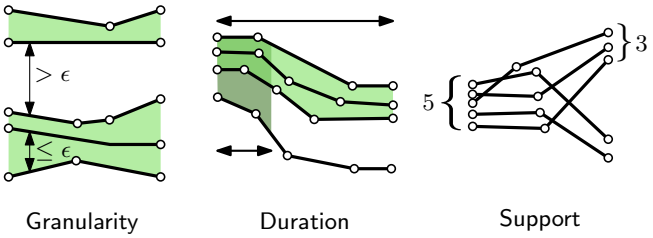

Fig. 11. The automated detection of trends can be steered via three natural parameters. Duration filters groups from the root of the tree, whereas support filters groups from the leaves.

strongly formed trends, small deviations may cause other trends to not be detected. Larger ε values allow more variation in the trends, ensuring that higher-level processes are captured in the trend detection.

Support Size and Duration. Besides the ε-parameter we allow the user to also filter on the size and duration of trends (T1a). The support parameter filters out trends that are not supported by a sufficient number of time series. As the trends form a forest, filtering on support removes small trends, starting at the leaves of the trees. By contrast, the duration parameter filters trends based on a minimum duration, starting from the roots of the trees in the forest. The combination of both is a powerful tool that allows the user to focus on specific trends.

5.1 Trend Interaction

Automated trend detection finds trends of arbitrary granularity in the data. However, users may be interested in the behavior of only a part of the data based on the information gathered (T2a). Manual interaction ensures data can be explored in an iterative manner while being supported by automated detection. Our approach supports three types of trend interaction (selection brushes, fixed initial configuration, and a time-sweep) as well as highlighting and selection.

Selection Brush. We support three types of brushes that allow selection of a subset of the original input time series (see Fig. 12(a)). The *all brush* selects only the data that passes through all brushed areas. This can be used to select data that follows a specific pattern (T2a, T2b). The *any brush* selects time series that pass through at least one of the brushed areas. It may be used to refine selection while allowing the user to select several diverging patterns. Finally, the *deselect brush* deselects all time series that pass through at least one of the brushed areas. It complements the *any brush*, allowing the same selective capabilities.

Initial Configuration. Users may want to study the behavior of different initial parameters settings (T2a, T2d). But if the subsets each diverge over time, this requires a coarse level of trend detection that may group all subsets together at the start. By selecting the initial configuration, users can define which subsets should remain separated (see Fig. 12(b)). Similar to the brushes, users can draw intervals at the initial time-stamp. Time series can only form groups within the same interval, so different starting conditions can be kept separate even when analyzing the data at a very coarse level. The initial configuration tool is more powerful in this sense than the basic selectionbrushes.

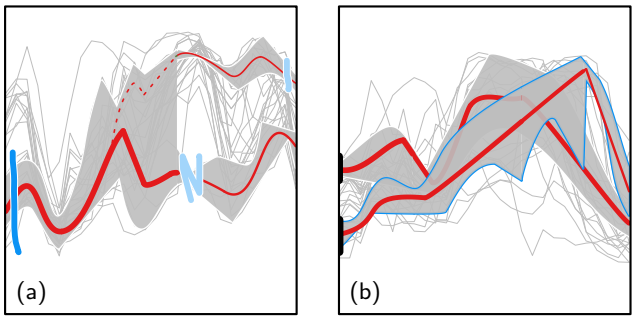

Fig. 12. (a) Selection brushes (*all brush* in dark blue and *any brush* in light blue) can be used to select the time series. Trend detection then operates only on the subset (3h forecast humidity at 2m in percent (BHM) [25]) (b) Selecting an initial configuration allows for trend detection in models with similar starting conditions. (3h forecast dew point temperature at 2m in degrees Fahrenheit (BHM) [25]))

Time Sweep. Our approach detects only trends that start at t_s. By changing the initial starting position we can also detect trends starting at different points in time. We visually encode the starting position by a vertical line at the start of the time series. The user can select and drag this line to initiate the algorithm from a later point in time (see Fig. 14). The time-sweep in combination with selection brushes allows exploration of correlations at later positions in time (T2a, T2c). The time-sweep allows us to find all the groups included in the complete grouping structure defined in Section 3.1. Nevertheless, for any particular setting only a subset of the groups are detected.

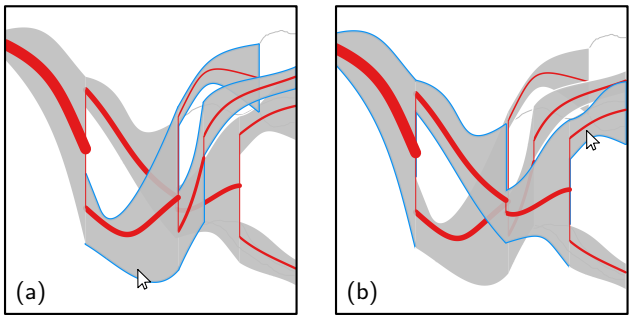

Fig. 13. Forward- and backward-highlighting of trends allows temporal exploration of structures. (a) Forward highlighting emphasizes trend development. (b) Backward highlighting emphasizes the trend origin.

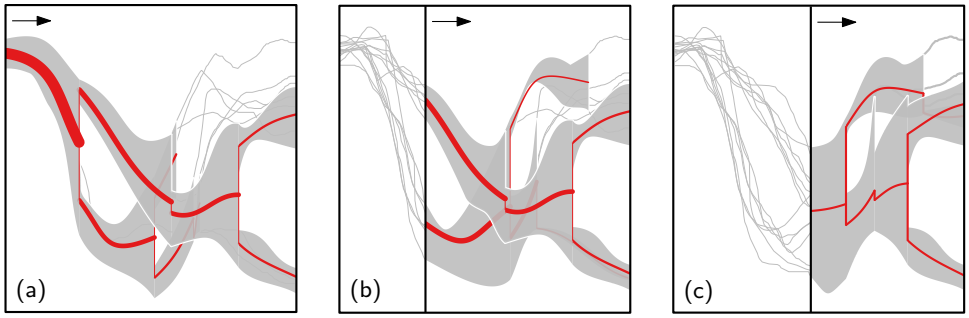

Fig. 14. (a-c) The time sweep enables trend detection starting from any point in time.

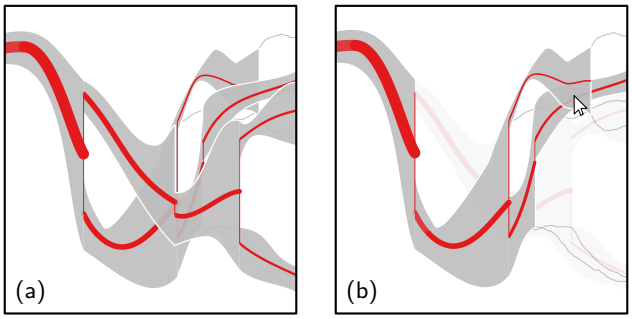

Fig. 15. (a) Basic trends detected in the input. (b) Selection helps focus on a subset of the trends using transparency and ordering.

Highlighting and Selection. Highlighting shows the structure of the currently selected trend (T2a, T2c, V4). *Forward highlighting* (see Fig. 13(a)) highlights all future possibilities branching out from the currently selected trend. *Backward highlighting* (see Fig. 13(b)) highlights the path of origin of the currently selected trend. It may be used to analyse coherence in the past tracks. Highlighting uses a colored casing for the selection - a distinctive light blue in Fig. 13. Alternatively, the highlighted selection may be displayed in a different color.

Selection allows trends to be layered in different orders. Complete subtrees can be moved to the front or the back of the view. The user can also select subtrees and place them partially transparent in the background to reduce visual clutter (see Fig. 15).

6 Example Applications

We now show how our system can be used to analyze real-world data sets. We first discuss a general time-series data set that concerns the value added by industry to the GDP (GapMinder; data provided by the World Bank [13]). Secondly, we explore an ensemble from the NOAA SPC Plume Viewer [25]. The ensemble is built around two core models with six positive and six negative perturbations per model. We illustrate how the different exploratory tools allow the detection of trends in the data and help correlate different time series.

Time-series data. The GapMinder data contains approximately 200 time series representing the value (in percentage) added by industry to the GDP of different countries across the world. Given the number of time series in the data set, direct visual analysis is infeasible. Animation can help reduce the amount of information displayed at any point in time, but might not be effective [36]. We display all information and combine automated trend detection with manual interaction to find key trends. As the data set is very dense we start our exploration with a very small setting of ε. The resulting trend structure appears to imply that there might be two major trends that start at the top and bottom of the data set (see Fig. 16(a)). We split the data set by fixing the initial intervals within which we detect trends. We can now safely increase ε and note that there appear to be two main trends that cross (see Fig. 16(b)). There is no clear outcome, though, as both trends now cover the complete area, so there might be a large bias in our observation. After changing the trend detection and checking different subsets, we notice that this trend is most strongly supported in the extremes of the initial distribution. We further restrict our input and note that, indeed, there is a strongly supported trend where the value added by industry significantly reduces (see Fig. 16(c)). Comparatively there is only a small trend where the value added by industry is on the rise.

Ensembles. The diverging nature of ensembles makes them a natural match with the trend structure we compute. The ensemble data we consider here are a forecast for the dew point – the temperature at which atmospheric water vapor condenses. There are two core models (the WRF-ARW and NMM-B NAM Model), and for each model 12 different perturbations. An initial exploration of the data indicates that there is a strong trend that splits off a minor sub-trend near the end (see Fig. 17(a)). We select different trends and subsets using the selection brushes. By refining the detected trends, we find that there are two very fine trends with reasonable support that significantly depend on the initial parameterization (see Fig. 17(b-c)). When we correlate the

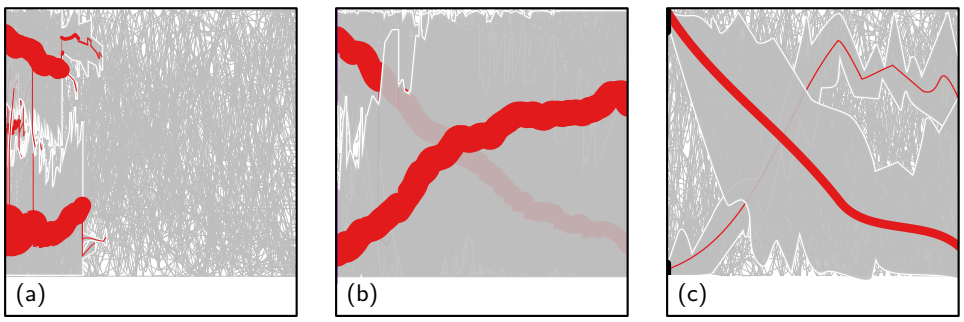

Fig. 16. (a-b) An initial exploration of the data indicates two strong trends starting in the upper- and lower-part of the data set and ending in the inverse corner. (c) Refining the search by restricting the initial values indicates a very strongly supported trend where for several countries the value added by industry strongly decreases, and a respectively minor trend where it strongly increases (Industry (% of GDP) [13]).

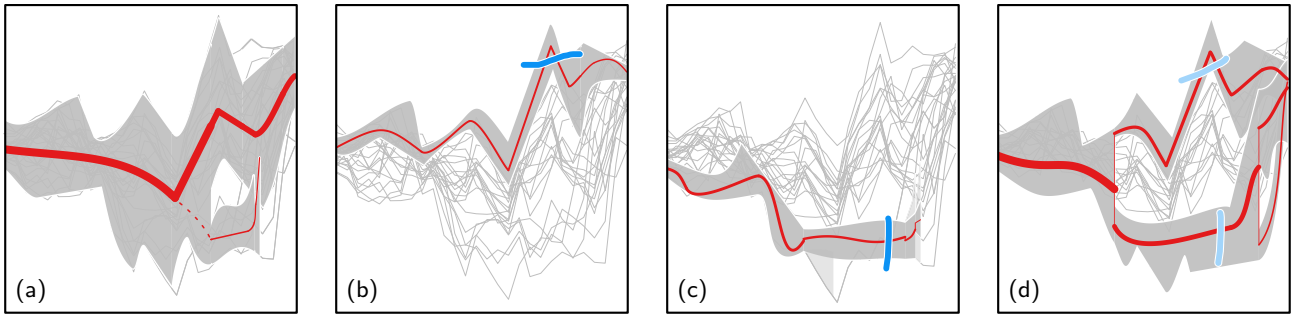

Fig. 17. (a) Initial exploration of the data implies a main trend (b-c) After further refining the search to the upper-/lower-part, two completely separate trends appear that strongly depend on the initial parameterization. (d) Visualizing both trends at the same time gives a complete overview, grouping both trends at the start in a single large trend (3h forecast dew point temperature at 2m in degrees Fahrenheit (EVW) [25]]).

time series back to the input data we note that the bottom trend consists solely of time series resulting from the WRF-ARW model (and mainly the negative perturbations). The top trend, by contrast, consists solely of time series resulting from the NMM-B NAM Model.

7 Limitations

While the computer-assisted analysis of trends using our visualization appears effective, there are some limitations. When many subtrends overlap, the visualization is problematic due to the occlusion of many subtrends (see Fig. 18(a)). While trends can manually be sorted on the z-axis, this does not help in getting an overview. It would be desirable to clearly visualize all subtrends even when they overlap. A solution could be to merge the different subtrends into larger groups, but interconnectivity between different subtrends should be maintained. This is problematic as subtrends might be interleaved. Animation, interaction and other modes of visual comparison [14] may be useful.

The detection of trends at different granularities helps to get an overview of the data and its structure. However, it may be desirable to detect trends at different granularities at the same time (see Fig. 18(b-c)). When a small group of entities is clearly separated from the rest of the ensemble, it is plausible to group them as a trend even when they are relatively far apart from each other. Currently the granularity of the detected trends is fixed across the complete ensemble. Therefore we cannot distinguish fine-trends and detect a coarser-level trend at the same time. This behavior is consistent with our model.

8 Discussion and Future Work

We proposed a geometric model that guarantees detection of trends (according to a precise mathematical definition) when they are present. Our model is based on three parameters: granularity, support-size, and duration. We explored different visual styles and showed how they relate to the primitive tasks that a trend-detection system should support. Finally, we introduced several tools for interactive exploration.

Combining automated trend-detection with interaction helps to quickly get an overview of the data and zoom in on trends in subsets of the data. Because trends are guaranteed to be detected, visual analysis is supplemented with automated detection and trends are less likely to be missed. While our approach looks promising, it is not validated by external users. Hence, in future work, it would be important to perform user studies to validate our approach.

We detect only those trends that start at time t_s. Naturally it would be interesting to also explore the complete trend structure, where trends may split into subtrends and then merge again into larger trends. The visualization of this braided graph of trends is a challenging problem, and might require the display of subtrends inside other trends as well as their connectivity.

Our visualization is not as simple as it could be. Trends sometimes have a jagged appearance and additional simplification and smoothing could be applied. However, such simplification necessarily removes the visualization further from the input data. After all, some sharp features might in fact be critical features of the data.

Furthermore, if a few time series split off from a main trend, they might significantly affect the outline. While such outlying time series could be completely eliminated from the trend-boundary, they are part of the detected (sub)trends. Removing them may significantly underrepresent the range and support of the trend.

Acknowledgments

This research started at the workshop "Geometric Algorithms in the Field" at the Lorentz Center. A.v.G. and B.S. are supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 612.001.102, and 639.023.208 respectively. F.S. is supported by the Danish National Research Foundation under grant nr. DNRF84.

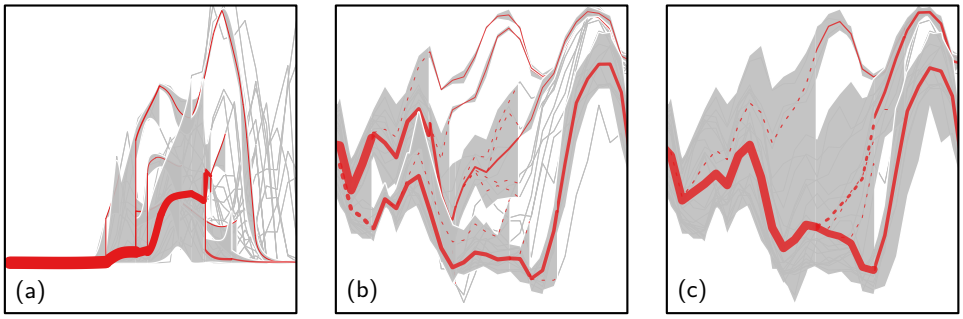

Fig. 18. (a) Overlap of subtrends causes occlusion of subtrends. (b-c) Concurrent different levels of granularity are required to capture all trends. Increasing the granularity causes a collapse of initial trends.

REFERENCES

[1] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[2] R. Allendes Osorio and K. Brodlie. Contouring with uncertainty. *Proceedings of Theory and Practice of Computer Graphics*, pages 59–66, 2008.

[3] G.-P. Bonneau, H.-C. Hege, C. Johnson, M. Oliveira, K. Potter, P. Rheingans, and T. Schultz. *Scientific Visualization: Uncertainty, Multifield, Biomedical, and Scalable Visualization*, chapter Overview and State-of-the-Art of Uncertainty Visualization, pages 3–27. Springer London, 2014.

[4] R. Brown. Animated visual vibrations as an uncertainty visualisation technique. In *Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPHITE)*, pages 84–89, 2004.

[5] K. Buchin, M. Buchin, M. van Kreveld, B. Speckmann, and F. Staals. Trajectory Grouping Structure. *Journal of Computational Geometry*, 6(1):75–98, 2015.

[6] L. Byron and M. Wattenberg. Stacked Graphs–Geometry & Aesthetics. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1245–1252, 2008.

[7] T. Cormen, C. Stein, R. Rivest, and C. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[8] L. Delle Monache, F. Eckel, D. Rife, B. Nagarajan, and K. Searight. Probabilistic Weather Prediction with an Analog Ensemble. *Monthly Weather Review*, 141(10):3498–3516, 2013.

[9] I. Demir, C. Dick, and R. Westermann. Multi-Charts for Comparative 3D Ensemble Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2694–2703, 2014.

[10] J. Dykes and C. Brunsdon. Geographically Weighted Visualization: Interactive Graphics for Scale-Varying Exploratory Analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1161–1168, 2007.

[11] F. Ferstl, K. Burger, and R. Westermann. Streamline Variability Plots for Characterizing the Uncertainty in Vector Field Ensembles. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):767–776, 2016.

[12] R. Gall, J. Franklin, F. Marks, E. N. Rappaport, and F. Toepfer. The Hurricane Forecast Improvement Project. *Bulletin of the American Meteorological Society*, 94(3):329–343, 2013.

[13] Gapminder (Data provider: World Bank). `http://www.gapminder.org/data/`, Mar 2016.

[14] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. Hansen, and J. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[15] G. Grigoryan and P. Rheingans. Point-Based Probabilistic Surfaces to Show Surface Uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 10(5):564–573, 2004.

[16] T. Haiden, L. Magnusson, I. Tsonevsky, F. Wetterhall, L. Alfieri, F. Pappenberger, P. De Rosnay, J. Muñoz-Sabater, G. Balsamo, C. Albergel, et al. ECMWF forecast performance during the June 2013 flood in Central Europe. In *ECMWF Technical Memorandum No 723*. 2014.

[17] J. Hintze and R. Nelson. Violin Plots: a Box Plot-Density Trace Synergism. *The American Statistician*, 52(2):181–184, 1998.

[18] D. Holten. Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[19] D. Holten and J. van Wijk. Force-Directed Edge Bundling for Graph Visualization. In *Computer Graphics Forum*, volume 28, pages 983–990, 2009.

[20] J. Kay, C. Deser, A. Phillips, A. Mai, C. Hannay, G. Strand, J. Arblaster, S. Bates, G. Danabasoglu, J. Edwards, M. Holland, P. Kushner, J.-F. Lamarque, D. Lawrence, K. Lindsay, A. Middleton, E. Munoz, R. Neale, K. Oleson, L. Polvani, and M. Vertenstein. The Community Earth System Model (CESM) large ensemble project: A community resource for studying climate change in the presence of internal climate variability. *Bulletin of the American Meteorological Society*, 96(8):1333–1349, 2015.

[21] Koninklijk Nederlands Meteorologisch Instituut (KNMI). `http://climexp.knmi.nl/start.cgi`, Mar 2016.

[22] I. Kostitsyna, M. van Kreveld, M. Löffler, B. Speckmann, and F. Staals. Trajectory Grouping Structure under Geodesic Distance. In *Proceedings of the 31st International Symposium on Computational Geometry*, 2015.

[23] M. Löffler, F. Staals, and J. Urhausen. New Results on Trajectory Grouping under Geodesic Distance. In *Abstracts of the 32nd European Workshop on Computational Geometry*, 2016.

[24] M. Mirzargar, R. T. Whitaker, and R. M. Kirby. Curve Boxplot: Generalization of Boxplot for Ensembles of Curves. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2654–2663, 2014.

[25] National Oceanic and Atmospheric Administration (NOAA) / National Weather Service / Storm Prediction Center Plume Viewer. `http://www.spc.noaa.gov/exper/sref/srefplumes/`, Mar 2016.

[26] H. Obermaier and K. Joy. Future Challenges for Ensemble Visualization. *IEEE Computer Graphics and Applications*, 34(3):8–11, 2014.

[27] C. Perin, F. Vernier, and J.-D. Fekete. Progressive Horizon Graphs: Improving Small Multiples Visualization of Time Series. In *IEEE Conference on Information Visualization (INFOVIS)*, 2012.

[28] V. Peysakhovich, C. Hurter, and A. Telea. Attribute-Driven Edge Bundling for General Graphs with Applications in Trail Analysis. In *IEEE Pacific Visualization Symposium (PacificVis)*, pages 39–46, 2015.

[29] K. Pöthkow and H.-C. Hege. Positional Uncertainty of Isocontours: Condition Analysis and Probabilistic Measures. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1393–1406, 2011.

[30] K. Pöthkow and H.-C. Hege. Nonparametric Models for Uncertainty Visualization. *Computer Graphics Forum*, 32(3pt2):131–140, 2013.

[31] K. Potter, J. Kniss, R. Riesenfeld, and C. Johnson. Visualizing Summary Statistics and Uncertainty. *Computer Graphics Forum*, 29(3):823–832, 2010.

[32] K. Potter, P. Rosen, and C. Johnson. From Quantification to Visualization: A Taxonomy of Uncertainty Visualization Approaches. In *Uncertainty Quantification in Scientific Computing*, pages 226–249. 2012.

[33] K. Potter, A. Wilson, P. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. Johnson. Ensemble-Vis: A Framework for the Statistical Visualization of Ensemble Data. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 233–240, 2009.

[34] H. Reijner. The Development of the Horizon Graph. In *Proceedings of the Vis08 Workshop From Theory to Practice: Design, Vision and Visualization*, 2008.

[35] M. Roberts, P. Vidale, M. Mizielinski, M.-E. Demory, R. Schiemann, J. Strachan, K. Hodges, R. Bell, and J. Camp. Tropical Cyclones in the UPSCALE Ensemble of High-Resolution Global Climate Models. *Journal of Climate*, 28(2):574–596, 2015.

[36] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko. Effectiveness of Animation in Trend Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, 2008.

[37] J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead. Noodles: A Tool for Visualization of Numerical Weather Model Ensemble Uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1421–1430, 2010.

[38] I. Tobin, R. Vautard, I. Balog, F.-M. Bréon, S. Jerez, P. M. Ruti, F. Thais, M. Vrac, and P. Yiou. Assessing climate change impacts on European wind energy from ENSEMBLES high-resolution climate projections. *Climatic Change*, 128(1-2):99–112, 2015.

[39] A. van Goethem, M. van Kreveld, M. Löffler, B. Speckmann, and F. Staals. Grouping Time-varying Data for Interactive Exploration. In *Proceedings of the 32th Annual Symposium on Computational Geometry*. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

[40] M. van Kreveld, M. Löffler, F. Staals, and L. Wiratma. A Refined Definition for Groups of Moving Entities and its Computation. In *Abstracts of the 32th European Workshop on Computational Geometry (EuroCG)*, 2016.

[41] B. Zehner, N. Watanabe, and O. Kolditz. Visualization of gridded scalar data with uncertainty in geosciences. *Computers & Geosciences*, 36(10):1268–1275, 2010.