

# Trajectory-Based Visual Analysis of Large Financial Time Series Data

**Tobias Schreck**  
Technische Universität Darmstadt  
tschreck@gris.informatik.tu-darmstadt.de

**Tatiana Tekušová**  
Fraunhofer IGD Darmstadt  
tatiana.tekusova@igd.fraunhofer.de

**Jörn Kohlhammer**  
Fraunhofer IGD and Technische Universität Darmstadt  
joern.kohlhammer@igd.fraunhofer.de

**Dieter Fellner**  
Fraunhofer IGD and Technische Universität Darmstadt  
d.fellner@igd.fraunhofer.de

## ABSTRACT

Visual Analytics seeks to combine automatic data analysis with visualization and human-computer interaction facilities to solve analysis problems in applications characterized by occurrence of large amounts of complex data. The financial data analysis domain is a promising field for research and application of Visual Analytics technology, as it prototypically involves the analysis of large data volumes in solving complex analysis tasks.

We introduce a Visual Analytics system for supporting the analysis of large amounts of financial time-varying indicator data. A system, driven by the idea of extending standard technical chart analysis from one to two-dimensional indicator space, is developed. The system relies on an unsupervised clustering algorithm combined with an appropriately designed movement data visualization technique. **Several analytical views on the full market and specific assets are offered for the user to navigate**, to explore, and to analyze. The system includes automatic screening of the potentially large visualization space, preselecting possibly interesting candidate data views for presentation to the user. The system is applied to a large data set of time varying 2-D stock market data, demonstrating its effectiveness for visual analysis of financial data. We expect the proposed techniques to be beneficial in other application areas as well.

**Keywords:** Visual Analytics, Trajectory Clustering and Visualization, Self-Organizing Maps, Financial Data.

## 1. INTRODUCTION

Visual Analytics is an emerging research discipline aiming to extract useful information and knowledge from large data volumes. Exploration of large data volumes to find potentially useful information is a challenging task. While several research areas try to achieve similar goals, the driving idea of Visual Analytics is to combine both automatic, intelligent data analysis methods with highly effective visualization and interaction facilities. In the Visual Analytics model, automatic data analysis is applied to screen, filter, sort, aggregate and abstract the raw data. The user interacts with the preprocessed intermediate results, requesting specific data views to confirm or reject hypothesis as they emerge. Designing effective Visual Analytics systems for a given ap-

plication domain not only requires a sound understanding of the problem domain. It also requires selection, adaption or development of suitable analysis, visualization, and interaction techniques. A promising application domain for Visual Analytics is financial analysis. It features **immense amounts of multivariate and time-dependent data**, which need to be effectively processed by the financial analysts. Analysis tasks in finance are rarely completely well-defined in advance. Rather, hypothesis generation and verification drives the analysis process.

Some of the most fundamental analysis tasks in the financial data analysis domain include the identification of market trends and the prediction of asset prices. While many analysis tools are used in practice, technical chart analysis is among the most widely used tools. Technical chart analysis relies on the graphical analysis of charts plotting single indicators against time (we therefore call it 1-dimensional technical chart analysis). However, often, the *co-behavior* of two indicators over time is of special interest. Visual support for this kind of technical chart analysis is regarded highly desirable, but so far remains a largely unsolved problem.

In this paper, we address this shortcoming by introducing a new framework for visual analysis of time-dependent data in *2-dimensional* financial indicator space. Plotting 2-D time-dependent financial indicator data yields 2-D trajectories. In our approach, these trajectories are input to an automatic clustering preprocessing step which identifies prominent patterns in the data. We propose a number of effective visualizations for the clustered trajectory data, and build a system for visual analysis of large 2-D time-dependent financial data sets. The system allows interactive exploration and visual analysis of the data. It also includes basic automatic analysis capabilities helping the user to quickly identify interesting visualization views, narrowing down the huge search space, which otherwise is given in visual data exploration with no a-priori knowledge. We believe that main findings of this work are applicable to other problem domains relying on trajectory-based visual analysis.

The remainder of this paper is structured as follows. Section 2 introduces key concepts of the financial application domain, followed by a review of related work in Section 3. Section 4 then introduces our pattern extraction and visualization approach, which in Section 5 is applied on a large financial data set. Finally, Section 6 concludes and outlines future work in the area.

## 2. FINANCIAL DATA ANALYSIS

In the international financial markets, data providers such as Bloomberg, Reuters or Thompson Financial offer access to large streams of data taken in real time from international electronic trading systems. Financial analysts need to effectively and efficiently analyze the underlying large amounts of data to make good investment decisions or offer financial consulting services. Faced with significant competition, the analysts need to arrive at the best possible decisions in a short amount of time, leveraging as much information as possible. Many techniques exist for the analysis of stock market data, however *technical chart analysis* and *analysis of correlations between financial indicators* are among the most popular ones.

Technical chart analysis [20; 5] relies on the graphical analysis of the development of a single financial indicator over time. Typical indicators are, e.g., the price or volume traded of a given asset. The analysis is expected to be useful for the identification of market trends, and it is believed by its users that it is also helpful for correctly predicting future market developments. Technical chart analysis is criticized as it mostly neglects “fundamental” market indicators in the analysis. However, it is quite popular and in wide use owing to its simplicity and illustrative power. Technical analysts search for occurrences of certain chart patterns, e.g. the so-called “head and shoulders reversal patterns”, and graphically identify “support” and “resistance” thresholds, to predict future developments.

In the analysis of indicator correlations, the relationship between financial indicators is conventionally analyzed using scatter plots. In a widely used type of scatter plot, two of the most important financial analysis indicators, asset *return* and asset *risk*, are mapped to the diagram axes. According to typical observations, higher asset risks usually imply higher return rates, and consequently, financial assets can be categorized in risk-return space [16] as shown in Figure 1. As risk and return indicators of assets usually vary over time, also the correlation of the 2-dimensional observations along the *time axis* is considered important, apart from correlating indicators among cross-sections of the data at a given point in time. In a scatter plot framework, this can be supported by displaying *trajectories* of asset movements over time-dependent movements in 2-dimensional indicator space [3; 24; 28].

## 3. RELATED WORK

In recent years, a number of visualization techniques have been proposed specifically for financial data. A comparison of the effectiveness of some of them can be found in [19]. A visualization technique specifically popular in the financial domain is the TreeMap visualization, as represented e.g., by the Map of the Market [26] system. It represents assets in a TreeMap view encoding market volume by area and asset return by color-coding. Visual analysis of the distribution of assets returns over all possible time intervals in a given time horizon is possible by growth matrices (return triangles) [12]. Long time series of asset prices can be effectively displayed by so-called Fisheye views [18]. As analysts often wish to combine quantitative asset information with actual financial news, a bimodal visualization was developed in [27]. The composition of fund portfolios and their changes over time have been visualized in a 2.5D-view in [4]. In [28; 21] stock

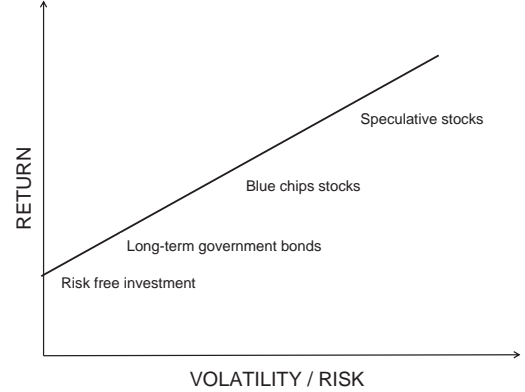


Figure 1: Risk-Return diagram and various asset types classified inside this 2-D indicator space.

market development has been visualized using trajectories of stock indicator movements.

Visualization and knowledge discovery in time series data is a research topic of its own. A survey on time series visualization approaches is presented in [1]. Time series knowledge discovery methods often require an appropriate definition of similarity between time series or subintervals thereof, and many subtle implications regarding data preprocessing and normalization arise. A good overview over the topic can be found in a series of tutorials given by Eamonn Keogh [13].

The analysis of chart movements can be regarded as an instance of the trajectory analysis problem treated in Spatio-Temporal and Geographic Information Systems research. Respective work in that area deals with movement data observed in real-world coordinates, with applications, e.g., in traffic and transportation control, facility monitoring, or socio-political analysis applications. In these applications, characterization and understanding of movement patterns is at the center of the analysis. Work on analysis and visualization of GIS trajectory data is reported in [2; 23]. Recent results applicable to facility monitoring are given in [11].

Previously, we implemented a visual analysis system supporting financial data analysis [28]. The system supported the analysis of time-dependent risk-return data, allowing visual exploration by drawing trajectories along interpolated risk-return observations. While the system works well for a limited number of assets and short observation horizons, the display quickly gets cluttered with increasing data set size (cf. Figure 2). In this paper, we therefore develop new analysis and visualization methods for accommodating large amounts of time-dependent risk-return data.

## 4. EXTRACTION AND VISUALIZATION OF CHART PATTERNS

Visual analysis of non-trivial amounts of 2D time-dependent data requires appropriate data aggregation and abstraction to support the analysis of the most important data features. We describe our approach to reduce long, complex risk-return trajectories to salient trajectory patterns.

### 4.1 Calculation of Chart Movement Patterns

The data under concern consists of time-dependent observa-

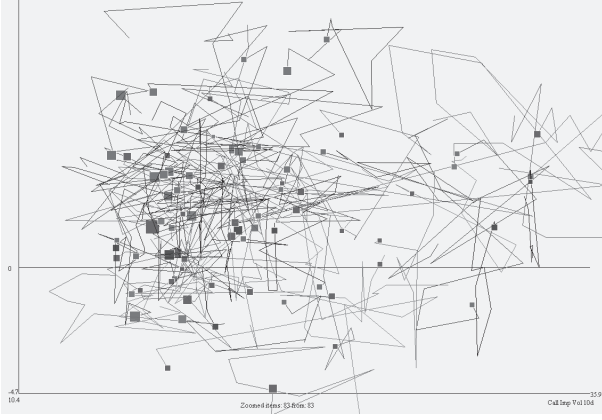


Figure 2: Direct visualization of risk-return trajectory data quickly leads to crowded, ineffective displays.

tions in 2-D risk-return space  $(\sigma, \pi)_t^a$ , where  $(\sigma, \pi)$  refers to pairwise measures of risk  $\sigma$  and return  $\pi$ , observed for each financial asset  $a \in A$  from a set of assets  $A$ , and for each time stamp  $t \in [0, T - 1]$  of the observation horizon. Linear interpolation between all consecutive observations  $(\sigma, \pi)_t^a$  and  $(\sigma, \pi)_{t+1}^a$  results in risk-return movements (or trajectories) for the individual assets (see Figure 2 for an illustration).

Reducing large amounts of trajectory data to smaller numbers of salient patterns can be achieved by cluster analysis. Recently, clustering of trajectory data has received considerable attention in the geo-spatial database and in related research areas. Clustering of trajectory data requires the definition of an appropriate similarity concept. In principle, many different similarity concepts are possible based directly on the geometry of the trajectories, or on derived numeric, structural, or symbolic features. Furthermore, the space of trajectory similarity notions is enhanced by data preprocessing choices such as normalization steps applied, and level of detail considered.

In this paper we rely on a pragmatic approach by combining a simple data representation scheme with a proven, robust clustering algorithm suited for visualization. In the financial analysis domain, days and weeks are meaningful analysis granularities. We therefore decompose the risk-return sequences of each asset  $a$  into slices of 5 consecutive, daily observations representing full trading weeks (Monday to Friday), obtaining trajectories consisting of 5 vertices and 4 segments linearly interpolating the vertices. Let

$$s_w^a = [(\sigma, \pi)_{\tau(w)}^a, \dots, (\sigma, \pi)_{\tau(w)+4}^a]$$

denote the sequence of daily risk-return observations for asset  $a$  during week  $w$ , where  $w \in W$  is an index over the set of weeks  $W$  contained in the observation horizon, and  $\tau(w) : w \rightarrow t$  is a function yielding the time stamp of Monday in week  $w$ . The task of the clustering algorithm is to reduce the (possibly large) set of weekly risk-return sequences  $S = s_w^a | a \in A, w \in W$  to a smaller set of  $n$  prototype trajectories  $P = \{p_1, \dots, p_n\}$ , where  $n$  is given by the user, or determined by unsupervised cluster validity analysis.

We represent each weekly trajectory segment  $s_w^a$  by a simple 10-dimensional feature vector obtained by concatenating its sequence of normalized  $(\sigma, \pi)$  coordinates. Normalization is done by linearly scaling each sequence to span the interval

$[0, 1]^2$  (local min-max normalization). Note that this representation implies invariance w.r.t. position and scale, but not w.r.t. rotation or more complex geometric transformations. Specific applications may require such invariances, which in turn require the definition of corresponding trajectory features. In Section 5 it will be demonstrated that already this simple representation leads to meaningful results, and our visual analytics framework is ready to accommodate additional vectorial trajectory representations as required by the application.

For clustering, we apply the well-known Self-Organizing Map (SOM) algorithm originally introduced by Kohonen [14]. We chose this algorithm as it is a robust algorithm which has been successfully used in numerous applications involving large standard and non-standard data sets, including HTML documents [10], audio [22] and image [17] data. It has also been successfully applied to 1-D financial time series data [30]. The SOM approach does not only represent large data sets by a user-defined fixed number of prototypes, but it also yields an approximately topology-preserving mapping of prototype vectors to positions on a regular, low-dimensional grid (usually, SOMs based on a 2-D grid are generated). This mapping can well be exploited for data visualization purposes [29]. We next introduce several visualizations for analysis of trajectory data in combination with information extracted from the SOM clustering.

## 4.2 Trajectory Bundle Visualization

A basis building block in our visual analysis system is the appropriate visualization of the distribution of trajectory samples w.r.t. a prototype trajectory. Visualizing a prototype trajectory together with a few associated samples of limited length is straightforward. For example, for a given week and small subset of assets, we may simply overlay the respective polyline paths  $s_w^a$  over the associated prototype  $p$ , connecting corresponding path segments by dashed alignment lines (cf. Figure 3 (a)). We indicate start and end points of the prototype trajectory at  $\tau(w)$  and  $\tau(w)+4$  using green (start) and red (end) markers. Overlaying more than a few sample trajectories, the display quickly gets crowded. We therefore resort to omitting rendering of the polylines directly, but instead use a coloring scheme such that color intensity reflects the density of trajectories around the given prototype. We fill the (possibly self-intersecting) polygons given by each pair of corresponding prototype and sample trajectory segments (cf. Figure 3 (b)) using a basic semi-

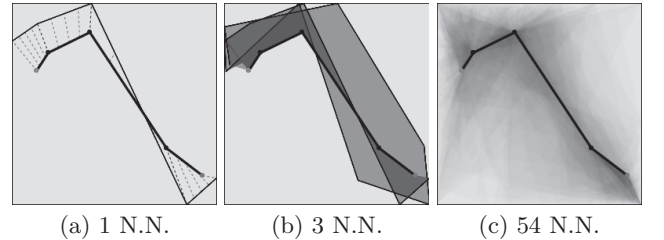


Figure 3: Prototype trajectory (thick polyline) and one (a) and three (b) nearest neighbor (NN) sample trajectories. The *trajectory bundle* visualization (c) uses an overlay of transparently colored segments to indicate the spatial distribution of many trajectories simultaneously.

transparent color. Specifically, we set the transparency proportional to the overall number of trajectory samples to visualize. The effect is that areas sharing much segment overlap get colored more intensively, naturally communicating trajectory density information by a transparent “veil” around the prototype trajectory. We like to call this technique the *trajectory bundle* visualization, noting its scalability for large numbers of trajectory samples. Figure 3 (c) illustrates a trajectory bundle consisting of 54 samples. We point out that the trajectory bundle visualization has a foundation in an analytical trajectory dissimilarity metric, as it can be interpreted as a visual generalization of the recently proposed *LIP* trajectory dissimilarity metric [23] to *sets* of trajectories. The technique was also inspired by the so-called *opacity bands* visualization originally proposed for Parallel Coordinate plots [7].

### 4.3 Task-centered View Design and Intelligent Interaction Support

In collaboration with financial analysts, around these clustering and visualization building blocks, we designed several analytical views and interactive querying facilities. Three main views were identified to be most desired. We briefly sketch them in the following, and apply them in the next section. As we are representing many weekly trajectories with a smaller number of trajectory prototypes, the first question naturally occurring is that of the overall distribution and quality of the patterns identified in the data set. The *market view* allows to assess this question by rendering the trajectory prototypes and represented data samples on the SOM grid, using the trajectory bundle visualization. Restricting this view to a selected asset results in the *asset view*, which allows the analysis of the distribution of chart movement patterns occurring for a given asset of interest. This view can also be used to visualize the transition of weekly patterns for a given asset. The third main view is the *pattern sequence view*, which is a comparative view on the time-dependent sequence of patterns of many assets simultaneously. It allows specifically to search for co-occurring and correlated patterns among time and assets. In Section 5, these views will be described in more detail and practically applied to real data.

The analytical process might be driven by specific *hypotheses* the user wishes to evaluate on the views, predetermining views and parameter selections. We also support the *hypotheses free* exploration by several analytical data analysis functions, the output of which is used for generating interesting candidate visualizations for presentation to the user. Specifically, we implemented several functions to automatically search for outliers, most frequent patterns, and correlations among the distribution of patterns across time and assets. The next section will give examples for these functions and their usage for automatically proposing candidate views to the user.

## 5. APPLICATION

In this section, we visually analyze chart movement patterns in a large real-world financial data set, demonstrating the effectiveness of our approach.

### 5.1 Data Set and Pattern Extraction

The data set used contains 83 blue chips European stocks for which daily risk and return data was recorded between

May 2005 and November 2006. Decomposing the data set into full trading weeks yielded 66 weeks of observations, totaling 5478 chart movement (trajectory) samples as input to the SOM-based cluster analysis. We normalized each sample as described in Section 4.1, and used that data to train a Self-Organizing map consisting of  $12 \times 9$  prototype vectors arranged on a grid of rectangular topology. The SOM learning algorithm was configured using standard parameter settings as suggested in [15]. Owing to the topology-preserving properties of the SOM, neighboring prototypes show similar patterns. In the remainder of this Section, we exploit this important characteristic in form of a grid-based distance measure used for data preprocessing as well as to control certain visualization characteristics.

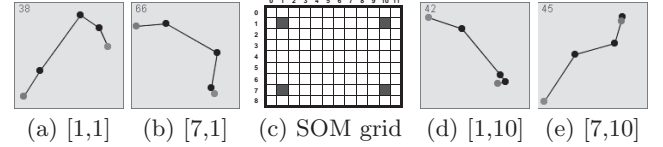


Figure 4: Prototype patterns occurring at the corner areas of the SOM grid (coordinates indicate row and column).

Inspection of the generated Self-Organizing Map indicates that a suitable, meaningful clustering result was obtained. The SOM nicely organizes the space of movement patterns by arranging prototype trajectories on the SOM grid such that neighboring patterns are similar to each other, and the different patterns are smoothly transitioning over the map. Figure 4 shows four example movement patterns located at the corner areas of the SOM. Note that the patterns roughly represent the four possible diagonal movements in the risk-return chart space, e.g., pattern (a) represents a decline in both dimensions, while pattern (e) represents the opposite direction. These four pattern types reflect the most salient, discriminative chart movements possible, and serve as a good starting point for interpretation of the pattern distribution w.r.t. the given SOM grid.

In the following, we rely on that data preprocessing result. We note that based on application considerations, different data segmentation (e.g., usage of monthly or yearly intervals) or movement representation (e.g., usage of position dependent features) could easily be integrated into our system. Also, the number of SOM prototype vectors is a parameter which can easily be adapted to application requirements. We here use about 100 chart movement prototypes as a compromise between accuracy of the representation, and level of abstraction achieved.

### 5.2 Market View

The market view gives a global view on the distribution of chart movements of all the assets and over the full time horizon. We visualize the market view by mapping each sample trajectory  $s_w^a$  to its nearest neighbor trajectory prototype  $p^* : p^* \in P \wedge \forall p \in P, p \neq p^* : d(p^*, s_w^a) \leq d(p, s_w^a)$ , using the  $L_1$  norm between the trajectories feature vectors as the distance function  $d()$ . We construct the market view by visualizing the trajectory bundles at the location of their underlying prototype pattern on the SOM grid. In conjunction with the SOM clustering process this yields an effective overview of the market-characteristic chart movement patterns. The display also effectively visualizes the quality



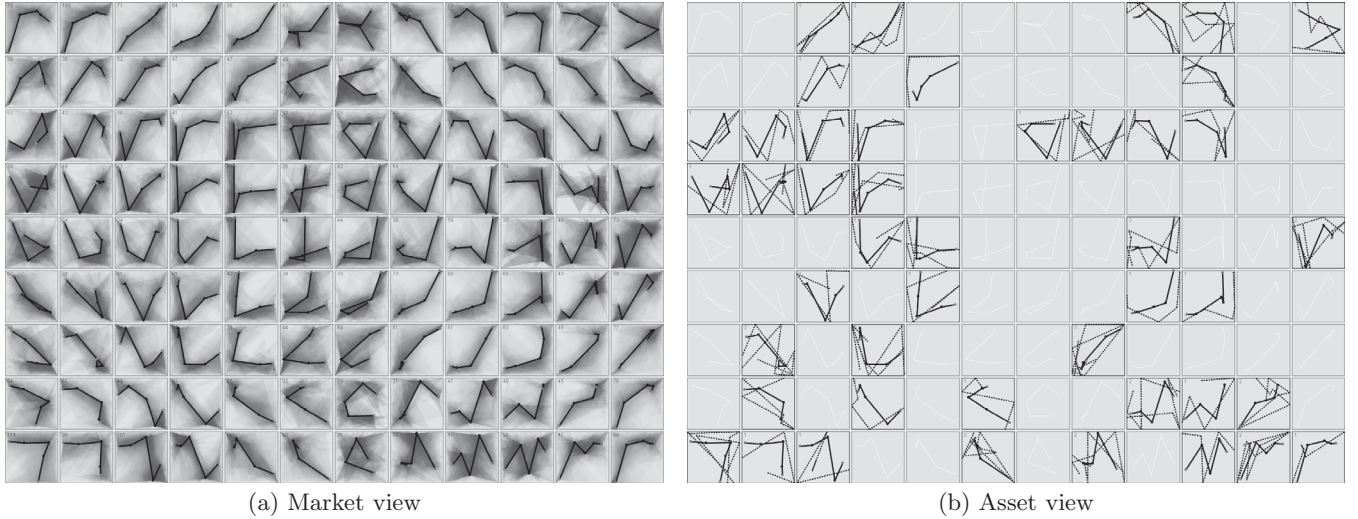


Figure 5: The market view (a) visualizes the distribution of chart movement patterns and the quality of representation of data samples by prototypes. Owing to the topology-preserving properties of the SOM algorithm, the map can be meaningfully interpreted in terms of pattern transitions. The asset view (b) is a version of the market view, restricted to a given asset of interest. It shows the distribution of patterns occurring for a specific asset, and over the full time span.

and frequency of the representation of actual samples by the prototypes. The larger the variance of the trajectories around their associated prototype is, the more intense the respective bundle visualizations get. A simple yet powerful interaction is achieved by filtering the set of sample trajectories by user-defined time subintervals. Thereby, the user can easily obtain an understanding of the distribution of patterns over time, and analyze for inter-temporal pattern correlations.

Figure 5 (a) shows all movement prototypes on the SOM grid in the market view. As indicated in Section 5.1, the SOM grid organizes the four possible diagonal chart movements in the corner area of the map, with smoothly transitioning patterns in between. The middle area shows more complex patterns, with circular or self-intersecting characteristics. In addition, the pattern fit to the actual data is shown by trajectory bundles, communicating the frequency of matched data samples and the variance around their associated prototypes.

### 5.3 Asset View

The asset view is obtained from the market view by restricting the set of sample trajectories to a selected asset. Thereby, individual assets can be analyzed for occurrence of specific chart movement patterns. Practically, due to the small number of matches per prototype, we directly overlay each matched sample to its prototype, and keep the patterns not matched by any sample in context by rendering only the prototype in a lighter color. For the matched prototypes, we use a coloring of the background to indicate the number of represented samples, scaling the color saturation proportional to the maximum occurring frequency.

Figure 5 (b) shows the asset view for a selected stock from our data set (Schering AG). Examining the view, the following observations stand out. Although, the market view shows a high frequency of patterns of type (a) in Figure 4, these do not occur for the selected asset. Also, the asset

does not show complex chart movement patterns located centrally on the prototype grid. Some of the most frequent patterns occur in the bottom-right area of the map, meaning that these patterns are typical for this asset. Based on the asset view, a semi-automatic alert system can be set up, monitoring current chart patterns, notifying the analyst of possibly atypical patterns occurring in real time.

Another view may be constructed based on the asset view, allowing visual analysis of the temporal sequence of patterns for a selected asset. As each asset sample trajectory can be matched to a prototype trajectory, the sequence of weekly chart movements can be visualized by connecting the respective SOM prototype positions. By restricting the distance between two consecutive weekly movements, the user can easily filter for gradual or abrupt inter-week pattern transitions. This is demonstrated in Figure 6, which shows abrupt inter-week pattern sequences for several assets, by filtering for a minimum grid distance of 14. The view allows the identification of weeks where the chart movement in the following week roughly reversed. Filtering for small distances on the other hand would reveal periods of roughly recurring weekly patterns, or smoother transitions thereof.

We state that depending on the selectivity of the filtering, overplotting effects could arise in this basic line-based view. A solution would be to rely on more advanced approaches for visualization of larger numbers of pattern connectors. To this end, an adoption of the recently introduced edge bundle technique [9] seems promising.

### 5.4 Sequence View

The sequence view is a comparative view of chart movement patterns, for all assets over time. The view is organized in a row-by-column scheme where each row refers to an asset, each column refers to a given week, and each cell contains the prototype representation of the actual chart movement sample. It combines the advantages of the familiar, linear time flow with the representation of individual chart move-

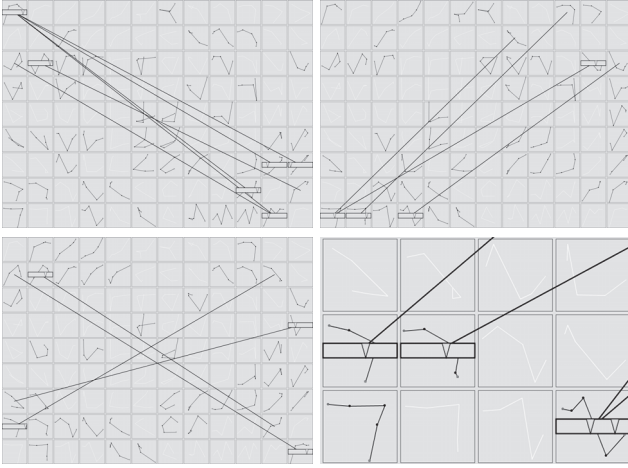


Figure 6: Asset views showing abrupt inter-week pattern transitions. Time tick scales may be used to indicate the date of occurrence of the pattern transition, relative to the global time scale (see bottom-right for a closeup).

ments by their best matching prototypes.

One main use case of this view is visual analysis for correlations, co-occurrences, and frequency of patterns. To this end, an automatic preprocessing of the sequence view is undertaken using statistical evaluation methods. The results of this analysis are input for the visualization, controlling filtering and highlighting. In particular, we implemented a two-stage analysis scheme for automatic identification of possibly interesting view configurations. Firstly, the algorithm calculates the Entropy measure [6] for the distribution of movement patterns across weeks. In the second stage, an analysis of pattern frequency considering the grid-based pattern distances finds the most prominent patterns in the weeks of lowest Entropy. These patterns represent those market trends in the identified weeks, where presumably, important global factors dominate the market dynamics.

The results of the Entropy analysis are presented to the user in form of thumbnail sequence views. To easily spot weeks with dominant market dynamics, a sorting of rows (assets) according to distance from the market trend is undertaken before the results are visualized. In the views, the prominent patterns are highlighted. Our highlighting scheme assigns highlighting color saturation to reflect the similarity of each sequence pattern to the identified prominent pattern. Specifically, we assign three highlight color saturation grades centered around the selected pattern as shown in Figure 7 for two different patterns.

Figure 8 illustrates four candidate patterns proposed by the automatic analysis for inspection by the user. Based on these thumbnail images, the user may generate a detailed view for closer inspection. Owing to the large amount of input data, the detailed sequence view requires a high resolution display. The thumbnails form only a preview of the data, and full view is ideally supported by a high-resolution desktop display. We note that the sequence view is also an interesting application for usage with large-scale displays such as the HEyeWall<sup>®</sup> [8] or PowerWall [25] systems.

Figure 9 shows the sequence view with two selected patterns highlighted. It is the most detailed view which at the same

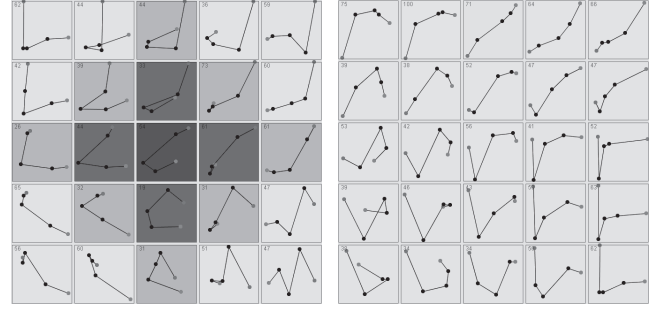


Figure 7: Two trajectory patterns located at positions [6,6] (blue) and [0,2] (yellow) on the market view displayed in Figure 5. Neighboring patterns with grid distance 1 and 2 are highlighted at decreasing saturation. This color-coding is used in the sequence view shown in Figure 9.

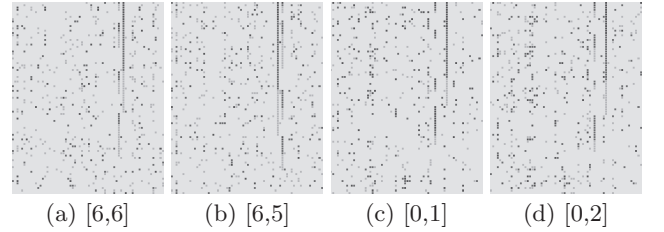


Figure 8: Candidate sequence visualizations.

time, suppresses small asset-specific detail and noisy patterns (the reader is encouraged to zoom into the image for closer inspection). The full sequence view supports visual analysis of the distribution and correlation of the two selected patterns over the full time period. The pattern highlighted in blue occurs most prominently in the 46<sup>th</sup> and 48<sup>th</sup> weeks, while the yellow highlighted pattern does so in week 45 and 50. The sorting of the asset rows by the first pattern in the two most frequent weeks reveals (a) a strong correlation between the blue pattern for weeks 46 and 48, and (b) correlations between the two patterns for subsets of the assets. This is an interesting finding, as the yellow pattern represents a decrease in both risk and return dimensions, while the blue pattern indicates an increase in risk combined with a moderate change in return. This insight encourages further investigation of the analyst, considering additional information sources such as a financial news archive.

## 6. CONCLUSIONS

We presented a Visual Analytics system for pattern-based analysis of 2-dimensional time-varying chart data. Focusing on the financial domain, the system was tailored to reflect domain-dependent time granularities and trajectory similarity notions, but we expect the general approach to be applicable to other important domains, e.g., traffic control, facility monitoring, or socio-political applications. A key feature of our solution is the tight coupling of automated data mining methods (specifically, cluster and statistical analysis) with appropriately designed, effective visualization techniques. Interactive view manipulation and data querying facilities allow for deep explorative data analysis and hypothesis generation.

In Visual Analytics, it is generally recognized that intelligent



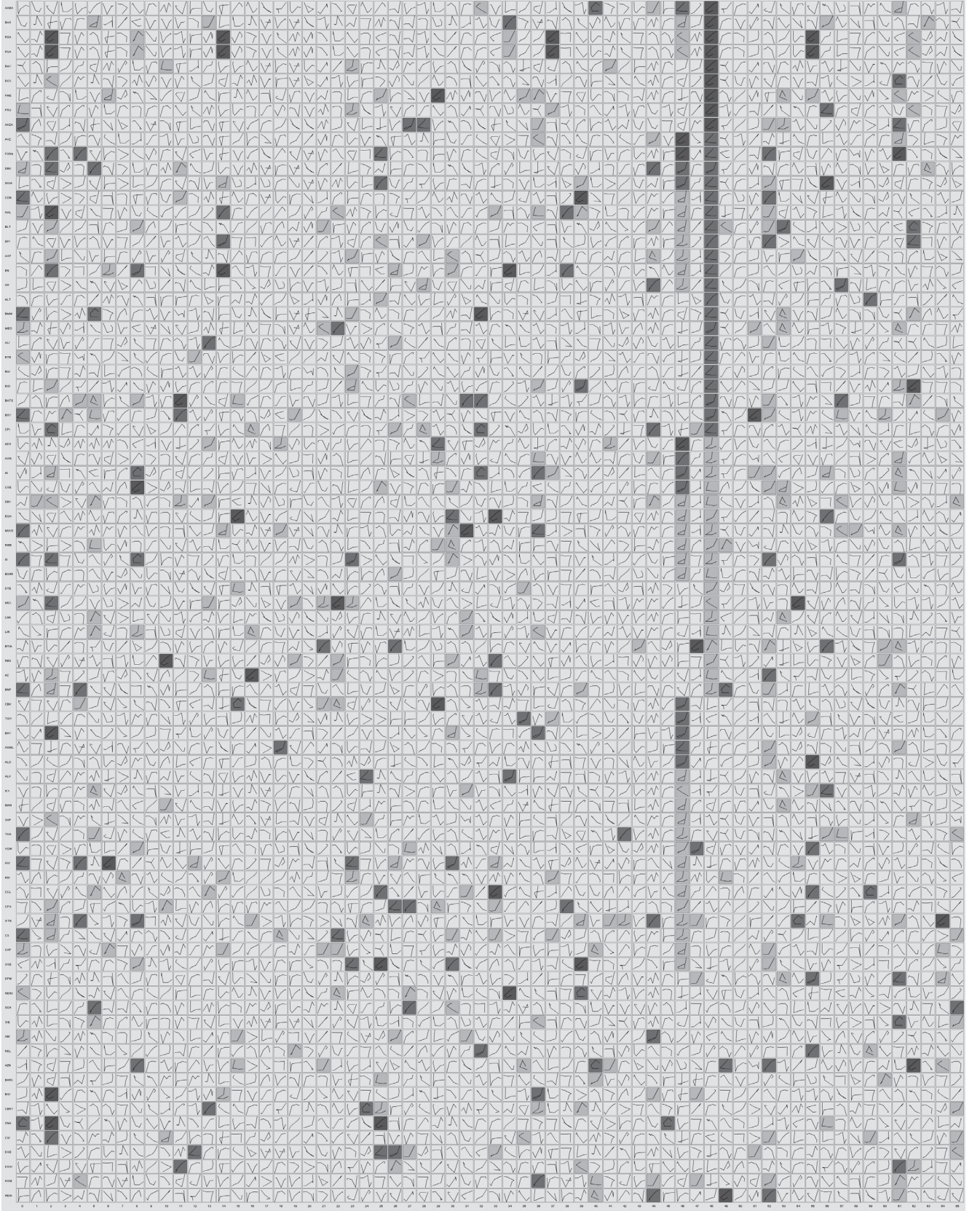


Figure 9: The sequence view visualizing chart movements of 83 assets during 66 weeks of observation. Patterns [6,6] and [0,2] as selected by the user from a set of automatically generated candidate patterns are highlighted (cf. also Figures 7 and 8).

data preprocessing and automatic data analysis are vital for constructing potentially useful views from complex data, supporting the visual analysis process. For our sequence view, we implemented a rather simple, frequency-based pattern analysis for generating candidate views, which in turn should support correlation analysis in chart movement patterns. Future work will identify and take into account more advanced analysis algorithms introduced in the time series and trajectory mining communities. Specifically, we like to incorporate additional unsupervised analysis algorithms for identification of potentially interesting patterns. A promising starting point to this end could be [31].

Closely related to the pattern identification is the definition of relevant data features. We plan to include additional data normalization and trajectory feature extraction algorithms into our system, to support a wider variety of application areas. Specifically, trajectory similarity notions relying on derived and structural properties will be implemented. In particular, regarding the financial domain, inclusion of sectoral, market volume, or other derived indicators is desirable. Finally, regarding the interaction side, we plan to extend the system by improved similarity search functionality, allowing the user to graphically issue queries for certain trajectory characteristics that are of interest in a given context.

## Acknowledgements

We thank our colleagues Christoph Hornung and Thorsten May of Fraunhofer IGD for helpful discussions. We thank the anonymous reviewers for providing valuable comments which helped to improve this work.

## 7. REFERENCES

- [1] W. Aigner, S. Miksch, W. Mueller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics*, 2008. to appear.
- [2] N. Andrienko and G. Andrienko. Designing visual analytics methods for massive collections of movement data. *Cartographica*, 42(2):117–138, 2007.
- [3] W. Cleveland and B. Kleiner. A graphical technique for enhancing scatterplots with moving statistics. *Technometrics*, 17(4):447–454, 1975.
- [4] T. Dwyer and D. Gallagher. Visualising changes in fund manager holdings in two and a half-dimensions. *Information Visualization*, 3(4):227–244, 2004.
- [5] R. Edwards and J. Magee. *Technical Analysis of Stock Trends*. Amacom, 2001.
- [6] M. Esteban and D. Morales. A summary of entropy statistics. *Kybernetika*, 31(4):337–346, 1995.
- [7] Y.-H. Fua, M. Ward, and E. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proc. IEEE Conference on Visualization (VIS)*, pages 43–50, 1999.
- [8] HEyeWall System, Fraunhofer IGD Darmstadt, Germany. <http://www.heyewall.de/>.
- [9] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.
- [10] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen. WEBSOM—self-organizing maps of document collections. In *Proc. Workshop on Self-Organizing Maps (WSOM)*, pages 310–315. Helsinki University of Technology, 1997.
- [11] Y. Ivanov, C. Wren, A. Sorokin, and I. Kaur. Visualizing the history of living spaces. *Transactions on Visualization and Computer Graphics*, 13(6):1153–1160, 2007.
- [12] D. Keim, T. Nietzschmann, N. Schelwies, J. Schneidewind, T. Schreck, and H. Ziegler. A spectral visualization system for analyzing financial time series data. In *Proc. Eurographics/IEEE-VGTC Symposium on Visualization (EuroVis)*, 2006.
- [13] E. Keogh. A decade of progress in indexing and mining large time series databases. In *Int. Conference on Very Large Data Bases (VLDB)*, 2006. Tutorial.
- [14] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 3rd edition, 2001.
- [15] T. Kohonen, J. Hynninen, J. Kangas, and J. Laaksonen. Som-pak: The self-organizing map program package. Technical Report A31, Helsinki University of Technology, 1996.
- [16] J. Kráľovič and K. Vlachynský. *Financial Management*. Elita, Bratislava, 1996.
- [17] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja. PicSOM—content-based image retrieval with self-organizing maps. *Pattern Recogn. Lett.*, 21(13-14):1199–1207, 2000.
- [18] L. Lin, L. Cao, and C. Zhang. The fish-eye visualization of foreign currency exchange data streams. In *Proc. Asia-Pacific Symposium on Information Visualisation (APVis)*, pages 91–96, 2005.
- [19] C. Merino, M. Sips, C. Panse, R. Spence, and D. Keim. Task-at-hand interface for change detection in stock market data. In *Proc. ACM Advanced Visual Interfaces International Working Conference (AVI)*, 2006.
- [20] J. J. Murphy, editor. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. Prentice Hall Press, 1999.
- [21] C. North, U. Farooq, and D. Akhter. Datawear: Revealing trends of dynamic data in visualizations. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 8–11, 2001.
- [22] E. Pampalk, A. Rauber, and D. Merkl. Using smoothed data histograms for cluster visualization in self-organizing maps. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN)*, volume 2415 of *Lecture Notes in Computer Science*. Springer, 2002.
- [23] N. Pelekis, I. Kopanakis, G. Marketos, I. Ntoutsis, G. Andrienko, and Y. Theodoridis. Similarity search in trajectory databases. In *Proc. Int. Symposium on Temporal Representation and Reasoning*, 2007.
- [24] H. Piringer, R. Kosara, and H. Hauser. Interactive focus+context visualization with linked 2d/3d scatterplots. In *Proc. Int. Conf. on Coordinated and Multiple Views in Exploratory Visualization*, 2004.
- [25] PowerWall System, University of Konstanz, Germany. <http://infovis.uni-konstanz.de/powerwall>.
- [26] SmartMoney Map of the Market. <http://www.smartmoney.com/marketmap/>.
- [27] T. Taskaya and K. Ahmad. Bimodal visualisation: a financial trading case study. In *Proc. Int. Conference on Information Visualization*, pages 320–326, 2003.
- [28] T. Tekušová and J. Kohlhammer. Applying animation to the visual analysis of financial time-dependent data. In *Proc. Int. Conf. on Information Visualization (IV)*, 2007.
- [29] J. Vesanto. SOM-based data visualization methods. *Intelligent Data Analysis*, 3(2):111–126, 1999.
- [30] K. Šimunić. Visualization of stock market charts. In *Proc. Int. Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, 2003.
- [31] L. Wei, N. Kumar, V. Lolla, E. Keogh, S. Lonardi, and C. Ratanamahatana. Assumption-free anomaly detection in time series. In *Proc. Int. Conf. on Scientific and Statistical Database Management (SSDBM)*, 2005.