

Market Explorer: an exploration and visualization tool for high-frequency trading data

Frank Autenrieth*

Cristina Morariu†

Peter Ruch‡

Ivo Zeba§

Index Terms: visualization, exploration, high-frequency trading

1 INTRODUCTION

Each day high-frequency trading accounts for large amounts of the produced data. High-frequency trading contains information of the trading of the whole day, usually logs of every single trading activity, e.g. submissions or cancellations of orders, for thousands of stocks. Using this data it is even possible to reconstruct the actual order book state for every time point of the day. This data enables researchers to track the order book dynamics of a specific stock. Naturally one would expect that this type of information is relatively expensive. The computationally intensive integration of all this data and the economical advantages, the usage of the data implies, make these data very expensive. High-frequency trading data usually isn't publicly available and as of regulations relatively difficult to come by. For financial researchers there luckily already exist an option, the LOBSTER limit order book reconstruction service. After signup and verification of ones academic affiliation one can access very high quality data on the order book dynamics of a stock of ones choice.

Ask Price 1	Ask Size 1	Bid Price 1	Bid Size 1	Ask Price 2	Ask Size 2	Bid Price 2	Bid Size 2	...
1186600	9484	118500	8800	118700	22700	118400	14930	...
1186600	9384	118500	8800	118700	22700	118400	14930	...
...

Figure 1: lobster order book output structure format

Having access to this information allows researchers to come up with new hypotheses on high-frequency trading. Unfortunately exploration of these very dense datasets is relatively difficult and requires at least knowledge of an interactive programming language with visualization support. To our knowledge currently there doesn't exist a dedicated exploration and visualization framework to work with high-frequency trading data. On the following pages we will present a proposal applications which is supposed to fill this gap. We introduce market explorer(the name is a pretty bad word game on crayfish to go with the already existing LOBSTER service) an interactive visualization and data exploration framework that operates on output files from LOBSTER. market explorer is web-based and runs in every up-to-date browser.

2 RELATED WORK

While there currently exists no open available tool that offers the same functionality as our solution, there already exist a lot of standard views that focus on various order book characteristics and

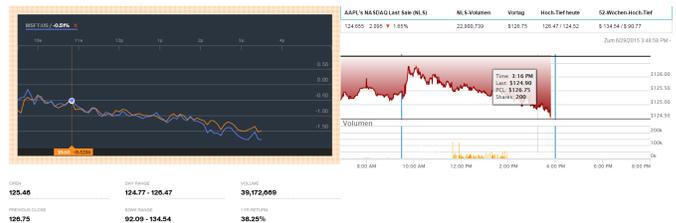
*e-mail: frank.autenrieth@web.de

†e-mail: cristina@morariu.ro

‡e-mail: rutschifengga@gmail.com

§e-mail: croivozeba@hotmail.com

statistics. Stock market related websites as Bloomberg.com or NASDAQ's own web presence offer overviews on the developments of a selected stock's price over the day. These views only account for a fraction of the relevant information. Elementary functionality like inspecting multiple characteristics at once or even **simple comparison of different stocks is often not possible**. Additionally for many sites one does not know the exact aggregation level that is used when visualizing the data.



(a) source: bloomberg

(b) source: nasdaq

Figure 2: standard views as found on finance related websites

Interestingly for many characteristics there already exist great views that manage to visualize the complex structure and dynamics of the order book. Schroeter [1] offers a good introduction into the domain of high-frequency trading, the publication additionally presents very interesting views of various limit order book components.

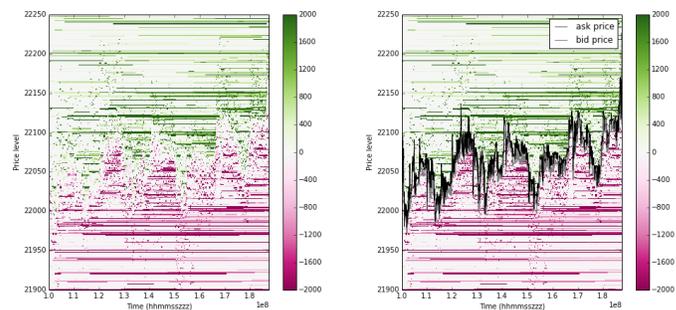


Figure 3: extended views by Schroeter

3 MARKET EXPLORER

market explorer is still in very early alpha stage and does not offer some of the planned functionality. We encountered serious difficulties when coming up with a solution which manages to live on lower screen estates. Currently we recommend the usage of high resolution monitors to account of weaknesses in the layout. For smaller resolutions the application will reorganize the views into tabs. We are aware that this is not ideal and we are currently working on various improvements of screen estate usage.

3.1 Approach

3.1.1 Single stocks

The first iterations were largely based on already available views as seen on many financial websites. We initially tried to use methods that are part of the visual vocabulary of our audience, researchers in quantitative finance. Early ideas mainly focused on the visualization of the order book. For the functionality we distinguish between the single-stock and multi-stock setting.

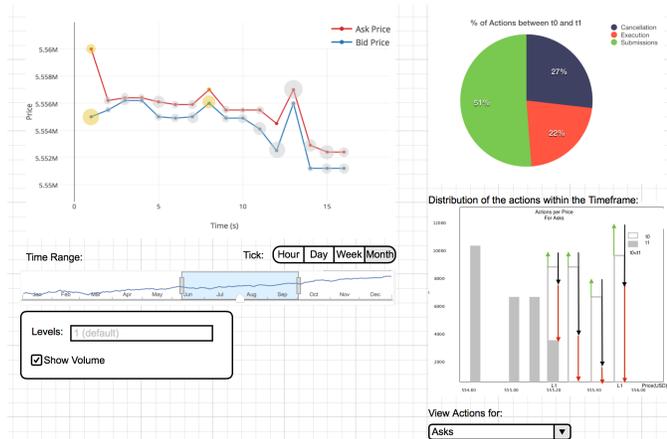


Figure 4: paper mockup

We realized relatively quickly that these standard views convey way to little of the available information, completely ignoring characteristics like activity. In the further iterations we tried to resolve these issues by paying special attention to the distribution of the trading events, that the whole order book structure implicitly is already based on.

The final task was to come up with a work flow on how our tool was going to be used. market explorer’s main purpose is exploration and pattern- and structure-detection in high frequency trading data. Despite humans being relatively skilled at these tasks on their own, the very large and dense information calls for computational support for highlighting interesting areas and filter for relevant information. For market explorer we added the option to specify filters to aid in the detection of what we call notable events (e.g. drops of price or volume). market explorer offers the possibility to highlight these events and select a prior time window that then can be compared to the general behaviour in this interval or another notable event’s prior window. Assessment of similarity is currently purely visual but it might be interesting to include specific dissimilarity-measures, such as dynamic-time-warp (DTW) or correlation (pearson correlation), that are usually used with time-series data.

3.1.2 Multiple stocks

The multi-stock approach is essentially different from the standard case. Handling of multiple stocks will pose serious requirements on the applications architecture and layout. We did not spend a lot of thoughts on this problem yet. **Some major issues we could think of are computational cost, layout and responsiveness of the user interface when working on multiple stocks from several days, possibly even weeks.** Selection and grouping of the stocks will probably an additional problem, although we might be able to sufficiently address this issue using a multi-dimensional-scaling (MDS) approach.

3.2 Implementation

market explorer runs in a virtual machine that’s provisioned using Vagrant. The frontend is a mix of html5 and javascript, the backend

The Stock Avatar

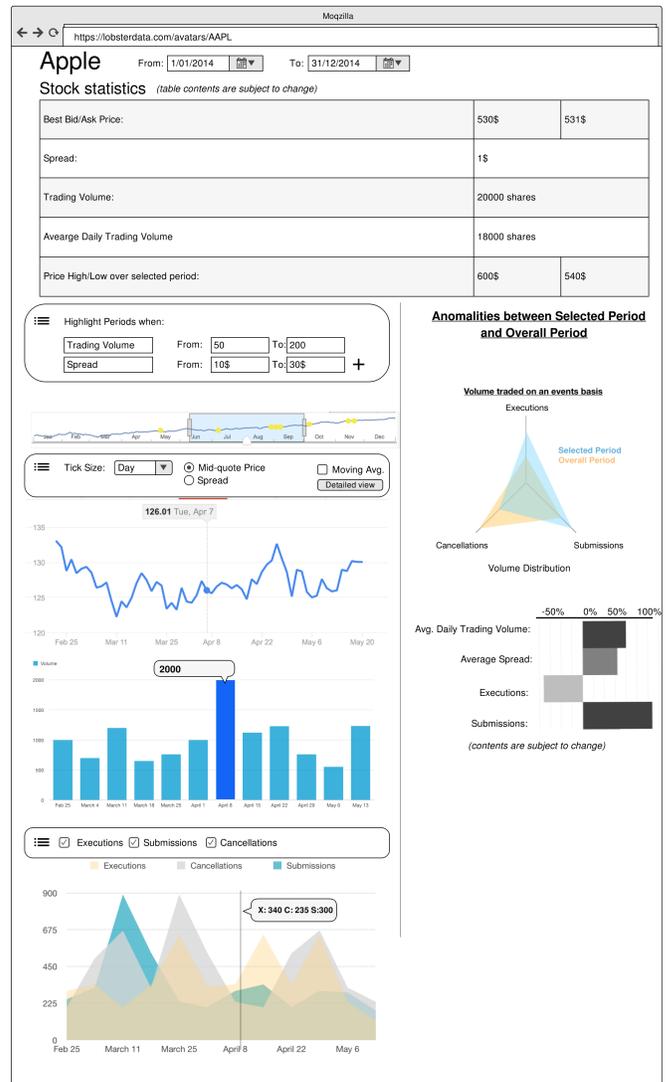


Figure 5: hifi prototype - first avatar

consists of a mysql server, that’s communicating with the frontend using several php-scripts. Initially it was planned that the backend would be entirely written in R. The initial idea was to use R’s excellent capabilities for data-wrangling and to utilize the wide range of available R-packages to easily extend the application’s functionality in the future. Following this directive we implemented an own R-package called "lobstR" that would act as an API to facilitate the R javascript interaction. Unfortunately interfacing R with javascript was not that easy as expected. We finally chose the openCPU framework that allows us to call R from the web and also offers an own javascript API. openCPU is relatively powerful but also has its caveats, as the final application has to be deployed as an own R-package. This is not necessarily a bad thing as we would have been able to use the CRAN network for easy deployment of our application, for non R users the whole setup nevertheless was relatively cumbersome and cause for a lot of problems. Unfortunately the whole R related work also consumed an enormous amount of time. In the meantime some team members already had started with the implementation of the front end using a mysql database as backend.

The Stock Avatar

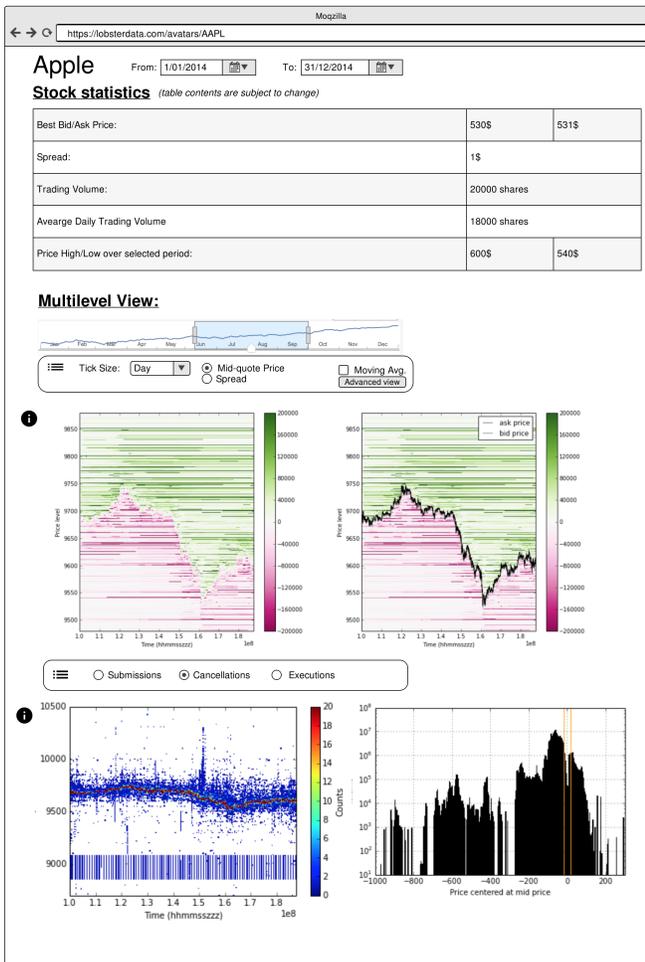


Figure 6: hifi prototype - second avatar

After that point the interest in a R-based backend quickly faded and the idea was dropped. Nevertheless the complete R backend is. The current application uses a combination of client-server, model-view-controller, and pipes and filters architectures. The back-end currently only holds a pre-defined stock for one day, if another stock or another day is available it can be easily imported using the `import.php` script in the `init` directory of the project. This can easily be modified to accept multiple stocks and change between them through simple modifications of queries. Communication between the back-end PHP-MySQL server and the front-end html-javascript client is performed using AJAX and uses the JSON format. The back-end takes care of filtering out data to a specific percentile to provide a very accurate look of the data. The front-end is comprised of its own MVC application which uses re-usable directives and creates multiple instances of them using the different data. Essentially when the user starts the front-end application and selects a tick size, an AJAX call is performed to the back-end server requesting data for the general overview and heatmaps. Since there are multiple heatmaps, a new heatmap is created using different data. When performing filtering or selecting a time frame to compare to the overall day, another AJAX request is performed which only asks the back-end for the data requested. Once again, each of the bar charts used for comparison is its own instance with its own data set.

3.3 Results

3.3.1 Use cases

As market explorer functionality is currently limited to the single stock case and suffers from abovementioned layout problems. The application is comprised of two avatars that are accessible organized in tabbed views. Once starting up the application the user is greeted by the tick-size input mask where he can specify the resolution of the data aggregation (one tick corresponds to a second trading time). At startup the first avatar is focused and the user is presented with an overview over the trading over the whole day. market explorer highlights events in the orderbook based on predefined filters using small orange circles. The tool already comes with a specific set of filters enabled so users are instantly ready to go and explore the specific stock.

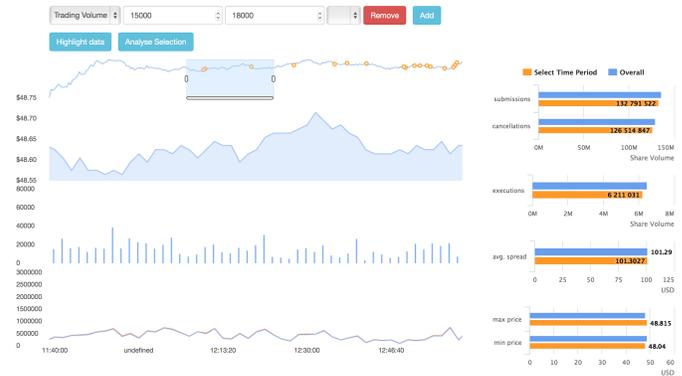


Figure 7: current implementation - first avatar

The first avatar is comprised of views for the midpoint price, trading volume and number of executions for each tick in the selected trading interval. market explorer allows the user to specify own intervals whose summary statistics will be presented on the right side of the avatar. In these graphics the interval's statistics are compared to the global trends. The initial hypotheses is that the time before previous detected relevant events should show some special behaviour, that might be visible by inspecting the respective summary statistics. Initially it was planned to also be able to compare intervals themselves. Due to time restraints this feature was omitted in the current iteration.

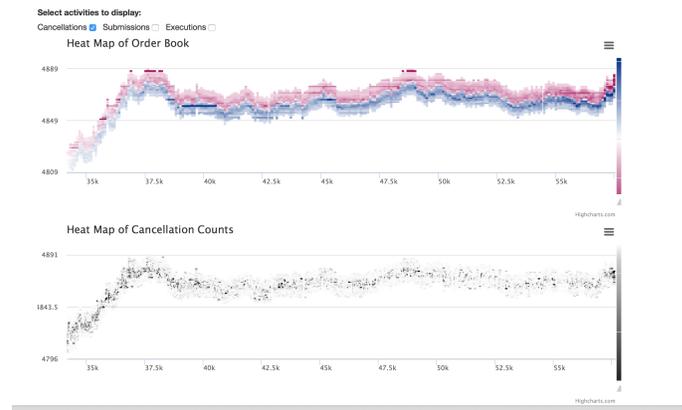


Figure 8: current implementation - second avatar

The second avatar represents the stock's orderbook states and trading activity using a heatmap. Utilizing this view one can easily get a deeper understanding of the general dynamics within the

dataset. By default only the heatmap for the orderbook is shown, additional views are available by clicking the respective checkboxes.

3.3.2 Performance

Working with very large datasets performance is always an issue. We use non-svg javascript plotting libraries for improved response times when redrawing (e.g. when the user is zooming in at a specific region). These libraries use canvas elements and are only drawn upon user request to reduce the chance of wasting resources. Frequently used data is also cached by the MySQL database into memory, therefore significantly reducing load times after the first request. Temporary tables are also used during computations to save results for later re-use. These tables are dropped when the connection is closed.

3.3.3 Feedback

During evaluation of our current application we got a lot of positive feedback. Most of the people we asked instantly understood the user interface. Still the evaluation phase helped us to detect a lot of minor usability issues that we continuously resolved. Interestingly people in the survey rarely criticized our use of tabs. While we are aware that the usage of tabs is not ideal a lot of people seemed not be bothered by them. Some might now say that these findings don't support a complete revision of the layout. Nevertheless we have a different explanation for this phenomenon: The whole evaluation phase was simply to short to get any meaningful results that are generalizable to the whole lifetime of our application. Big design issues might only unfold after a specific time of use. The whole problem can be interpreted to be analogous to classic complexity theory as encountered in computer science. An additional "unnecessary" loop might significantly increase the complexity of an algorithm, similarly the usage of tabs significantly worsens the usability of an application and usually result in a lot of wasted time.

4 DISCUSSION

One of the main weaknesses is that market explorer is still in very early alpha, hence there are still a lot of details that need attention. One of the main issues is the layout. Currently the tabbed view is not ideal, but the current size of the plots and number of statistics make it very hard, to organize everything in one single view. From this point we will probably still need several iterations to arrive at a usable design. Another major caveat of the current implementation is that we only focus on a selection of possible statistics. What does that mean? Time series analysis offers a rich theory for analysing sequences of time dependent measurements. Many different methods exist that can roughly be divided into time- and frequency-domain methods. In our application we only scratch the time-domain methods, by looking at the different time windows and completely ignore the frequency based approaches (e.g. fourier analysis). Neglecting these methods means that we somewhat neglect the statistical opportunities that the wide range of different models offer (e.g. the chance to perform inference on specific time series and utilize the complete theory that's already available). This very basic view is not completely our fault. During the iterative refinement with our advisor from the financial sector we never got a real picture of what he wanted to achieve with the data. We actually came up with a lot of the now offered functionality. At the beginning our understanding of the data and the problem as well as the theoretical background was very low. At this point a lot of missed opportunities start to unfold that could have easily been exploited at an earlier stage of development, given that one would also consider the actual workflows and models used in this domain. Considering these not-given factors the current state of the project is really good and it was an actually a really nice and interesting problem to work on. Considering our current workflow the tool does a really good

job covering these questions, but if we need to extend it and add functionality a lot more work will be necessary.

ACKNOWLEDGEMENTS

The authors wish to thank Goekhan Cebiroglu for his support and input related to the high-frequency data and Thomas Torsney-Weir for his comments on several of our early visualization designs.

REFERENCES

- [1] J. Schroeter. Limit order book reconstruction, visualization and statistical analysis of the order flow. Master's thesis, ETH Zurich, 2014.