

Innovative Filtering Techniques and Customized Analytics Tools

VAST 2009 Flitter Mini Challenge Award: Innovative analytic tool
VAST 2009 Video Mini Challenge Award: Outstanding video analysis tool
VAST 2009 Grand Challenge Award: Excellent example of analytic tradecraft

Harald Bosch

Julian Heinrich
Christoph Müller

Benjamin Höferlin
Guido Reina

Markus Höferlin
Michael Wörner

Steffen Koch

Visualization and Interactive Systems Institute and Visualization Research Center Universität Stuttgart*

ABSTRACT

The VAST 2009 Challenge consisted of three heterogeneous synthetic data sets organized into separate mini-challenges with minimal correspondence information. The challenge task was the identification of a suspected data theft from cyber and real-world traces. The grand challenge required integrating the findings from the mini challenges into a plausible, consistent scenario. A mixture of linked, customized tools based on queryable models and rapid prototyping as well as generic analysis tools (developed in-house) helped us correctly solve all of the mini challenges. A collaborative analytic process was employed to reconstruct the scenario and to propose the correct steps for the reliable identification of the criminal organization based on activity traces of its members.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—GUI; I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—Video Analysis

1 INTRODUCTION

Two aspects of visual analytics are of special importance in this context. The first one is to show large amounts of data in a way that analysts can see patterns and outliers, allowing the deduction of hypotheses. The second one is to enable the analyst to create views that prove or disprove these hypotheses.

Both aspects require the adaptation of the visualization for the specific scenario, either through run-time customization or source-code modification. This is especially necessary for tasks such as filtering, highlighting, and cross-referencing the data visually according to the scenario requirements. Additionally, the data itself needs to be adapted and enriched to allow for customizing the views. This can be achieved by (semi-)automatic data preprocessing. For the VAST 2009 Challenge we created and adapted tools featuring both aspects with respect to the provided data.

2 ANALYSIS PROCEDURE AND TOOL SET

Solving the Grand Challenge required integrating the findings from the mini challenges (MCs). As there was no data to link the MC data sets, such as a mapping of badge IDs to employee names, data integration was not possible. The only correspondence was temporal and valid only for traffic logs and video footage.

Therefore, we formulated hypotheses from the findings of different MCs. Together, we tried to prove or disprove these hypotheses by cross-checking their validity in each of the tool sets. When multiple valid scenarios remained, we included all of the options in the analytic debrief. For instance, from the video analysis we knew

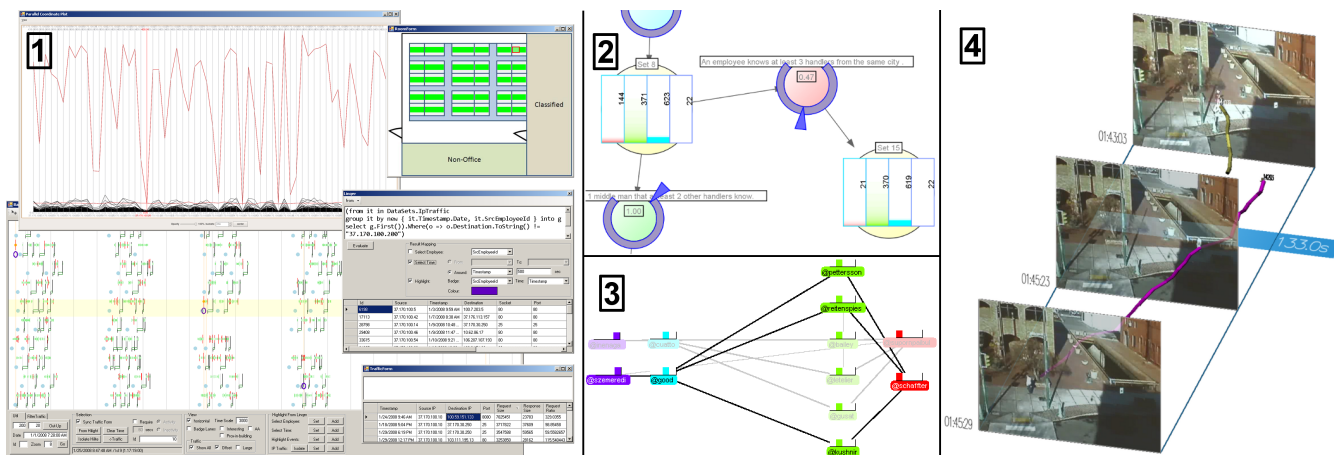
that there was a woman meeting at least two men who act suspicious. The social network analysis provides information about who in the criminal network knows at least two other persons located in their vicinity. We inferred plausible mappings from these persons to their role in the organization and formulated ways to confirm these hypotheses, e.g. by checking the embassy roster. The following sections describe our MC tool sets to discover and relate findings.

MC1 - Badge and Network Traffic The starting point of identifying suspicious network traffic was the formalization of suspicious transactions. We started straightforward with the ratio between request and response sizes per destination. Exploring this ratio in a source/destination matrix in parallel coordinates using SpRay [2], we identified a prominent outlier, which we asserted to be the mail server based on IP address and port used. Filtering this traffic suggested the actual solution as the next suspicious ratio.

Based on the badge usage rules for the confidential room, we made two additional assumptions: violations thereof (piggy backing) are suspicious, and persons who are badged into the room are very likely to really be there thus being unable to use their computers. Using C# for rapid prototyping, we built a tool that lets us check for any badge inconsistencies programmatically and visually correlate badge and network traffic via stacked time series (cf. Fig. 1). We map the input records to objects and each column to a property. Aggregating these objects into lists implicitly provides dynamic querying capabilities through LINQ. Our tool can transform arbitrary LINQ queries dynamically into highlighting and filtering at runtime. This made it easy to hide all users that exhibit IP activity or are in the confidential room within a user-defined time span around the suspicious uploads—leaving exactly one suspect. Brushing and linking between all four windows allows for the verification of hypotheses and for formulating of new ones. However, none of the complex theories that we developed held out against this step: for instance, one destination address has an exceptionally regular pattern in parallel coordinates. Highlighting the traffic in the time line display reveals that this address is contacted by every employee only once at the start of his working day. Formulating an exact query proves that this is always the first legitimate network traffic of the day, unless the user had entered the confidential room before. We assumed some kind of unsuspecting bulletin server, whose content is redundantly available in the confidential room.

MC2 - Social Network and Geospatial The natural language description defining the properties of a hidden criminal network in the social network data was rather vague, e.g. “handlers probably have between 30 to 40 contacts”. Therefore, we created a set of fuzzy rules that assign a confidence value to each *Flitter account/scenario role* combination. A confidence value of 1 denotes accounts that match a rule precisely (e.g. between 30 and 40). For accounts within a rule-specific falloff interval (e.g. 29 to 10 and 41 to 60), the confidence value is gradually reduced to 0 using a user-defined interpolation function.

*<http://www.vis.uni-stuttgart.de>



Applying the sequence of rules that describe a scenario, we calculate the confidence value for every account/role pair as the product of the confidence values resulting from the rules defining the role. Since rules may be interdependent, we iterate this calculation several times in a background thread. By excluding all combinations with a confidence value below a certain threshold (we used 0.5), we derive a set of candidates for each role.

Instead of using a static definition of the scenario, we created a hypothesis graph view (cf. Fig. 2) to interactively apply rules that successively reduce the number of candidates for each role. Every state node in the graph depicts a state in the analysis, containing one candidate set for each role. Initially, every candidate set contains all Flitter users. By attaching rule nodes to state nodes via drag&drop interaction, we derive hypotheses, thereby continuously reducing the number of candidates. State nodes contain bars showing the confidence distribution of each role and the number of remaining candidates. Thus, it becomes instantly apparent which rules greatly influence the result sets and which have limited effect. In addition to using fuzzy rules, analysts can express their confidence in a rule by specifying a weight value on a circular slider around a rule node. It is possible to branch this graph to model different scenarios or to go back to earlier hypotheses if a rule application resulted in an empty candidate set, keeping those “dead-end” hypotheses visible. As a side effect, the resulting graph provides provenance information about the analysis process. Selecting a node enables additional views on the corresponding result set.

Fig. 3 shows a network view of the remaining role candidates and the potential criminal networks for visual evaluation. Nodes are grouped into layers according to their possible roles. Four bars on every node depict how well the candidate satisfies the role description. Candidates for more than one role are represented by multiple nodes. This is also shown by mixed role colors and multiple confidence bars at these nodes. Hovering over an entity highlights possible criminal networks by emphasizing the links between adjacent layers recursively. The analyst can discard networks identified as incomplete or invalid without leaving the view by using a context menu. This information is fed back into the analysis and updates the view accordingly. The map view shows the whereabouts of the identified network’s members along with the relevant links between cities. Some of the views for this MC have adapted from [3].

MC3 - Video Analysis The basic idea of our approach is to identify the encounter of people by their movement trajectories. The characteristics of these trajectories should help us localize relevant parts of the video and thus yield a scalable method.

First, we separate the video sequence by an automated analysis step into the four locations which were captured by the camera. Then we extract the trajectories using a combined approach of

optical flow computation and background subtraction. For optical flow computation we use the popular pyramidal Lucas-Kanade approach. Object tracking is done by a linear Kalman filter in screen space. Finally, object properties are calculated based on manual camera calibration. This information includes camera location, the trajectories’ temporal start and end positions and their spatial positions in pixel coordinates. The perspective-corrected positions as well as the mean speed and average direction are also computed.

The visualization applies the VideoPerpetuoGram methodology [1] to get a summarization of the actions in the video. We combine keyframes at a sparse interval of frames with the trajectories of moving objects (cf. Fig. 4). This makes it possible to keep track of these movements even while playing the video in fast-forward mode. For scalability reasons we omit scenes that are of minor importance for the analyst. The importance of trajectories is defined by a real-time filter framework. Filters can be specified in an intuitive manner based on the above mentioned properties calculated in the automated analysis step. Especially the possibility to filter trajectories by their relationship enables users to detect splits and merges. This iterative refinement finally results in a manageable amount of trajectories for further investigation.

3 CONCLUSION AND FUTURE WORK

We demonstrated that our tool set can be used to analyze the given test scenario successfully. Some of the applied tools can be generalized and adapted to other analytic problems. One focus of research will be the development of interactive visual filter metaphors in order to simplify the query mechanisms of different tool sets. Furthermore, we will improve the tool for hypothesis building with respect to screen space, clarity, and new features for analysis provenance.

ACKNOWLEDGEMENTS

We thank the VAST Challenge 2009 team, all reviewers, and our supervisors Thomas Ertl, Gunther Heidemann, and Daniel Weiskopf. This work was partially funded by the German Science Foundation (DFG) and the state of Baden-Württemberg as part of SPP 1335, BW-FIT, and GSaME.

REFERENCES

- [1] R. P. Botchen, S. Bachthaler, F. Schick, M. Chen, G. Mori, D. Weiskopf, and T. Ertl. Action-based multifield video visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):885–899, 2008.
- [2] J. Dietzsch, J. Heinrich, K. Nieselt, and D. Bartz. Spray: A visual analytics approach for gene expression data. In *IEEE Symposium on Visual Analytics Science and Technology (to appear)*, 2009.
- [3] S. Koch, H. Bosch, M. Giereth, and T. Ertl. Iterative integration of visual insights during patent search and analysis. In *IEEE Symposium on Visual Analytics Science and Technology (to appear)*, 2009.