RISKWHEEL: INTERACTIVE VISUAL ANALYTICS FOR SURVEILLANCE EVENT DETECTION

Yu Cheng, L. Brown, Q. Fan, R. Feris, S. Pankanti

Exploratory Computer Vision Group
IBM T.J. Watson Research Center
{yucheng, lisabr, rsferis, qfan, sharat}@us.ibm.com

Tao Zhang

Department of Automation
Tsinghua University
taozhang@mail.tsinghua.edu.cn

ABSTRACT

Detecting human behaviors in vast amounts of video is a challenging task in a variety of real-world applications. Thus an interactive tool designed to support this task with human in the loop is of significance in various domains including public safety and security. In this paper, we design and develop an interactive visual analytics system, RiskWheel, that enables effective analysis of detection results and utilization of user feedback to improve surveillance event detection. In particular, we propose 1) an interactive approach to visualize data with temporal relations and 2) a novel risk ranking method to differentiate detection results and present more informative ones to the user for better interaction. In our experiments, we demonstrate RiskWheel through a case study on TRECVID Surveillance Event Detection (SED) task [1]. The experimental results quantitatively show that RiskWheel outperforms multiple baselines, demonstrating the power of the risk ranking technique.

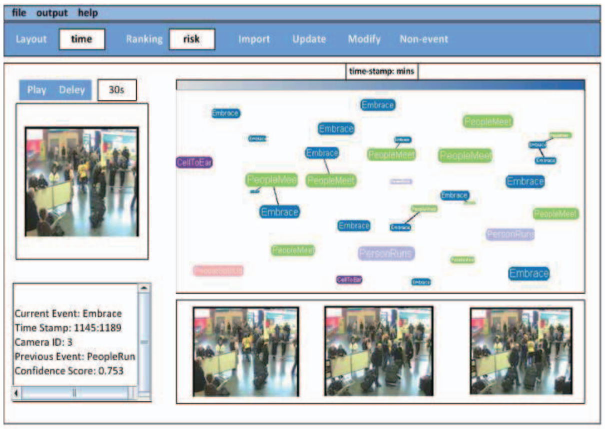
Index Terms— Surveillance Event Detection, Interactive Visualization, Risk Ranking, Temporal Modeling

1. INTRODUCTION

Surveillance event detection (SED) addresses the need for automatic detection of events in large amounts of surveillance videos. It is a fundamental problem for a variety of high-level applications of critical importance to public safety and security. Generally speaking, the task is to identify the temporal range of a specific event such as person running when it occurs in a video. While there have been increasing efforts recently to tackle this problem, it remains a challenging task [1], due to many confounding issues such as cluttered background, occlusions, and viewpoint changes.

Recently interactive and relevance-feedback techniques have attracted increasing research interest and effectively employed in many applications such as image retrieval and annotation. These techniques aim at exploiting human efforts to further push classification/detection/search performance with interactive learning. Fogarty *et al* [2] developed a novel system CueFlik to web image search based on interactive concept learning. The system allows the end-user to quickly create rules for re-ranking images according to their visual characteristics. In [3], Gosselin *et al.* addressed the problem of image retrieval, and proposed a scheme to combine different active learning strategies to select only a few query samples. Yao *et al.* [4] proposed an interactive object annotation method that incrementally trains an object detector with a focus on minimizing human annotation time rather than pure algorithm learning performance. In addition, a few works have focused on making the annotation process itself more efficiently for the human annotator. In [5], the authors proposed a method to categorize images using binary queries, thus eliminating the need for the annotator to to select a predefined category. In the system of [6], video annotations are initially derived from tracking results, and active learning is used to intelligently query the human annotator for corrections on the tracks. In the domain of video surveillance, there have existed some interactive systems for SED such as [7, 8]. However, most of these systems do not involve interactive learning, and only adopt a linear presentation of events. This greatly limits the user's capability of exploring the interdependencies among events. Thus, there is a significant need of developing effective visual tools that reveal analysis results in a more intuitive way for SED.

In this paper, we propose RiskWheel (Figure 1), a novel interactive visual analytics system, to improve the performance of SED with optimal feedback from human users. RiskWheel is designed as a platform to provide effective interactions for the user and support interactive annotation (collecting training samples) and re-annotation. To make the most utilization of interaction available in a limited time, the system design was driven by considerations from two perspectives: 1) efficient visualization of intermediate detection results for user interaction; and 2) effective utilization of user feedback for performance boost. Our first major contribution is a novel visual design that presents the intermediate detections with temporality. We capture successive events from a sequence of detections and represent them as streaming belts, i.e. groups of events with temporal patterns. A series of layouts corresponding to the evolution of detections



Current Event: Embrace
Time Stamp: 1145:1189
Camera ID: 3
Previous Event: PeopleRun
Confidence Score: 0.753

Fig. 1. The interface of RiskWheel, an interactive system for surveillance event detection. In each theme, the bars over time line show how likely an event occurs at the time. The line between two detection nodes represents temporal relations and the size of the node indicates the risk of a detection.

are designed to display these connected streaming belts, providing an overview of event context. Compared to the linear presentation used by the currently existing systems such as [7, 8], such a two-dimensional representation helps to facilitate data understanding and better portrays rich interactions that enables explorative analysis. Our second contribution is the development of a method based on risk ranking to present detection results effectively to the end user for analysis. Here the risk of a detection indicates the potential value or impact that the detection has on the system performance upon analysis by the user. We propose a novel way to measure the risk of a detection by combining several factors into an overall score, including the margin of top two candidate events for the detection, temporal relations between events and potential annotation costs.

Evaluation of the RiskWheel system was conducted on the above two aspects using the challenging TRECVID SED data set [1]. We first compared the risk ranking approach with several other ranking methods. We then further validated the effectiveness of our interactive system in a user study of interactive re-annotation. Our approach has demonstrated its superiority over others in both experiments.

2. RISKWHEEL DESIGN

RiskWheel as an interactive visual analysis system contains two fundamental functions: the temporal-driven risk analysis that identifies potential risks of the intermediate detection results and the interactive visualization that helps users to better understand contextual relationships between events. The visual analysis in the RiskWheel system contains three major steps, as illustrated in Figure 2. First, when the results of an automated event detection system are imported into RiskWheel for interactive analysis, RiskWheel explores temporal relationships among events and rank sorts detections by their risk scores (see Section 3). The temporal and ranking information are then transformed into a visual representation with rich contexts for visualization. In the meanwhile interactions are provided for the users to explore the data. Finally, user feedback is utilized to update risk ranking dynamically as the interaction proceeds.

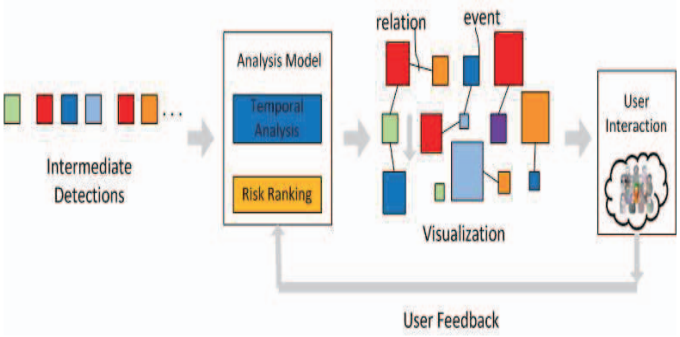


Fig. 2. The framework of RishWheel: 1) temporal and risk analysis, 2) interactive visualization, 3) user feedback.

On the visual presentation part, *streaming belt* and *event visualization* are two of the most important components. Below we describe more of them in detail.

Streaming Belt A streaming belt represents a cluster comprising a group of detections with contextual relations over time. In our design, we encode the temporal information on event detections to enhance visual pattern discovery. Examples of streaming belts are illustrated in Figure 1. For example, *PeopleMeet* (in green) and *Embrace* (in blue) are connected with an edge, indicating the two detections occur successively. Note that temporal relations of events are rich in surveillance data, as illustrated in Figure 3, thus our representation focusing on exploring temporal relations can enable the user for more effective interactive analysis. The size of a node is decided by its risk score. A larger size indicates a higher risk score. In the time-stamp based view, areas with dense streaming belts present more detections over time. In addition, we provide detection metadata information for the users to browse.

Event Visualization We define a visualization unit as a single detection or a streaming belt. To help the users more efficiently view video content, RiskWheel has an embedded player, which, once clicked, repeatedly plays back the current visualization unit at twice of the original frame rate. In the meanwhile, multiple keyframes are also extracted from the current visualization unit and displayed at the bottom of the visualization tool for fast browsing. Video playback is usually beneficial in the analysis of streaming belts which have richer contextual information. For instance, it is easier to check two detections *PeopleMeet* and *PeopleSplitUp* using video playback. On the other hand, keyframes are favored by those detections that can be verified at a quick glance of a few representative images.

3. RISK RANKING

Risk analysis is the key module of RiskWheel, which we will detail below. We define the *risk* for a detection as the potential value or impact that the detection has on the system performance upon analysis by the user. Our approach computes the risk score of a detection by considering several factors such as detection margins, event relations and annotation costs.

3.1. System Detection

We first introduce some notations and definitions that we will use throughout the rest of the paper. Assume we have a video $\mathbf{X}$, the goal is to segment $\mathbf{X}$ into a sequence of event segments $\{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$, and each segment $\mathbf{x_i}$ is assigned with a label $y_i \in \mathbf{E} = \{e_1, e_2, \cdots, e_m\}$. Here $\mathbf{E}$ is a set of classes including all pre-defined events and a *null* class representing clutter background (i.e. no event of interest). There are many systems such as [7, 9] can provide such a function for video event segment and classification. In our work, we exploit a system proposed in [10], which performs video segmentation and classification jointly with a temporal model. For each segmentation $\mathbf{x_i}$, the probability $p(y_i|\mathbf{x_i})$ is computed with y_i over all the events $\mathbf{E}$ and the first probable class label is output as the final detection label.

3.2. Risk Analysis

Let $y_i^1 (y_i^1 = y_i)$ and y_i^2 be the first and second most probable class labels for segment $\mathbf{x_i}$ and their classification scores be $p(y_i^1|\mathbf{x_i})$ and $p(y_i^2|\mathbf{x_i})$, respectively. Similar to [11, 12], we define a margin of $\mathbf{x_i}$ as,

$$\mathbf{M}(\mathbf{x_i}) = p(y_i^1|\mathbf{x_i}, \mathbf{y}_{1:i-1}) - p(y_i^2|\mathbf{x_i}, \mathbf{y}_{1:i-1}) \tag{1}$$

where $\mathbf{y}_{1:i-1} = \{y_1, \cdots, y_{i-1}\}$ are all the event labels observed prior to y_i.

By using the Bayesian rule and putting temporal sequence into consideration, the margin $\mathbf{M}(\mathbf{x_i})$ can be further expressed by,

$$\mathbf{M}(\mathbf{x_i}) = \mathrm{p}(\mathrm{y_i^1}|\mathbf{x_i}) * \mathrm{p}(\mathrm{y_i^1}|\mathbf{y}_{1:\mathrm{i}-1}) - \mathrm{p}(|\mathrm{y_i^2}|\mathbf{x_i}) * \mathrm{p}(\mathrm{y_i^2}|\mathbf{y}_{1:\mathrm{i}-1}) \tag{2}$$

where $p(y_i|\mathbf{y}_{1:i-1})$ is the probability of predicting y_i as the next label after seeing all previous labels. We defer how $p(y_i|\mathbf{y}_{1:i-1})$ is computed to Section 3.3. Intuitively, detections with large margins are easy, since the classifier has little doubt in differentiating between the two most likely class labels. Detections with small margins are more ambiguous, thus knowing the true labels would help the model discriminate more effectively between them [13].

Re-annotation provides the capability to correct wrong detections, but the potential gain of doing so is different depending on what's corrected. For example, when correcting an event label to *null*, one just removes a false alarm. However, when changing a wrong event label to a true event, one makes contribution to both precision and recall. Additionally, the time consumed on re-annotation should also be considered in the cost. Thus, in this work we model the re-annotation cost by:

$$\mathbf{C}(\mathbf{x_i}) = \frac{1}{||\mathbf{x_i}||} \cdot \begin{cases} w_m & \mathit{null} \rightarrow \mathit{true\,event} \\ w_f & \mathit{wrong\,event} \rightarrow \mathit{null} \\ w_m + w_f & \mathit{wrong\,event} \rightarrow \mathit{true\,event} \end{cases} \tag{3}$$

where $||\mathbf{x_i}||$ is the length of $\mathbf{x_i}$, w_m is the gain weight of identifying a missed event and w_f is the gain of removing a false alarm. Generally speaking, w_m and w_f can be set based on the tradeoff between false alarms and detection misses in an application (See Section 4.2 for an example). The three types of annotations in the equation are presumably predicted according to the top two most probable class labels (i.e. y_i^1 and y_i^2).

Finally the risk score of detection $\mathbf{x_i}$ is defined as:

$$\mathbf{R}(\mathbf{x_i}) = (\mathrm{Const.} - \mathbf{M}(\mathbf{x_i})) \times \mathbf{C}(\mathbf{x_i}) \tag{4}$$

where $\mathrm{Const.}$ is a constant set to be 1.0 for scaling.

This formulation can also be adapted to the pairwise and triplet cases. For example, for the risk score of a pair of temporally related events, a similar form of Eq. 4 can be deduced as:

$$\begin{aligned} \mathbf{M}(\mathbf{x_i}, \mathbf{x_{i+1}}) = \\ (p(\mathbf{x_i}|y_i^1) + p(\mathbf{x_{i+1}})|y_{i+1}^1) * p(y_i^1, y_{i+1}^1|\mathbf{y}_{1:i-1}) - \\ (p(\mathbf{x_i}|y_i^2) + p(\mathbf{x_{i+1}}|y_{i+1}^2) * p(y_i^2, y_{i+1}^2|\mathbf{y}_{1:i-1}) \\ \mathbf{C}(\mathbf{x_i}, \mathbf{x_{i+1}}) = \mathbf{C}(\mathbf{x_i}) + \mathbf{C}(\mathbf{x_{i+1}}) \end{aligned} \tag{5}$$

Theoretically, we could compute the risk score for any given number of sequential detections. However, in reality, the most common patterns are detections with single, paired and triple events. In this paper, we only compute these patterns and rank them from high to low with the same scale.

3.3. Temporal Modeling by Sequence Memoizer

To compute a risk score, it's necessary to know $p(y_i|\mathbf{y}_{1:i-1})$, i.e. the probability of a detection label y_i conditioned on an observed sequence $\{y_1, \cdots, y_{i-1}\}$. We exploit the Sequence Memoizer (SM) [14, 15] for this purpose.

Sequence Memoizer is is a non-parametric Bayesian approach for modeling discrete sequence data. Compared to other techniques for sequence modeling, it is powerful in capturing long-range dependencies and power-law properties [16] both effectively and efficiently. The approach has demonstrated state-of-the-art results in language modeling and compression, and recently in visual event detection [17]. Given a sequence of discrete random variables $\mathbf{s_{1:T}} = \{s_1, s_2, \cdots, s_T\}$ of arbitrary length T, each taking values in a symbol set. SM can predict the probability of the symbol s_i given the previous context $s_{1:i-1}$.

Given any y_i, we can compute the $P(y_i|\mathbf{y}_{1:i-1})$ by SM. By taking SM and applying a chain rule, we can obtain $P(y_{i+t}, \cdots, y_i|\mathbf{y}_{1:i-1})$ by

$$P(y_{i+t}, \cdots, y_i|\mathbf{y}_{1:i-1}) \\ = \prod_{j=i-1}^{i+t-1} p(y_{j+1}|y_1, \cdots, y_j) \tag{6}$$

which completes Eq. 2 and Eq. 5. We refer the interested reader to [17] for more detail.

4. PERFORMANCES EVALUATION

To demonstrate the effectiveness of our approach, we applied RiskWheel to the interactive TRECVID SED evaluation task [1]. The goal of the task is to develop techniques with minimum human efforts for visual event detection in a large collection of streaming video data. In the 2013 task, at test time a user can take no more than 25 elapsed minutes to search for a known surveillance event in a collection of surveillance video clips. The input to an interactive system is usually a list of detected events with temporal ranges identified by some automatic approach. More details about the task can be found in [1].

The dataset used in the task was captured from 5 surveillance cameras at different locations in a busy airport. There are a total of 10 annotated surveillance events with people engaged in particular activities. Seven of them were used in the TRECVID 2013 evaluation, including *CellToEar, Embrace, ObjectPut, Pointing, PeopleMeet, PeopleSplitUp* and *PersonRuns* (Figure 4). An overview of the visualization of all these events is depicted in Figure 3, on two SED cameras. Exploring the data by time, we found some interesting evolution patterns. For example, on Camera 1, "PeopleMeet" → "PeopleSplitUp" and "PeopleMeet" → "Embrace" of Camera 2, which are reasonable for scenario on each cameras. The development set has about 100 hours of video footage with ground truth available.

We used an event detection approach proposed in [10] to generate a list of event candidates (including *null* events) for RiskWheel for re-annotation by the user. The approach of [10] combines sequence modeling with event classification, performing segmentation and detection of events jointly in a video. The output of detection is an optimal segmentation of the video with each segment labeled as either a targeted event or *null* event. We split the development data into two equal parts, half for training and half for testing. The events detected on the test set as well as their classification scores were then fed into RiskWheel for further re-annotation by human.

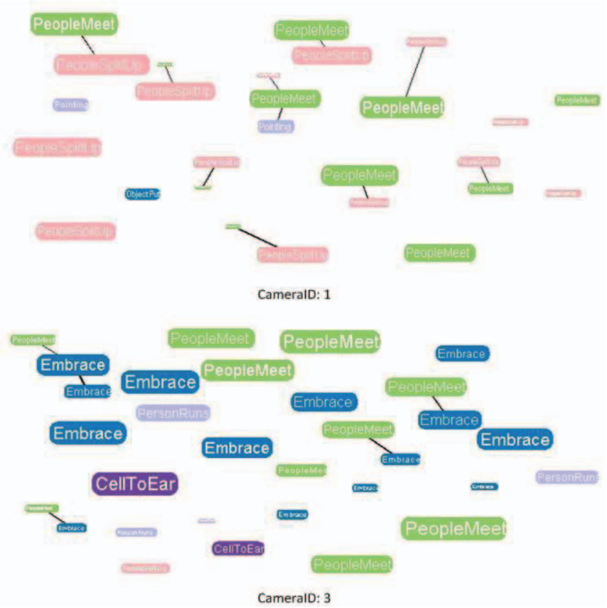


Fig. 3. Event visualization in the RiskWheel system on the TRECVID SED data set: a) Camera #1 and b) Camera #3. There are rich temporal patterns in the data for interactive exploration.

Events	Camera #1			Camera #2			Camera #3		
	R-P	R-F	NDCG	R-P	R-F	NDCG	R-P	R-F	NDCG
Confidence [10]	0.92	0.13	0.48	0.90	0.10	0.46	0.90	0.10	0.42
Margin [10]	0.96	0.08	0.67	0.92	0.06	0.63	0.94	0.06	0.55
Entropy [3]	0.98	0.08	0.74	0.94	0.05	0.71	0.95	0.06	0.59
Risk (proposed)	0.99	0.06	0.80	0.97	0.04	0.76	0.97	0.02	0.62

Table 1. Ranking performance of different approaches with metrics R-precision (R-P), R-Fall-Out (R-F) and NDCG.

4.1. Evaluation of Risk Ranking

Before conducting a formal user study with RishWheel, we first considered risk analysis proposed in this paper as a general ranking practice. While the focus of risk analysis is on confusing detections, an effective approach of such kind tends to yield better detection performance as well. We evaluated our risk analysis against 4 other approaches, including 1) confidence ranking based on the classification scores of events [10]; 2) margin ranking based on the margin score of the first and second most probable class labels [10]; and 3) entropy ranking based on the entropy of each detection [3].

In evaluation, we need to know whether or not a detection is correct. For this purpose, the TRECVID evaluation uses a Bipartite matching method [18] to align all detections with the ground truth (i.e the reference annotations). If a detection is matched to a true event, it is considered a correct detection (i.e. true positive). otherwise it is a false detection (i.e. false positive). We applied the same method here in this work.

The ranking effectiveness is then measured by R-Precision, R-Fall-Out and NDCG (Normalized Discounted

Fig. 4. Events annotated in the SED dataset: CellToEar, Embrace, ObjectPut, Pointing, PeopleMeet, PeopleSplitUp and PersonRuns.

Cumulative Gain) [19, 20]. NDCG is calculated by considering different weights of the ranked detections, i.e., the weight of a true event is higher than that of a null event. Summarized in Table 1 are the ranking results of all the approaches in comparison from 3 cameras in our data. In general, margin ranking performs slightly better than entropy ranking, and both of them are better than the confidence-based method. On the other hand, our proposed risk ranking demonstrates the best ranking quality consistently in all three ranking metrics. Especially with NDCG where the quality of events are considered, risk ranking achieves significantly better performance than the others, suggesting that it can generate more informative candidates for the user to analyze.

4.2. Evaluation of Re-annotation in Interactive SED

In this section, we reported our findings on the interactive task with RishWheel for the purpose of re-annotation. The experiments were run using two setups, 5 and 25 minutes per event in search, respectively. Since human annotation is very time-consuming, we only used one annotator in the experiments, who did two runs for each approach in comparison. The results reported here are the average of the two runs.

The performance in the interactive runs are evaluated based on a metric called Detection Cost Rate (DCR) adopted in the TRECVID evaluation [1]. Basically, DCR is a linear combination of two errors: missed detections (MD) and false alarms (FA). A lower DCR indicates a better performance. DCR reflects a tradeoff between missed detections (MD) and false alarms (FA) by weighing them differently in scoring. In our experiments, w_f and w_m were set based on the weight used in DCR i.e. $w_m = 10w_f$.

We compared RishWheel on the TRECVID dataset with multiple interactive systems: (1) Interaction with simple confidence ranking (*Conf-Sys*); (2) Interaction based on confidence ranking and temporal locality search method [7]; (3) Interaction based on margin ranking with temporal information considered [10]; and (4) Interaction based on entropy ranking [3]. Note that the system [3] was originally developed for image annotation, so we modified it for event annotation here.

	Intermediate	Conf-Sys	Cai[7]	Cheng[10]	Gosselin[3]	RishWheel
CellToEar	1.0004	1.0005	1.0002	**1.0001**	1.0005	**1.0001**
Embrace	0.9118	0.9103	0.9035	0.9011	0.9085	**0.8985**
ObjectPut	0.9545	0.9518	0.9467	0.9325	0.9455	**0.9246**
PeopleMeet	0.9789	0.9683	0.9567	0.9489	0.9668	**0.9401**
PeopleSplitUp	0.8755	0.8665	0.8721	0.8566	0.8734	**0.8432**
PeopleRuns	0.7288	0.7258	0.7254	0.7104	0.7225	**0.7089**
pointing	0.9542	0.9428	0.9401	0.9375	0.9411	**0.9322**

Table 2. DCRs of different interactive systems on the TRECVID SED dataset (5-minute setup). A total of 5 minutes is alloted for each event in search. 'Intermediate' is the original input to all the interactive systems listed here.

	Intermediate	Conf-Sys	Cai[7]	Cheng[10]	Gosselin[3]	RishWheel
CellToEar	1.0004	1.0002	1.0001	0.9985	1.0002	**0.9965**
Embrace	0.9118	0.8823	0.8835	0.8783	0.8744	**0.8434**
ObjectPut	0.9545	0.9126	0.9058	0.8925	0.9023	**0.8675**
PeopleMeet	0.9789	0.9358	0.9275	0.9095	0.9118	**0.8813**
PeopleSplitUp	0.8755	0.8442	0.8362	0.8247	0.8326	**0.8035**
PeopleRuns	0.7288	0.7125	0.7094	0.6705	0.6998	**0.6414**
pointing	0.9542	0.9196	0.9124	0.9025	0.9082	**0.8796**

Table 3. DCRs of different interactive systems on the TRECVID SED dataset (25-minute setup). See Table 4.2 for more descriptions.

As can be seen from Table 2 and 3, all interactive systems show improvements over the original input (i.e. Intermediate) from the automatic event detection system (Except on *CellToEar* in some systems). Again as expected, the system based on the proposed risk analysis achieves the best performance. The margin-based and entropy-based systems outperform the simpler confidence-based interaction, which is consistent to the finding in the ranking quality evaluation described in Section 4.1. For the event *CellToEar*, none of the systems seem to yield improvement. This is largely because *CellToEar* is the one of the most difficult events to detect by automatic methods and it lacks temporal relations with other events. On the other hand, *Pointing*, another difficult event to detect, clearly gains substantial benefit from interactive analysis due to the rich temporal context surrounding the event (in the data, *Pointing* indicates strong temporal relations with *PeopleMeet* and *PeopleSplitUp*).

5. CONCLUSION

In this paper we have presented RishWheel, an interactive system for surveillance event detection. Driven by a novel risk ranking technique, the system allows for effective and dynamic human exploration of temporal relations in the intermediate detection results. The effectiveness of RiskWheel has been demonstrated in a formal user study where re-annotation was performed by human to improve event detection performance within a limited lime.

RishWheel can not only be used as a visual analytics tool for SED task, but as an interactive annotation/degugging system. In future work, we plan to apply RiskWheel to more applications, such as object detection and face tracking [21, 22], and conduct more thorough user studies.

6. REFERENCES

[1] P. Over, G. Awad, M. Michel, and J. Fiscus et al, "Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[2] J. Fogarty, D. Tan, A. Kapoor, and S. Winder, "Cueflik: Interactive concept learning in image search," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, CHI 2008, pp. 29–38.

[3] P. Gosselin and M. Cord, "Active learning methods for interactive image retrieval," *Trans. Img. Proc.*, vol. 17, pp. 1200–1211, 2008.

[4] A. Yao, J. Gall, C. Leistner, and L. Van Gool, "Interactive object detection," in *CVPR*, 2012, pp. 3242–3249.

[5] Ajay J. Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos, "Breaking the interactive bottleneck in multi-class classification with active selection and binary feedback," in *CVPR 2010*. pp. 2995–3002, IEEE.

[6] Carl Vondrick and Deva Ramanan, "Video annotation and tracking with active learning," in *Neural Information Processing Systems*, 2011, NIPS 2010.

[7] C. Yang, C. Qiang, and L. Brown et al, "Cmu-ibm-nus@trecvid 2012:surveillance event detection," in *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[8] Z. Zhao, Y. Zhao, , and W. Wang et al, "Bupt-mcprl at trecvid 2012," in *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[9] Qiang Chen, Yang Cai, Lisa Brown, Ankur Datta, Quanfu Fan, Rogerio Feris, Shuicheng Yan, Alex Hauptmann, and Sharath Pankanti, "Spatio-temporal fisher vector coding for surveillance event detection," in *Proceedings of the 21st ACM International Conference on Multimedia*, 2013, MM '13, pp. 589–592.

[10] Quanfu Fan Rogerio Feris Sharath Pankanti Yu Cheng, Lisa Brown, "Ibm-northwestern@trecvid 2013: Surveillance event detection," in *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[11] Yu Cheng, Zhengzhang Chen, Lu Liu, Jiang Wang, Ankit Agrawal, and Alok Choudhary, "Feedback-driven multiclass active learning for data streams," in *Proceedings of the 22Nd ACM International Conference on Conference on Information and Knowledge Management*, 2013, pp. 1311–1320.

[12] Hongliang Fei Fei Wang Alok Choudhary Yu Cheng, Zhengzhang Chen, "Batch mode active learning with hierarchical-structured embedded variance," in *SDM*, 2014.

[13] Burr Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, 2009.

[14] W. Frank, C. Archambeau, and J. Gasthaus et al, "A stochastic memoizer for sequence data," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 1129–1136.

[15] W. Frank, J. Gasthaus, and C. Archambeau et al, "The sequence memoizer," *Communications of the ACM*, vol. 54, no. 2, pp. 91–98, 2011.

[16] G. Zipf, *Selective Studies and the Principle of Relative Frequency in Language*, Harvard University Press, Cambridge, MA, 1932.

[17] Sharath Pankanti Alok Choudhary Yu Cheng, Quanfu Fan, "Temporal sequence modeling for video event detection," in *In the 27th IEEE Conference on Computer Vision and Pattern Recognition*. CVPR 2014, IEEE.

[18] Harold W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1–2, pp. 83–97, March 1955.

[19] Kalervo Järvelin and Jaana Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Trans. Inf. Syst.*, pp. 422–446, 2002.

[20] Bruce Croft, Donald Metzler, and Trevor Strohman, *Search Engines: Information Retrieval in Practice*, Addison-Wesley Publishing Company, USA, 1st edition, 2009.

[21] Alexander Sorokin and David Forsyth, "Utility data annotation with amazon mechanical turk," *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8.

[22] Welinder, "Online crowdsourcing: rating annotators and obtaining cost-effective labels," .