The TRECVid 2008 Event Detection Evaluation

Travis Rose, Jonathan Fiscus, Paul Over, John Garofolo
NIST
100 Bureau Dr. Stop 8940
Gaithersburg, MD 20899
{jfiscus,paul.over,john.garofolo}@nist.gov

Martial Michel
Systems Plus
One Research Court, Suite 360
Rockville, MD 20850
martial.michel@nist.gov

Abstract

This paper is a summary of the 2008 TRECVid Event Detection evaluation track. TRECVid is a laboratory-style evaluation that aims to model real world situations or significant component tasks. The event detection evaluation was organized to address detection of a set of specific events that would be of potential interest to an operator in the surveillance domain. This paper describes the video data, evaluation tasks, evaluation metrics, and results of the event detection evaluation.

1. Introduction

TRECVid [14] is a laboratory-style evaluation series coordinated by the National Institute of Standards and Technology (NIST) that is intended to provide a research, development, and evaluation testbed for video/multimedia analysis and content-based retrieval technologies. It provides large realistic datasets, common metrics, and a forum for technical information interchange. The annual series includes a post-evaluation workshop where the evaluation results and approaches are discussed. TRECVid evaluation tracks (tasks) have recently included automatic segmentation, indexing, content-based retrieval of digital video (broadcast news in English, Arabic, and Chinese), high-level feature extraction, search (interactive, manually-assisted, and/or fully automatic), and content-based copy detection.

In 2008, a new event detection evaluation track was created within TRECVid. The goal of the evaluation track was to support the development of technologies to detect visual events (people engaged in particular activities) in a large collection of video data. The evaluation was implemented by NIST using 100 h. of multi-camera airport surveillance domain data collected by the UK Home Office and was ground-truthed by the University of Pennsylvania's Linguistic Data Consortium. The primary goal of the evaluation track was to present the research community with a minimally constrained evaluation task representing a variety of computer vision challenges and a realistic annotated video dataset. The secondary goal of the evaluation was to create a realistic baseline of the state-of-the-art, both to gauge the maturity of the technologies regarding future behavior-based video surveillance applications and to accelerate research in this area. With a few notable exceptions, previous computer vision evaluation challenges employing small amounts of data that enabled researchers to implement impractical-to-field algorithms that run in hundreds or thousands of times realtime. The TRECVid Event Detection evaluation sought to address the deficiencies of these efforts with regard to visual event/activity detection along several dimensions:

1) Realistic data: the dataset was realistic for the surveillance domain in terms of size, duration, environment/scene complexity, people movement and interaction, and (realistically low) density of the target events per hour of video.

2) Naturally-occurring target events: the target event set was developed from an analysis of the data with needs of surveillance applications in mind – events were naturally occurring and of varying complexity, duration, and frequency.

3) Human-centered event definitions: minimally constrained descriptive (rather than the more traditional prescriptive) event definitions were used that did not arbitrarily exclude event occurences due to occluded persons or objects.

4) Temporally-oriented tasks: the task was defined with temporal rather than spatial extents; temporal-based metrics shifted the focus from the traditional spatial detection and clip detection domains to the temporal domain – thus enabling an expansion of the annotated data by orders of magnitude (and creating a requirement for faster algorithms), and a realistic examination of detection tasks where the prior probability of detection is relatively low.

5) Significant data: the amount of collected data was

sufficiently large (100 h.) to estimate performance for low-frequency events. The dataset was divided in half: 50 h. of evaluation data and 50 hours of training/development data. The quantity of development data supports novel research in automatic modeling techniques. The dataset and accompanying annotations were two orders of magnitude larger than previous comparable datasets (ETISEO 2006 [1], AVSS 2007 [8], and PETS 2009 [3]).

Previous visual event/activity detection evaluations have been extremely constrained and under-resourced with regard to the above dimensions. This has resulted in significantly overstating performance. These incrementally more difficult evaluations, while having provided the computer vision research community with attainable goals, did not provide a point of reference for the true maturity of the technology's ability to address real application challenges. It is important to complement these formative evaluations with a more realistic picture of the challenges that are yet to be addressed as well as sufficient resources to carry out the research to address these challenges. These challenges include achieving robustness, real-time processing speed, and accuracy. The TRECVid Event Detection evaluation supports these needs and has provided an important baseline against which the next several years of progress can be measured.

2. Evaluation overview

2.1. Evaluation tasks

The event detection evaluation included two tasks: retrospective event detection and freestyle analysis.

2.1.1 Retrospective Task

For the retrospective task, participants were given a textual definition of an event for which their automatic systems were required to temporally locate (via a start and end time) all observations of the event within a single camera view. For each detected event observation, systems were to provide a numeric "detection score" indicating the strength of evidence that the event occurred, and a binary decision (based on a threshold applied to the detection score) whether or not the event occurred, so as to optimize against a single metric. Researchers were asked to build algorithms to detect at least three of a set of ten pre-defined events (see section 2.3 for the event defintions). The task was retrospective because systems could perform multiple passes over the video prior to outputting a list of putative event observations. To simplify the retrospective task, systems were not required to make use of the cross-camera synchronization.

Figure 1. London Gatwick Airport camera views.

2.1.2 Freestyle Task

In the freestyle task, participants were asked to define tasks that are pertinent to the airport video surveillance domain and that could be implemented on the dataset. Freestyle submissions were required to include a rationale, clear definitions of the task, performance measures, reference annotations, and baseline system implementation.

2.2. Data

The 2008 dataset consisted of about 100 h. of indoor airport surveillance video collected in a busy airport environment by the United Kingdom (UK) Home Office Scientific Development Branch (HOSDB). The dataset utilized five frame-synchronized cameras on ten different days, recording for about two hours each day. It was collected in the same location and using the same equipment as the Imagery Library for Intelligent Detection System's (iLIDS) multiple camera tracking scenario [7].

The dataset was divided into equally-sized development and evaluation subsets. The videos were distributed as Moving Picture Experts Group (MPEG)-2 compressed, deinterlaced, Phase Alternating Line (PAL) format, 720 x 576 resolution, 25 frames/second files. Because of the size of the dataset, both the development and evaluation video data were released at the same time to allow for the most compute time for feature extraction, tracking algorithms, etc.

The camera views are shown in Figure 1. The views show (from left to right, top to bottom) a controlled access door, a waiting area with benches, a waiting area with kiosks, an elevator close-up view, and a transit area.

2.3. Annotation

The events used for 2008 (listed below) were chosen based on their range of expected difficulty. They are briefly defined as follows:

1. CellToEar: someone puts a cell phone to his/her ear.

2. ElevatorNoEntry: elevator doors open with a person

waiting in front of them, but the person does not get in before the doors close.

3. Embrace: someone puts one or both arms at least part way around another person.
4. ObjectPut: someone drops or puts down an object.
5. OpposingFlow: someone moves through a controlled access door opposite to the normal flow of traffic.
6. PeopleMeet: one or more people walk up to one or more other people, stop, and some communication occurs.
7. PeopleSplitUp: when one or more people separate themselves from a group of two or more people, who are either standing, sitting, or moving together communicating, and then leaves the frame
8. PersonRuns: someone runs.
9. Pointing: someone points.
10. TakePicture: someone takes a picture.

These events represented several different types of actions such as: single person tracking (events 2, 5, 8), single person interacting with objects (events 1, 4, 10), multi-person tracking people interaction (events 3, 6, 7), and body limb movement (events 1, 3, 4, 9, 10), which pose difficult computer vision challenges. The events were defined in an annotation guidelines document [13], which was created in order to define the characteristics of the events for annotators as well as system developers. The guidelines were designed to be simple in order to reasonably capture human intuition and not artificially constrain the tasks. Event observations were made according to a "reasonable interpretation rule." The rule was, "if according to a reasonable interpretation of the video, the event must have occurred, then it is a taggable event." The videos were annotated by the Linguistic Data Consortium (LDC) using the Video Performance Evaluation Resource (ViPER) tool [2]. Annotations were represented in a ViPER XML format to encode each observation's time interval. Annotators were given 5 events to annotate for each pass over the data. 5 % of the data was dually annotated to study consistence rates.

Figure 2 shows the rates of instances of each event per hour (called $Rate_{Target}$) for both the development and evaluation datasets. The graph shows the select events have a wide range of occurrence frequencies. The rates of occurrence for the OpposingFlow and TakePicture events indicate that a 50 h. test set is too small based on the Rule of 30[1][16]. Nevertheless, they are highly relevant to the surveillance application and were therefore kept in the evaluation. For the ObjectPut and Pointing events, there was some disparity in their rates of occurrence in the development vs. the evaluation datasets which reflects a change in the annotation specification.

[1]To be 90 % confident that the true error rate for a binary detection task is within +/- 30 % of the measured error rate, the dataset must be large enough for 30 errors to occur.

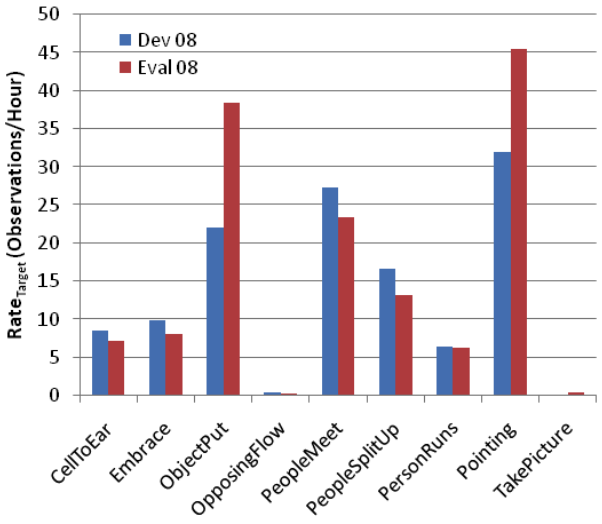

Figure 2. The density of event observations in the development and evaluation datasets.

2.4. Evaluation Procedure

The retrospective event detection task was defined as a detection task in which system performance was measured in terms of missed detection and false alarm rates. The evaluation procedure, which is closely related to the procedure that was used for the NIST Spoken Term Detection evaluation [5], was composed of three steps: (1) the system outputs were aligned to ground truth, (2) a Decision Error Tradeoff [10] (DET) curve was computed to graphically depict the interaction of the two error types, and (3) the DET curve was analyzed to find the Minimal Normalized Detection Cost Rate and the Actual Normalized Detection Cost Rate. The following section is a summary of the evaluation procedures, which are more fully documented in the evaluation plan [6].

2.4.1 System Output/Reference Alignment

Event instances (observations) could occur at any time in the video and have any duration. Therefore, the evaluation did not rely on an arbitrary predefined segmentation scheme. In order to determine which observations were correctly detected, an optimal, one-to-one alignment between system observations and reference observations was found which minimized the error rates. Alignment was performed using the Hungarian solution to the bipartite graph

matching problem[15], in which system observations were represented as one set of nodes and reference observations were represented as a second set of nodes. Correct detections were mapped nodes based on a kernel function that measured the temporal similarity between the annotated reference event observations and the system-generated event observations combined with the system's detection scores. Missed detections were unmapped reference nodes and false alarms were unmapped system nodes.

2.4.2 Decision Error Tradeoff Curves

The performance of the computer vision algorithms were assessed using a variation of the Detection Error Tradeoff (DET) curve [10]. Traditionally, DET curves plot the probability of false alarms (P_{FA}) vs. the probability of missed detection (P_{Miss}) .

$P_{FA} = N_{spurious}/N_{NT}$
$P_{Miss} = N_{misses}/N_{Ref}$

where: $N_{spurious}$ is he number of incorrect detections, N_{NT} is the number of opportunities for incorrect detection (i.e., non-target trials), N_{Miss} is the number missed detections, and N_{Ref} is the number of true event observations.

Counting N_{NT} is problematic for "streaming" detection technologies like event detection in that: multiple event observations can occur simultaneously, observations can begin at any frame, and observations can have any duration. While it would be possible to develop a formula to calculate N_{NT} based on these factors, the resulting normalization (via the denominator) would be arbitrary and unintuitive. Instead, a more natural expression of a system's false alarm rate for event detection is to normalize the false alarm errors by the amount of processed source material via the Rate of False Alarms (R_{FA}) where N_{CamHrs} is the number of camera hours of processed material.

$R_{FA} = N_{spurious}/N_{CamHrs}$

The unit of R_{FA} is false alarms per hour which is easily interpreted by down-stream users of the technology. Strictly speaking, R_{FA} is a biased estimate of the Type I Errors since true observations are included in N_{CamHrs}. However, when the rate of occurrence is low, as it is for event detection, the bias is small.

2.4.3 Random System DET Curves

Another result of using R_{FA} was that random DET curves could no longer be constructed analytically. Instead, we computed random DET curves using a Monte Carlo simulation technique. The random DET curve for each event type was constructed by averaging the DET Curves for 50 pairs of a random test set and system outputs. Each pair was generated as follows:

- The development set's observation densities defined the number of reference observations.
- One thousand system observations per hour were generated.
- Temporal placement of the observations were random with a uniform distribution throughout the timeline.
- The duration of the observations, both reference and system, were modeled as random variables with their means and standard deviations measured from the development dataset.
- Decision scores were randomized using a unit normal distribution.

The algorithm was able to generate DET curves for all events except OpposingFlow and TakePicture. The test set size to develop statistically valid random curves for these two events would be 642 and 170 h. respectively which was greater than our computing infrastructure could handle.

2.4.4 Normalized Detection Cost Rates

While DET curves provide a view of system performance across a wide range of ratios between misses and false alarms, it is difficult to compare performance across systems because developers may tune to different operational points. The Speaker Recognition [11], Topic Detection and Tracking, and Spoken Term Detection [5] communities have used Detection Cost Functions (DCFs) as a means to combine the miss and false alarm error rates into a composite metric. DCFs are a linear combination of the two error types using a set of predefined constants that include the event prior and weights for each error type. A cost statistic is a measure of the increased *cost* to the user for using the system when the system emits either miss or false alarm errors. For the event detection evaluation, we could not use the DCF model because DCFs make use or P_{FA}. Instead, we defined a Normal Detection Cost Rate (NDCR) model that used R_{FA} instead of P_{FA}. Thus, NDCR is:

$NDCR = P_{miss} + \beta \times R_{FA}$,
where
$\beta = \frac{Cost_{FA}}{Cost_{Miss} \times R_{Target}}$
$C_{Miss} = 10; C_{FA} = 1; R_{Target} = 20/hour$

The constants chosen for the evaluation were motivated by discussions with the research and user communities. R_{Target}, the *a priori* rate of event observations, was arbitrarily selected to be in the middle of the distribution of event densities in Figure 2. While the selection of a single

R_{Target} is not a good fit for all events, it was a simplifying assumption for the evaluation. NDCR was normalized to have a range of [0,∞) where 0 would be for perfect performance, 1 would be the cost of a system that provides no output, and ∞ was possible.

Two NDCRs were calculated for each event. The Minimum NDCR (MinNDCR) is computed by finding the point on the DET curve that minimizes NDCR. The Actual NDCR (ActNDCR) is computed by using P_{Miss} and R_{FA} calculated from the set of putative system observations having "yes" decisions based on a system-tuned threshold applied to the decision scores. The difference between Minimum NDCR and Actual NDCR provides insight into how well the system was optimized.

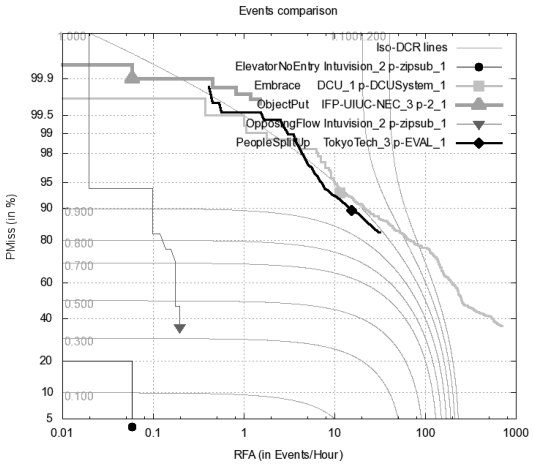

Figure 3. Lowest MinNDCR system runs the ElevatorNoEntry, Embrace, ObjectPut, OpposingFlow, and PeopleSplitUp events.

3. Evaluation analysis

3.1. 2008 Results

Sixteen (16) teams consisting of 17 organizations submitted 72 event-runs in the retrospective event detection task. The sites included: Athens Institute of Technology (AIT), Brno University of Technology, (BUT), Carnegie Mellon University (CMU), Dublin City University (DCU), Fudan University (FD), University of Illinois and NEC (IFP-UIUC-NEC), intuVision, Chinese Academy of Science (MCG-ICT-CAS), NHK Science and Technology Research Laboratory (NHKSTRL), Queen Mary University of London (QMUL-ACTIVA), Shanghai Jiao Tong University (SJTU), Tunhai University (THU-MNL), TokyoTech, Toshiba, Universidad Autonoma de Madrid (UAM), and University of Central Florida (UCF). University of Southern California (USC) [9] was the only site that participated on the freestyle analysis track. The USC study, which is not reported on here, focused on a small sample of PeopleMeet event instances.

The lowest and average MinNDCRs for each run/event submitted for the retrospective task appear in Table 1. We report only the MinNDCRs for each event to focus on the best possible score for each run by factoring out threshold setting for the ActualNDCRs. This gives a better, less noisy, view of system performance but ignores threshold setting and cross-event calibration. The events that most sites submitted systems for were the single person tracking events, OpposingFlow, PersonRuns, and ElevatorNoEntry with 15, 14, and 12 submissions respectively. For the rest of the events, either 4 sites or 8 sites submitted system runs.

The lowest MinNDCRs for events across all systems formed two "clusters": two of the "single person tracking" events (ElevatorNoEntry and OpposingFlow), and the rest of the events. The lowest MinNDCR for the ElevatorNoEntry was 0.0003. The lowest MinNDCR for OpposingFlow was 0.354. Several factors are likely to have made high performance for these tasks possible: (1) for each event, the observations could only occur in one specific location in the camera field, (2) the size of the person(s) in frame was up to 1/2 of the image height, and (3) existing optical flow, door open/close detection, and person detection/tracking technologies could be used. The lowest MinDCRs for all systems for the rest of the events ranged from 0.851 to 1.0, with the PersonRuns being the only single person tracking event within this group. Several factors are likely to have made these events difficult to detect. Two factors in particular were expected to play a role: (1) the observations could occur in all camera views in any location making the size of the person in frame highly variable and requiring extensive compute time, and (2) most events involve either limb tracking, object detection (besides people), or both.

Figures 3 and 4 contain the "best" DET curve for each event across systems. The graph shows that for each event, at least one site was able to achieve a MinNDCR less than 1.0. As noted above, a MinNDCR of 1.0 corresponds to a system that produces no output. The points on the curve correspond to the MinNDCR point for each curve.

3.2. System-Mediated Reference Adjudication

The results presented in Table 1 were obtained after reviewing the system outputs for missing events in the reference annotation. We refer to this process as 'adjudication': *using additional knowledge sources to improve reference quality*. NIST and LDC designed an adjudication process to review the top 100 system-generated observations that were most likely to be erroneously counted as missed detections. The criteria for ordering the list of observations took into account the number of systems that agreed the observation occurred and then by the average decision score after being converted to a within-system percentile score. Figure 5 shows the process to build ViPER annotation files for adju-

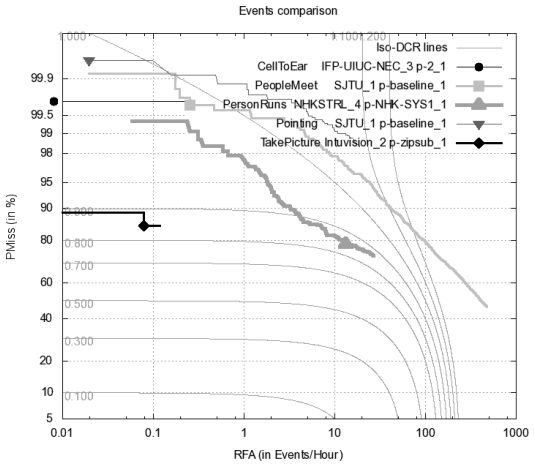

Figure 4. Lowest MinNDCR system runs for the CellToEar, PeopleMeet, PersonRuns, Pointing, and TakePicture events.

Event	Lowest Minimum NDCR	Average Minimum NDRC	Number of Submissions
CellToEar	0.997	1.018	4
ElevatorNoEntry	0.0003	0.719	12
Embrace	0.990	1.013	5
ObjectPut	0.999	1.133	5
OpposingFlow	0.354	0.790	15
PeopleMeet	0.998	1.003	8
PeopleSplitUp	0.973	0.994	5
PersonRuns	0.851	1.000	14
Pointing	1.000	1.061	4
TakePicture	0.852	0.955	6

Table 1. Lowest Minimum Normalized Detection Cost Rates for systems designed to automatically detect events. A full table can be found in the expanded version of this paper at http://www.itl.nist.gov/iad/mig/tests/trecvid/2008

dicators to review. This process consisted of five steps:

1. Score each system against the reference annotation to generate a list of system false alarms (i.e., unmapped sys) and use those for the next step,
2. Perform an iterative, multi-system alignment, using a technique similar to the multi-file string alignment technique used by ROVER (Recognizer Output Voting Error Reduction) [4], to find observations that multiple system's "agree" on,
3. Build a ViPER annotation file for each temporally separated observation by incorporating any existing reference annotations, the temporal extent of the suspected observation, and the temporal extents of all system observations,
4. Sort the set of putative observations,
5. Human annotators judge whether or not the observation should have been annotated,

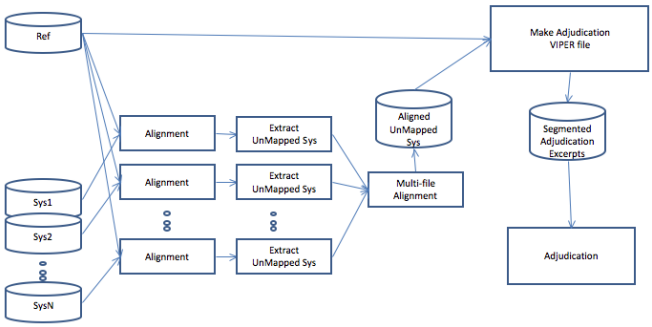

Figure 5. Adjudication of reference annotation using multi-aligned system output.

6. Newly found observations were added to the reference set of observations.

Only 185 new observations (or an additional 2.5 %) were found across all events, and the average MinNDCR change was -0.0003. The greatest change in measured performance was a decrease of 0.14 in MinNDCR for the TakePicture event, where the event density is very low and missing 4 observations had a big effect. In general, the net effect of adjudication on error rates was minimal. We believe it is likely due to the high error rates of the systems. For the 2009 evaluation, we will use multiple humans to annotate the video and adjudicate their annotations with the same process to improve the completeness of the reference annotations.

3.3. Human Performance Baselines

Comparing the results of the systems to that of humans and to a random DET curve is necessary to put the results of the evaluation with in the context of common baselines. During the annotation process, the LDC dually annotated 5 % of the video to check consistency. Upon review of the dual annotations, we found the annotators were very likely to miss valid event observations. In order to assess the extent of the recall problem, the LDC conducted 6-way annotation passes over about 35 minutes of data for all events. For this small initial study, the dataset was not balanced by cameras, the annotations were conducted by more than six people, and some camera views were reviewed twice by the same annotator.

Figure 6 shows the average number of unique event observations for all possible combinations of 1,2,3...6 annotators. The graph, which is similar to results in [12], shows that as additional annotators reviewed the data, more observations were found. Despite its limitations, this study indicates that recall was improved by increasing the number of annotators to review the video. All observations were reviewed using an adjudication process similar to that used

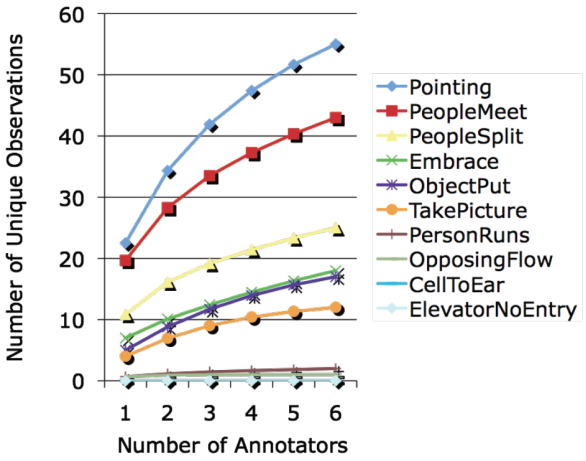

Figure 6. Study of unique event observations found with 1-6 annotators.

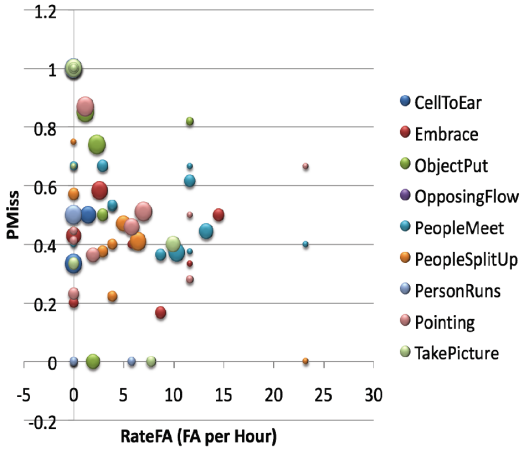

Figure 7. Plot of human detection performance across all events.

for the system outputs. As a result of the process, annotation error rates were computed for each event/annotator and plotted in the bubble plot of Figure 7. The size of the bubble is proportional to the amount of data reviewed by an individual annotator, with the maximum size corresponding to 35 minutes of data. On average, humans missed half of the observations when they attempted to annotate five events within the same annotation pass. The low recall could be a limitations of human performance or the annotation procedure. In a subsequent small study, when three annotators were asked to look for three events (ObjectPut, PeopleMeet, and Pointing) per pass rather than five, recall was improved by 438 %, 78 %, 164 % respectively. Additional studies are planned for the 2009 evaluation.

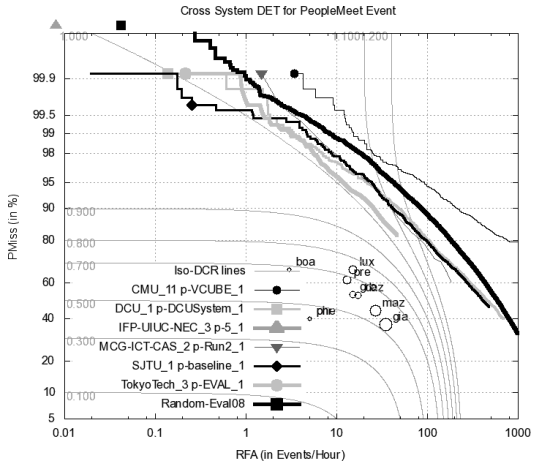

Figure 8. Best run for systems that detected the PeopleMeet event.

3.4. Random System Performance Baselines

Figure 8 is a DET plot for all sites that built a system to detect PeopleMeet events. As the graph shows, most sites' DET curves were above the 1.000 NDCR line that goes off the graph towards the upper left hand corner. However, the darkest curve indicates the average performance of a random system (as described in Section 2.4.2). Although most systems' MinNDCRs were above 1.0, all but one site beat the random system. The graph also contains circles indicating the error rates for the annotators involved in the 6-way study using the adjudicated 6-way annotation as the reference. As expected, the humans out-performed the systems.

We characterize system performance with respect to two important baselines, namely that of the human annotation and a random system. Importantly, we note there is room for improvement in both the creation of annotations and the detection performance that systems can achieve. This initial evaluation provided a baseline for algorithms aiming to detect a variety of events and operating on a large, realistic surveillance video corpus. We anticipate that the next evaluation cycle will show improvements in automatic detection. Provided detection systems improve over time, future event annotation methods may ultimately benefit from the application of automatic detectors to the annotation process as well.

4. Conclusions

The TRECVid Event Detection evaluation provided an initial baseline of algorithms designed to process a realistic dataset. The primary evaluation task focused on naturally occurring events that were derived from an analysis of the data, resulting in selection of events that are challenging to computer vision technology. The event definitions were created with an emphasis on human intuition that did not

artificially constrain the task due to occlusion. In addition, the events were evaluated with respect to temporal extents rather than spatial extents. Temporal-based metrics shifted the focus from traditional spatial detection and clip detection to the temporal domain, enabling increased annotation by orders of magnitude and a realistic examination of detection tasks where the prior probability of detection is relatively low. The amount of data (100 h.) was challenging and sufficiently large that a statistically meaningful evaluation of event detection could be carried out. The evaluation resulted in state-of-the-art benchmarks for several naturally occurring events that demonstrated the feasibility of automatically detecting these events in video. Several teams' systems did better than the random system baseline despite less than a full year of development time.

5. Acknowledgments

We thank the LDC for performing the annotation and collaborative analyses for the TRECVid surveillance event detection. We thank the UK Home Office for contributing data for use in the TRECVid evaluation, and the University of Maryland for the open source ViPER-GT software package. We also thank George Doddington for collaborating with us to define the NDCR function and Mehmet Yilmaz for computing human error rates. This evaluation is funded in part by the Department of Homeland Security Science and Technology Directorate Predictive Screening project.

6. Disclaimer

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

References

[1] D. Cher, F. Bremond, and J.-L. P. Vilchis. ETISEO: Video Understanding Evaluation. http://www-sop.inria.fr/orion/ETISEO/, 2007 (accessed June 30, 2009).

[2] D. Doermann and D. Mihalcik. Tools and Techniques for Video Performance Evaluation. In *ICPR*, volume 4, pages 167–170, 2000.

[3] J. Ferryman, A. Shahrokni. An Overview of the PETS 2009 Challenge. In *Proc. of 11th IEEE Wks. on Perf. Eval. of Tracking and Surveillance (PETS 2009)* pages 25–30, 2009.

[4] J. Fiscus A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER) . In *Proceedings of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347-352, Santa Barbara CA.

[5] J. Fiscus, J. Ajot, J. Garofolo, G. Doddington. Results of the 2006 Spoken Term Detection Evaluation. In *2007 Workshop in Searching Spontaneous Conversational Speech EuroSpeech 1999*, ACM SIGIR Forum, Vol. 41, No. 2, Dec. 2007.

[6] J. Fiscus and R. T. Rose. 2008 TRECVid Event Detection Evaluation Plan. http://www.itl.nist.gov/iad/mig/tests/trecvid/2008/doc/EventDet08-EvalPlan-v07.htm, 2008 (accessed June 30, 2009).

[7] U.K. Home Office Centre for Protection of National Infrastructure. Imagery library for intelligent detection systems. http://scienceandresearch.homeoffice.gov.uk/hosdb/cctv-imaging-technolo gy/video-based-detection-systems/i-lids/, 2007 (accessed June 30, 2009).

[8] P. Hosmer. Advanced Video and Signal-based Surveillance. http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007_ss_challenge.html, 2007 (accessed June 30, 2009).

[9] S. C. Lee, C. Huang, and R. Nevatia. Definition, detection, and evaluation of meeting events in airport surveillance videos. In *TRECVid workshop papers*, 2008.

[10] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In *Proc. Eurospeech '97*, pages 1895–1898, 1997.

[11] A. Martin and M. Przybocki. The 1999 NIST Speaker Recognition Evaluation, Using Summed Two-Channel Telephone Data for Speaker Detection and Speaker Tracking. In *EuroSpeech 1999*, pages 2215–2218, 1999.

[12] J. Nielsen and T. K. Landauer. A mathematical model of the finding of usability problems. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 206–213, New York, NY, USA, 1993. ACM.

[13] H. Simpson, P. Over, J. Fiscus, and T. Rose. TRECVid 2008 Event Annotation Guidelines Version 1.6. http://www.itl.nist.gov/iad/mig/tests/trecvid/2008/doc/TRECVid08_Guidelines_v1.6.pdf, 2008 (accessed June 30, 2009).

[14] A. F. Smeaton, P. Over, and W. Kraaij. High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements. In A. Divakaran, editor, *Multimedia Content Analysis, Theory and Applications*, pages 151–174. Springer Verlag, Berlin, 2009.

[15] Hungarian Algorithm http://en.wikipedia.org/wiki/Hungarian_algorithm.

[16] G. Doddington. Speaker Recognition Evaluation Methodologies in *Reconnaissance du Locuteur et Ses Applications Commerciales et Criminalistiques*, pages 60-66, Avignon 1998