Framework for Performance Evaluation of Face, Text, and Vehicle Detection and Tracking in Video: Data, Metrics, and Protocol

Rangachar Kasturi, *Fellow*, *IEEE*, Dmitry Goldgof, *Fellow*, *IEEE*,
Padmanabhan Soundararajan, *Member*, *IEEE*, Vasant Manohar, *Student Member*, *IEEE*,
John Garofolo, Rachel Bowers, *Member*, *IEEE*, Matthew Boonstra, *Student Member*, *IEEE*,
Valentina Korzhova, *Student Member*, *IEEE*, and Jing Zhang, *Student Member*, *IEEE*

Abstract—Common benchmark data sets, standardized performance metrics, and baseline algorithms have demonstrated considerable impact on research and development in a variety of application domains. These resources provide both consumers and developers of technology with a common framework to objectively compare the performance of different algorithms and algorithmic improvements. In this paper, we present such a framework for evaluating object detection and tracking in video: specifically for face, text, and vehicle objects. This framework includes the source video data, ground-truth annotations (along with guidelines for annotation), performance metrics, evaluation protocols, and tools including scoring software and baseline algorithms. For each detection and tracking task and supported domain, we developed a 50-clip training set and a 50-clip test set. Each data clip is approximately 2.5 minutes long and has been completely spatially/temporally annotated at the I-frame level. Each task/domain, therefore, has an associated annotated corpus of approximately 450,000 frames. The scope of such annotation is unprecedented and was designed to begin to support the necessary quantities of data for robust machine learning approaches, as well as a statistically significant comparison of the performance of algorithms. The goal of this work was to systematically address the challenges of object detection and tracking through a common evaluation framework that permits a meaningful objective comparison of techniques, provides the research community with sufficient data for the exploration of automatic modeling techniques, encourages the incorporation of objective evaluation into the development process, and contributes useful lasting resources of a scale and magnitude that will prove to be extremely useful to the computer vision research community for years to come.

Index Terms—Performance evaluation, object detection and tracking, baseline algorithms, face, text, vehicle.

1 Introduction

In this paper, we present a framework for objectively evaluating the performance of detection and tracking algorithms for face, text, and vehicle objects in video. Evaluations of performance are necessary not only to allow sponsors to gauge progress in their investments and determine the state of the art but also to assist researchers in developing and refining their algorithms. We present a comprehensive framework for evaluating object detection and tracking algorithms in video across several tasks and domains. In this paper, we provide a formal description of each of the evaluation tasks (Section 2.1), as well as the guidelines that were followed to create the reference annotations (Section 2.2). These tasks and guidelines were used to measure system performance in a series of evaluations that were administered for the US Government Video Analysis and Content Extraction (VACE) program. In the course of these evaluations, we developed two comprehensive metrics that were designed to roll up important measures of performance into a single score. By adopting a corpus and metric-centered approach to evaluation, the systematic effect of specific changes to algorithmic parameters could be objectively measured. The comprehensive metrics were designed to address both spatial (Section 3.2.1) and spatiotemporal (Section 3.2.2) performance dimensions. These measures were first employed in a previously published work in the evaluation of face and text detection and tracking algorithms [1], [2]. We further present a set of metrics (a pair each for detection and tracking), developed as part of the CLassification of Events, Activities, and Relationships (CLEAR) consortium, which are complementary in that they generate both a precision-centric and accuracy-centric view of performance (Sections 3.3.1, 3.3.2, 3.3.3, and 3.3.4). In order to characterize and validate these measures, we developed and evaluated

- R. Kasturi, D. Goldgof, V. Manohar, M. Boonstra, V. Korzhova and J. Zhang are with the Department of Computer Science and Engineering, University of South Florida, 4202 E. Fowler Ave., ENB 118, Tampa, FL 33620-5399.
E-mail: {rlk, goldgof, vmanohar, boonstra, korzhova, jzhang2}@cse.udf.edu.
- P. Soundararajan was with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620.
E-mail: paddu.sound@gmail.com.
- J. Garofolo is with the National Institute of Standards and Technology, 100 Bureau Drive, Stop 8940, Gaithersburg, MD 20899-8940.
E-mail: john.garofolo@nist.gov.
- R. Bowers was with the the National Institute of Standards and Technology, Gaithersburg, MD 20899.

Manuscript received 13 July 2007; revised 18 Jan. 2008; accepted 19 Feb. 2008; published online 10 Mar. 2008.
Recommended for acceptance by L. O'Gorman.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2007-07-0424.
Digital Object Identifier no. 10.1109/TPAMI.2008.57.

baseline algorithms for face detection and tracking (Section 4.1), as well as text (Section 4.2) and vehicle (Section 4.3) detection and tracking. These validation steps were accomplished by the consistent employment of a formal evaluation infrastructure that consists of a set of tools and processes required for evaluation (Section 5).

1.1 Overview of Related Evaluations

There is increasing interest in the application of task/corpus-based evaluation approaches for computer vision borne out by the number of recent public evaluation campaigns and the steadily increasing availability of annotated research and evaluation corpora. Image thresholding methods were investigated by Sezgin and Sankur [3]. Heath et al. [4] presented their work on evaluating edge detection algorithms. Range segmentation algorithms were compared by Hoover et al. [5]. Mikolajczyk and Schmid [6] analyzed the performance of descriptors computed for local interest regions. The gait identification challenge problem was formally addressed by Sarkar et al. [7]. The Fingerprint Vendor Technology Evaluation (FpVTE) [8] and the work by Cappelli et al. [9] evaluated fingerprint matching, identification, and verification systems with a goal of establishing a new common benchmark for an unambiguous comparison of fingerprint-based biometric systems. To provide a means for measuring progress and characterizing the properties of face recognition, the Face Recognition Technology (FERET) [10], the Face Recognition Vendor Test (FRVT) [11], and the Face Recognition Grand Challenge (FRGC) [12] programs were introduced by the US National Institute of Standards and Technology (NIST). Similarly to these, we systematically addressed the problem of object detection and tracking in video. The infrastructure created through this work will provide the broader computer vision community with useful lasting resources such as data, metrics, and tools for the evaluation of algorithms.

One of the major object detection and tracking evaluation frameworks is the Performance Evaluation of Tracking and Surveillance (PETS) program, a workshop series that was launched with the goal of evaluating visual tracking and surveillance algorithms. The first PETS workshop was held in March 2000. Since then, there have been several such workshops exploring a variety of surveillance domain applications. While the theme of early PETS workshops was target detection and tracking, as the technology has matured, the program has evolved to focus on event-level tasks. PETS Metrics [13], [14] is a derivative effort focused on providing an online service for automatically self-evaluating performance results. As of this writing, the PETS Metrics website supports motion segmentation metrics, but is expected to extend to tracking and event detection metrics.

Computers in the Human Interaction Loop (CHIL) [15], Augmented Multiparty Interaction (AMI) [16], Video Event Recognition Algorithm Assessment Evaluation (VERAAE), Evaluation du Traitement et de l'Interpretation de Sequences Video (ETISEO) [17], Cognitive Agent that Learns and Organizes (CALO) [18], NIST Rich Transcription Meeting Recognition Evaluations (RT) [19], and TREC VIDeo Retrieval Evaluation (TRECVID) [20] are additional programs that share a focus on developing and evaluating new algorithms for visual or audio-visual understanding technologies (see [21] for a review of several of these).

With respect to research in computer vision, especially those efforts addressing object detection and tracking in video, the VACE program was established to develop novel algorithms for automatic video content extraction, multimodal fusion, and event understanding. As of this writing, the VACE program is in Phase III. During VACE Phases I and II, the program made significant progress in the automated detection and tracking of moving objects such as faces, hands, people, and vehicles, as well as text in four primary video domains: broadcast news, meetings, surveillance, and Unmanned Aerial Vehicle (UAV) motion imagery. Initial results were also obtained on an automatic analysis of human activities and an understanding of video sequences. The performance evaluation effort in VACE Phase II (2002 to 2006) was carried out by the University of South Florida (USF) in collaboration with NIST and guided by an advisory forum including the evaluation participants. The formal evaluations described in this paper were part of the Phase II evaluations and took place in the summer of 2005.

The CLEAR [22] Evaluation Workshop was the first international evaluation effort that brought together two programs—VACE and CHIL. Some of the important benefits of this collaboration were the immediate availability of more data for the research community for algorithm development and evaluation and the evolution of widely accepted performance metrics that provide an effective and informative assessment of system performance [23]. NIST coordinated the VACE contributions to the consortium and the University of Karlsruhe coordinated the CHIL contributions.

1.2 Existing Work on Object Detection and Tracking Evaluation

Empirical evaluation is highly challenging due to the necessity of establishing a valid "ground truth" or "gold standard" to compare the system output to. In order for the evaluation to effectively measure performance, the ground truth must be the "ideal output" for exactly what the algorithm is expected to generate. The secondary challenge with quantitative validation is assessing the relative importance of different types of errors. Earlier work on the empirical evaluation of generic object detection and tracking algorithms [24], [25], [26], [27], [28], [29], [30], [31] presented a plethora of metrics specialized to measure different aspects of performance. While useful for error analysis, these metrics do not provide an effective measure of application task performance. Further, the resulting multitude of sometimes conflicting scores makes it difficult to quantitatively compare any two systems.

Earlier tracking technology evaluations either focused on the spatial aspect of the task (i.e., assessed correctness in terms of number and spatial location of tracked objects in each frame [26], [28]) or the object identity (which emphasizes maintaining a consistent identity for tracked objects over long periods of time [29]). In more recent work [30], [31], a spatiotemporal approach toward the evaluation of tracking systems was adopted. However, neither of these approaches provides the flexibility to adjust the relative importance of the spatial and temporal aspects of performance.

TABLE 1
Corpus Partitioning for Each Evaluation Task

	Data	Number of clips	Total mins
Per Domain	Micro-Corpus	5	10
	Training	50	175
	Testing	50	175

TABLE 2
Task-Domain Support Matrix (- = No, Y = Yes)

Task	Domain	
	Broadcast News ABC & CNN	Surveillance i-LIDS
Face Detect & Track	Y	–
Text Detect & Track	Y	–
Vehicle Detect & Track	–	Y

1.3 Evaluation Framework and Resources

Through this work, we created a set of summative metrics that addresses the important types of errors (misses, false alarms, spatial mismatches, and tracking ID switches) in a holistic way. The metrics center around an error minimization algorithm that maps the system output objects to the ground-truth objects. In order to support realistic automatic machine learning techniques, each evaluation included an unprecedented 2 hours of training data and 2 hours of test data. The video was Motion Picture Experts Group (MPEG)-2 encoded and each I-frame (*Intracoded frame*) was hand-annotated with bounding boxes to indicate the spatial location of the target objects. Only the I-frames were annotated so as to reduce the expense of the tedious hand-annotation process and to permit a maximal amount of data to be used. Approximately 29,000 I-frames were annotated for each evaluation task. A subset of the ground-truth data was also doubly annotated so that interannotator differences could be measured. This provided human baselines for the tasks as well as insight into the uncertainty of the evaluation results. Finally, we took evaluations further by measuring the performance of real algorithms on the generated data sets, developing baseline algorithms, and establishing performance benchmarks for each task and supported domain.

The remainder of this paper presents the details about the evaluation data, metrics, and process and is intended to be useful as a primer for future formal evaluations of object detection and tracking algorithms.

To support the continued use of this work, the following evaluation infrastructure is made available on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.57:

1. *Code.* This includes the source code for the scoring software and all of the baseline algorithms.
2. *Manual.* This includes the annotation guidelines document used to generate the ground-truth labeling, detailed description for using the source code, input/output file formatting, file naming nomenclature, and source video procurement procedure.
3. *Data.* This includes the XML-based ground-truth data (both training and testing) for each of the evaluation tasks described in this paper—face, text, and vehicle.

2 Data Resources and Annotation

This section describes the data sets developed during the evaluation process. These include both the source videos on which the algorithms were evaluated and the ground truth against which the algorithms were compared to compute the performance scores.

2.1 Video

The amount of training and testing data created for each domain, along with the microcorpus, are indicated in Table 1. It shows the data distribution for each domain. Initially, a small amount of data (“Microcorpus”) was created after extensive discussions with the research community to act as a seed for initial annotation experiments and to provide new participants with a concrete sampling of the data sets and the tasks. These discussions were coordinated as a series of weekly teleconferences with VACE contractors and other eminent members of the CV community. The discussions made the research community a partner in the evaluations and helped NIST and USF in selecting the video recordings to be used in the evaluations, in creating the specifications for the ground-truth annotations and scoring tools, and, most importantly, in defining the evaluation infrastructure for the program.

Evaluation support for a particular task-domain pair depends on the richness of the target objects being evaluated in that domain and the maturity of the state-of-the-art technology for the task. After careful deliberation, three task-domain pairs were chosen for evaluation in this paper (Table 2).

The broadcast news corpus for the face and text detection and tracking tasks was comprised of feeds from CNN and ABC and was distributed by the Linguistic Data Consortium (LDC)[1] [32]. The surveillance data set for the vehicle detection and tracking task was from the Imagery Library for Intelligent Detection Systems (i-LIDS)[2] corpus and was distributed by the United Kingdom’s Home Office via a collaboration with NIST. All of the source videos were presented to the systems in a consistent format (MPEG-2 standard, progressive scanned at 720 × 480 resolution. A group of Pictures (GOP) of 12 for the broadcast news corpus where the frame rate was 29.97 frames per second (fps) and a GOP of 10 for the surveillance data set where the frame-rate was 25 fps).

2.2 Task Definitions and Reference Annotations

A visual object detection and tracking task can be defined as spatially detecting the particular object(s) of interest in each individual frame and linking them through all of the frames

[1] http://www.ldc.upenn.edu.

[2] i-LIDS [Multiple Camera Tracking/Parked Vehicle Detection/Abandoned Baggage Detection] scenario data sets were developed by the UK Home Office and CPNI (http://scienceandresearch.homeoffice.gov.uk/hosdb/cctv-imaging-technology/video-based-detection-systems/i-lids/).

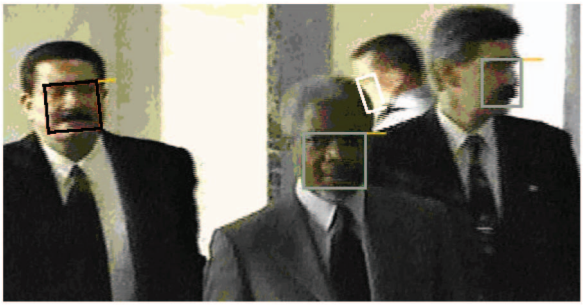


Fig. 1. Sample annotation for face. AMBIGUITY = 0 is marked by black boxes. AMBIGUITY = 1 is marked by gray boxes. AMBIGUITY = 2 is marked by white boxes.

in a given sequence. This definition applies for face, text, and vehicle objects.

Systems had to detect the specified object of interest and emit a unique object identifier number for each frame in which the object was detected/tracked. Each frame-object ID had to include the coordinates of the top-left corner of the bounding box for the object and the height and width of the box. The box coordinates were to be generated so as to match the rules for human annotation of the ground truth described later in this section. The primary test condition required the systems to be set for their expected Equal Error Rate operating point for each algorithm/task combination.

When generating the reference annotation, we adopted a bounding-box approach for marking object locations. This was done because a landmark-based approach was felt to be too algorithm specific. The major advantage of a bounding-box/spatial approach is that one can easily adapt the scoring algorithm to all sorts of different objects. However, as time went on, we found that we had to specify certain key features of the object being annotated in placing the bounding boxes to obtain the necessary level of annotation consistency. Hence, a clear and exhaustive set of guidelines was established and followed in order to reduce the intraannotator variability (the same annotator marking the boxes inconsistently at different times) and interannotator variability (mismatch between different annotators). Further effort was directed to developing a ground-truth markup that was rich with details useful for research, as well as evaluation. Thus, each object block was associated with a set of attributes that characterized the region both from an evaluation and informational point of view. This section explains the set of guidelines and additional flags used in this evaluation for the face, text, and vehicle annotation tasks.

Face ground truth. A face was enclosed by an oriented bounding box. Facial features were used as guides to mark the bounds of this box. If, for any reason, the features were obstructed, then the box was approximated. A face was defined to be VISIBLE in the scene as long as one eye, part of the nose, and part of the mouth were seen. Each face bounding box had additional metadata indicating how clear the face was (AMBIGUITY), whether it was a real face or SYNTHETIC (cartoon, for example), whether the face was OCCLUDED with another object in the scene, and whether a person was wearing hat/sunglasses (HEADGEAR). The AMBIGUITY attribute had three possible values ("0" when all of the three features were clearly seen, "1" when two of the three features were visible, and "2" when none of the three features was visible). This set of attributes made the annotations rich and could be used for subscoring in the evaluations. A sample annotation for face is shown in Fig. 1.

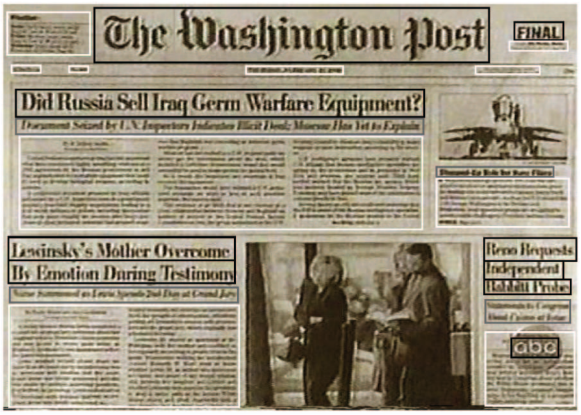

The Washington Post
FINAL
Did Russia Sell Iraq Germ Warfare Equipment?
Lewinsky's Mother Overcome By Emotion During Testimony
Reno Requests Independent Babbitt Probe
abc

Fig. 2. Sample annotation for text. READABILITY = 2 is marked by black boxes. READABILITY = 1 is marked by gray boxes. READABILITY = 0 is marked by white boxes.

Each face bounding box had a unique ID and was maintained for the entire clip as long as the face existed without any temporal breaks. If a face object exited the clip at frame n and came back in frame m, then a new ID was given to this face. The annotator was therefore performing a detection/tracking task and not a recognition task.

Text ground truth. Every new text area was marked with a box when it appeared in the video. The box was moved and scaled to fit the text as it moved in successive frames. This process was done at the text line level until the text disappeared from the frame. As with face objects, each block maintained the same ID to make tracking evaluation possible. A sample text annotation is shown in Fig. 2.

Text objects boxes had several metadata tags to support evaluation at varying levels of task difficulty. Two types of text were distinguished and tagged:

- *Overlay text* is text superimposed onto the video frame. Example, the "abc" logo in Fig. 2.
- *Scene text* is text in the background/foreground of what was actually being filmed. Example, all text regions on the newspaper in Fig. 2.

Text was annotated for *readability* by humans at one of the following three levels:

- Completely unreadable text was tagged as READABILITY = 0 (white boxes in Fig. 2) and was defined as text in which no character was identifiable.
- Partially readable text was tagged as READABILITY = 1 (gray boxes in Fig. 2) and contained characters where only some of them were identifiable.
- Clearly readable text was tagged as READABILITY = 2 (black boxes in Fig. 2) and was used for text in which all letters were identifiable. These blocks of text were *transcribed* for use in future evaluations (e.g., *text recognition*).

The OCCLUSION attribute was set to TRUE when the text was cut off by the bounds of the frame or by another object. The LOGO attribute was set to TRUE when the text region being marked was a company logo imprinted in stylish fonts. Example, the texts "The Washington Post" and "abc" in Fig. 2 had the LOGO attribute set to TRUE.

Of all of the objects of interest in video, text is particularly difficult to uniformly bound in an image because of the size variability of characters even within a single word. For this reason, text regions were tagged based on a comprehensive set of rules as follows:

- All text within a selected block must contain the same readability level and type.
- Blocks of text must contain the same size and font. Two allowances are given to this rule: A different font or size may be included in the case of a unique single character and the font color may vary among text in a group.
- The bounding box should be tight to the extent that there is no space between the box and the text. The maximum distance from the box to the edge of bounded text may not exceed half the height of the characters when READABILITY = 2 (clearly readable). When READABILITY = 0 or 1, the box should be kept tight but does not require separate blocks for partial lines in a paragraph.
- Text boxes may not overlap other text boxes unless the characters themselves are superimposed atop one another.

Vehicle ground truth. For vehicle objects, each vehicle was marked by a nonoriented bounding box. Unlike the face and text tasks, the objective of this task was to detect and track only the vehicles that were moving. The marking for a vehicle object was initialized only when it began to move in the sequence. A vehicle that was completely stationary in the entire sequence was not bounded at all. Therefore, if an algorithm detected a stationary vehicle, it would be treated as a false positive. Vehicle objects were annotated with the following set of attributes:

- LOCATION. Coordinates of the bounding box.
- PRESENT. TRUE if vehicle was visible in the frame and FALSE otherwise.
- OCCLUSION. TRUE if vehicle was occluded and FALSE otherwise.
- MOBILITY. Marked MOBILE if vehicle was moving and STATIONARY if standing still or parked.
- AMBIGUITY. TRUE if image region with vehicles was difficult to annotate and FALSE otherwise.

A vehicle was considered to be PRESENT when at least 25 percent of the vehicle body was visible. If more than 50 percent of the vehicle was occluded, the vehicle had the OCCLUSION attribute set to TRUE. In a situation when the vehicle body was cut off by the camera view, the OCCLUSION attribute was set to FALSE.

One of the important attributes is the MOBILITY attribute, which was used to describe the motion of a vehicle. When a vehicle was moving, MOBILITY was set to MOBILE and, when a vehicle that was initially moving came to a stop at a particular spot, MOBILITY was set to STATIONARY.

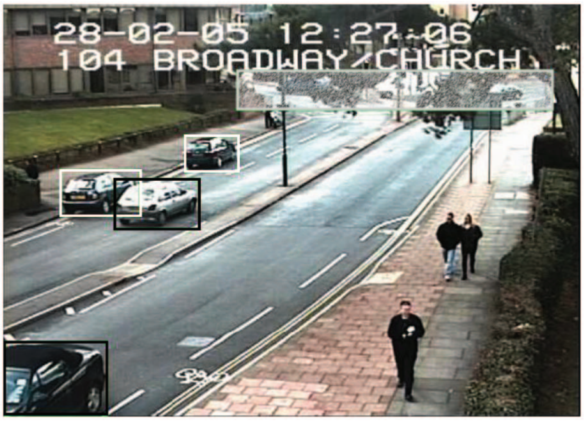


Fig. 3. Sample annotation for vehicle. MOBILITY = MOBILE is marked by black boxes. MOBILITY = STATIONARY is marked by white boxes. AMBIGUITY = TRUE is denoted by the shaded region.

The AMBIGUITY attribute was included to handle extreme occlusion and confusing situations. For example, in a region where there were many cars that were occluded by trees or when the camera view was not clear, this flag was set to TRUE. This attribute was used to exclude these areas from evaluation. It has to be noted that, while other attributes are at an object level, the AMBIGUITY attribute is used for regions. A sample annotation for the moving vehicle task is shown in Fig. 3.

2.3 Annotation Quality

When evaluation relies on manual labeling, it is important to empirically evaluate the consistency of the annotation. While high levels of consistency are not necessary for differentiating performance of immature technologies, the degree of consistency becomes increasingly important as systems approach human levels of performance. We understood that a high degree of consistency would be difficult to achieve with somewhat subjective attributes like readability. However, there was also a surprising amount of subjectivity in determining the spatial bounds of a complex object such as a person or a vehicle. Furthermore, humans fatigued easily when performing such tedious tasks. Therefore, a method was needed to check for human errors in ground-truth generation. For this reason, 10 percent of the entire corpus was doubly annotated by multiple annotators and checked for quality using the evaluation measures. Using the thresholded approach described in Sections 3.2.1 and 3.2.2, we found that at 60 percent spatial threshold, the average *Sequence Frame Detection Accuracy* ($SFDA$) and the average *Average Tracking Accuracy* (ATA) scores for the doubly annotated corpus on the face detection and tracking task were 0.95 and 0.85, respectively. Similarly, for the text tasks, these numbers were 0.97 and 0.90. The scores for the current state-of-the-art automatic algorithms (see Section 4 for details) are significantly lower than these numbers (28 percent relative for face tracking, 22 percent relative for text detection, and 61 percent relative for text tracking). However, the relative difference between the performance of humans and the state-of-the-art algorithm was low (4 percent) for the face detection task. This indicates that the performance of face

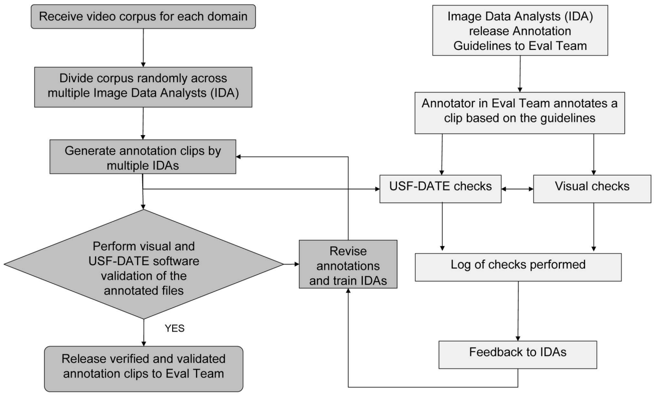

Receive video corpus for each domain
Divide corpus randomly across multiple Image Data Analysts (IDA)
Generate annotation clips by multiple IDAs
Perform visual and USF-DATE software validation of the annotated files
YES
Release verified and validated annotation clips to Eval Team
Image Data Analysts (IDA) release Annotation Guidelines to Eval Team
Annotator in Eval Team annotates a clip based on the guidelines
USF-DATE checks
Visual checks
Revise annotations and train IDAs
Log of checks performed
Feedback to IDAs

Fig. 4. Flowchart of annotation quality control procedure. Steps denoted by dark shaded boxes were carried out by the annotators. Steps denoted by light shaded boxes were carried out by the evaluators.

detection algorithms for the broadcast news domain is nearing the performance of humans.[3]

The process of arriving at the above numbers on the double annotation was iterative and provided significant information about the systematic misunderstandings of the annotation guidelines. These representative errors were logged and checked for on the remaining 90 percent of the singly annotated ground truth through a quality control process that employed both manual and automatic testing. These checks were performed independently from the annotators. Fig. 4 shows a flowchart of the annotation quality control procedure followed in this evaluation. This process assured that the reference annotations were reliable enough for an informative evaluation.

Tools used for annotations. The Video Performance Evaluation Resource (ViPER) [33] was developed as a tool for ground-truthing video sequences and was used to create the reference annotations for this evaluation. Objects were marked by both oriented and nonoriented bounding boxes. The objects were annotated in XML format following the ViPER schema.[4] The ground-truth annotation instructions for all of the tasks can be found in the companion annotation guidelines document.

3 Performance Measures

A significant challenge in scoring spatially corresponding bounding boxes is in finding an appropriate mapping between the system output objects and ground-truth reference. Such a mapping is needed for both the spatial (detection) and temporal (tracking) dimensions for precise and consistent computation of errors such as misses, false positives, and ID switches (in tracking). Longstanding evaluations in automatic speech recognition and speaker segmentation [34] employ a mapping strategy to match candidate system output objects to ground truth objects for scoring. These approaches employ an error minimization algorithm to find the most forgiving mapping with regard to the metric. The mapping is therefore tightly coupled with the metric. In developing the performance measures for detection and tracking evaluation, we extended the time-based mapping strategies used in speech technology evaluation to the spatiotemporal domain of computer vision technology.

Since all of our reference annotations adopted a bounding-box/spatial approach, the performance measures were defined so as to be area-based and dependent on the spatial overlap between the ground truth and the system output objects. This generalization facilitated consistence of scoring of the variety of object detection and tracking algorithms that the VACE set out to evaluate.

In the following sections, we describe the one-to-one mapping that generates the most forgiving performance scores for an algorithm (Section 3.1), the VACE metrics that provide a comprehensive picture of system performance through a single score (Section 3.2), and the CLEAR metrics that assist the algorithm developers in debugging their system by splitting the scores for the accuracy and the precision aspects of performance (Section 3.3).

3.1 One-to-One Correspondence between Ground-Truth and System Output Objects

The optimal matching problem was set up by computing the metric score for every combination of ground truth and system output pair. In our case, we used the spatial overlap between ground-truth objects and system output objects for detection evaluation and the spatiotemporal overlap between ground-truth tracks and system output tracks for tracking evaluation. This resulted in a matrix (N × M; N ground-truth objects, M system output objects) of metric scores where, an entity x_{ij} denoted the metric score for the output object, D_j, when mapped with the reference object, G_i.

$$\begin{array}{c|cccc} & D_1 & D_2 & \dots & D_M \\ \hline G_1 & x_{11} & & & \\ G_2 & & & & x_{2M} \\ \vdots & & & & \\ G_N & & x_{N2} & & \end{array}$$

The maximum score is obtained by optimally assigning ground truth and system output pairs. A brute force exhaustive search would have a prohibitive complexity, a result of having to try out every possible combination of matches (n!). However, in this work, we reduced the computational complexity by implementing a well understood optimization algorithm. We used the Hungarian algorithm [35], which is a numerical search algorithm that guarantees arriving at one optimal solution. The basic algorithm has a series of steps, which are followed iteratively, and has a polynomial time complexity; specifically, some implementations are $O(N^3)$. Faster implementations have been known to exist and, to the best of our knowledge, the current best bound is $O(N^2 logN + NM)$ [36]. There are many variations of the basic strategy, most of which exploit constraints from the specific problem domains they consider. In our case, since the spatial overlap matrix between the

[3] Because of the uncertainty in the ground-truth annotations, it is difficult to discern small differences in the system performance with statistical certainty at this level of accuracy. As system scores approach that of the ground truth they are measured against, it is possible for real differences in performance to be overwhelmed by the "noise" in the ground truth. Therefore, one should be cautioned to determine the statistical significance of score differences before making conclusions regarding algorithmic improvements for face detection in this domain.

[4] http://viper-toolkit.sourceforge.net/owl/viper/datatypes/.

reference objects and the system objects was mostly sparse, we took advantage of it by implementing a hash function for mapping subinputs from the whole set of inputs. More details about the assignment problem and its solution can be found in standard textbooks [37], [38], [39].

In usual assignment problems, the number of elements in both the sets should be equal, i.e., when $N = M$. In detection and tracking evaluation, it is most likely that the number of ground-truth objects is not equal to the number of detected objects. However, this was fixed easily and was, in fact, used to compute the errors. To begin with, we added imaginary objects to the set with smaller number of elements. The spatial overlap between any imaginary object and all objects in the other set was assigned a value of 0. This way, during the assignment process, the imaginary objects would not compete with any of the real objects for any object in the other set. Once the assignment was complete, all objects (including the ones that were imaginary) were associated with objects in the other set. All of those objects that were matched with imaginary objects were considered unmatched and therefore were either misdetected ground-truth object (if imaginary objects were added to the detected object set) or false positives (if the imaginary objects were added to the ground truth object set). Thus, though the one-to-one mapping between the ground truth and system output objects was primarily designed to compute the maximum score for an algorithm's performance, in the process, we also used it to determine the misses and the false positives by an algorithm.

3.2 VACE Metrics

In this section, we describe the two comprehensive metrics (one each for detection and tracking) developed during VACE Phase II. These measures account for important measures of system performance (number of objects detected and tracked, missed objects, false positives, fragmentation in both spatial and temporal dimensions, and localization error of detected objects) in a single score. These were the primary metrics used to score algorithm performance in the VACE Phase II evaluation tasks.

The following are the notations used in the remainder of the paper:

- G_i denotes the ith ground-truth object at the sequence level and $G_i^{(t)}$ denotes the ith ground-truth object in frame t.
- D_i denotes the ith detected object at the sequence level and $D_i^{(t)}$ denotes the ith detected object in frame t.
- $N_G^{(t)}$ and $N_D^{(t)}$ denote the number of ground-truth objects and the number of detected objects in frame t, respectively.
- N_G and N_D denote the number of unique ground-truth objects and the number of unique detected objects in the given sequence, respectively. Uniqueness is defined by object IDs.
- N_{frames} is the number of frames in the sequence.
- N_{frames}^i is the number of frames in which the ground-truth object (G_i) or the detected object (D_i), depending on the context, existed in the sequence.
- N_{mapped} is the number of mapped ground truth and detected object pairs when the mapping is done at the sequence level and $N_{mapped}^{(t)}$ is the number of mapped ground truth and detected object pairs in frame t (frame-level mapping).

3.2.1 Sequence Frame Detection Accuracy $(SFDA)$

The Sequence Frame Detection Accuracy $SFDA$ is a comprehensive frame-level measure that accounts for the number of objects detected, missed detects, false positives, and spatial alignment of system output and ground-truth objects.

For a given frame, the Frame Detection Accuracy (FDA) measure calculates the spatial overlap between the ground truth and system output objects as a ratio of the spatial intersection between the two objects and the spatial union of them. The sum of all of the overlaps was normalized over the average of the number of ground truth and detected objects. For a single frame t, where there are $N_G^{(t)}$ ground-truth objects and $N_D^{(t)}$ detected objects, we defined $FDA(t)$ as

$$\begin{aligned} FDA(t) &= \frac{\text{Overlap\_Ratio}}{\left[\frac{N_G^{(t)}+N_D^{(t)}}{2}\right]}, \\ \text{where Overlap\_Ratio} &= \sum_{i=1}^{N_{mapped}^{(t)}} \frac{\left|G_i^{(t)} \bigcap D_i^{(t)}\right|}{\left|G_i^{(t)} \bigcup D_i^{(t)}\right|}. \end{aligned} \tag{1}$$

Here, $N_{mapped}^{(t)}$ is the number of mapped object pairs in frame t, where the correspondence is established between objects that have the best spatial overlap in the given frame t using the method described in Section 3.1.

In order to measure the detection performance for the whole sequence, the FDA was calculated over all of the frames in the sequence and normalized to the number of frames in the sequence where at least a ground truth or a detected object existed. This way of normalization accounted for both missed detects and false positives. We thus obtained the $SFDA$, which is essentially the average of the FDA measure over all of the relevant frames in the sequence. This is expressed as

$$SFDA = \frac{\sum_{t=1}^{t=N_{frames}} FDA(t)}{\sum_{t=1}^{t=N_{frames}} \exists\left(N_G^{(t)}\ OR\ N_D^{(t)}\right)}. \tag{2}$$

Relaxing localization errors. To forgive minor inconsistencies in the localization of spatial boundaries between the system output and the ground-truth objects, we implemented a thresholding parameter based on a heuristic analysis of the reference annotations. For a given application, the threshold was derived from spatial disagreements between the annotators in the 10 percent double annotated data. The motivation behind this was to eliminate the error in the scores due to ground-truth inconsistencies in terms of bounding box location and size. In addition, such an approach of arriving at the spatial threshold reflected the difficulties in how humans perceived the task. The analysis resulted in determining that at least 20 percent of the system output object should spatially overlap the reference object to indicate correct detection. Lower thresholds were

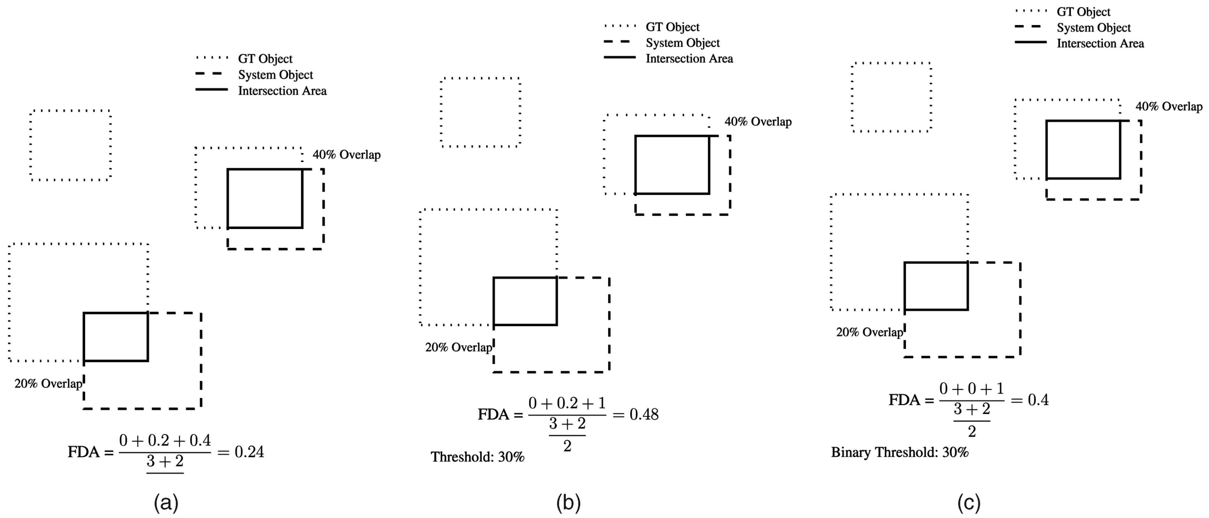


Fig. 5. Sample example illustrating the various styles of FDA score computation. (a) No thresholding. (b) Nonbinary thresholding. (c) Binary thresholding.

determined to forgive too many real detection errors and higher thresholds were determined to penalize too many correct detections.

To make the explanations easier, we present here a series of comparative examples, where the $SFDA$ scores are computed subject to different thresholding options. Fig. 5a shows an example on a particular frame. There are three ground-truth boxes and two system output boxes. The FDA scoring and the individual scoring are indicated below in the same figure.

The thresholding operation is defined in (3) and (4). Figs. 5b and 5c show an example with these styles of thresholding:

$$\text{Thresholded Overlap Ratio}_i^{(t)} = \frac{FDA\_T_i^{(t)}}{\left|G_i^{(t)} \cup D_i^{(t)}\right|}, \tag{3}$$

where

$$FDA\_T_i^{(t)} = \begin{cases} \left|G_i^{(t)} \cup D_i^{(t)}\right|, & \text{if } \frac{\left|G_i^{(t)} \cap D_i^{(t)}\right|}{\left|G_i^{(t)} \cup D_i^{(t)}\right|} \geq Threshold, \\ \left|G_i^{(t)} \cap D_i^{(t)}\right|, & \text{if } \frac{\left|G_i^{(t)} \cap D_i^{(t)}\right|}{\left|G_i^{(t)} \cup D_i^{(t)}\right|} < Threshold \\ & \quad \& \text{ nonbinary thresholding}, \\ 0, & \text{if } \frac{\left|G_i^{(t)} \cap D_i^{(t)}\right|}{\left|G_i^{(t)} \cup D_i^{(t)}\right|} < Threshold \\ & \quad \& \text{ binary thresholding}. \end{cases} \tag{4}$$

3.2.2 Average Tracking Accuracy (ATA)

For our purposes, tracking consisted of identifying and labeling detected objects across contiguous frames. The task was similar to detection, with detected objects linked by a common identity (object IDs) across frames. Therefore, objects that left the scene and returned later in the sequence were not identified as the same object. However, occluded objects were treated as the same object, but tracking was optional in frames where there was occlusion.

Unlike detection, this was a spatiotemporal task, yet its performance could be assessed with a measure similar to the $SFDA$ measure described in Section 3.2.1. The significant difference between the measures was that, in detection tasks, the mapping between the system output and reference annotation objects was optimized on a frame-by-frame basis, whereas, for tracking, the mapping was optimized at a sequence level. One of the advantages of making this task highly parallel to the detection task was that the $SFDA$ measure could also be applied to the tracking output to quantify the performance degradation due to misidentification of objects across frames.

The Average Tracking Accuracy (ATA) is a spatiotemporal measure that penalizes fragmentations in both the temporal and spatial dimensions while accounting for the number of objects detected and tracked, missed objects, and false positives.

A one-to-one mapping between the ground truth and the system output objects was established by computing the measure over all of the ground truth and detected object combinations and using an optimization strategy to maximize the overall score for the sequence (Section 3.1). The Sequence Track Detection Accuracy $(STDA)$ was then calculated as

$$STDA = \sum_{i=1}^{N_{mapped}} \frac{\sum_{t=1}^{N_{frames}} \left[\frac{\left|G_i^{(t)} \cap D_i^{(t)}\right|}{\left|G_i^{(t)} \cup D_i^{(t)}\right|}\right]}{N_{(G_i \cup D_i \neq \emptyset)}}. \tag{5}$$

Analyzing the numerator of (5), we observe that it is merely the overlap of the detected object over the ground truth, which is very similar to (1). The only difference is that, in tracking, we measured the overlap in the spatiotemporal dimension, while, in detection, the overlap was in the spatial dimension alone.

The $STDA$ is a measure of tracking performance over all of the objects in the sequence. It can take a maximum value of N_G, which is the number of ground-truth objects in the sequence. We defined the ATA, which can be termed as the $STDA$ per object, as

$$ATA = \frac{STDA}{\left[\frac{N_G + N_D}{2}\right]}. \tag{6}$$

In cases when it was desirable to measure the tracking aspect of the algorithm and not be concerned with the detection accuracy, we relaxed the detection penalty by using an area thresholded approach (see *Relaxing Localization Errors* in Section 3.2.1).

3.3 CLEAR Metrics

When the VACE and the CHIL programs decided to collaborate and form the CLEAR program, it was agreed that, in order to harmonize the common evaluation tasks, the metrics had to be harmonized first. A strong need was felt for a set of unified metrics that was widely accepted by both of the programs. This was necessary to provide the basis of discussion and exchange between the programs.

The comprehensive metrics, $SFDA$ and ATA, provided a single score comparison between systems and are useful for end users and optimization-based development strategies. However, some researchers viewed these as less usable because the measures did not let them identify failure components for debugging purposes. This was more pronounced because the localization error (*precision* of detection) was fused in the comprehensive metric scores.

In this section, we describe a set of four metrics (a pair each for detection and tracking) that was developed through a joint effort among NIST, USF, and the University of Karlsruhe. These measures split the *accuracy* and the *precision* aspects of the system in two separate scores. They were the primary measures used to score algorithm performance in the CLEAR 2006 evaluation tasks.

3.3.1 *Multiple Object Detection Accuracy* ($MODA$)

To assess the *accuracy* aspect of system performance, we utilized the missed detection and false positive counts. Assuming that the number of misses is indicated by m_t and the number of false positives is indicated by fp_t for each frame t, we computed the Multiple Object Detection Accuracy ($MODA$) as

$$MODA(t) = 1 - \frac{c_m(m_t) + c_f(fp_t)}{N_G^{(t)}}, \tag{7}$$

where c_m and c_f are the cost functions for the missed detects and false positives and N_G^t is the number of ground truth objects in the tth frame. c_m and c_f are used as scalar weights and can be varied based on the specific application. For example, if missed detects are more critical than false positives, we can increase c_m and reduce c_f. c_m and c_f were both equal $(= 1)$ in this evaluation.

We computed the Normalized $MODA$ ($N\text{-}MODA$) for the entire sequence as

$$N\text{-}MODA = 1 - \frac{\sum_{t=1}^{N_{frames}} \left(c_m(m_t) + c_f(fp_t)\right)}{\sum_{t=1}^{N_{frames}} N_G^{(t)}}. \tag{8}$$

3.3.2 *Multiple Object Detection Precision* ($MODP$)

We used the spatial overlap information between the ground truth and the system output (similar usage as in (1)) to compute the Mapped Overlap Ratio, as defined in (9):

$$Mapped\ Overlap\ Ratio = \sum_{i=1}^{N_{mapped}^{(t)}} \frac{\left|G_i^{(t)} \bigcap D_i^{(t)}\right|}{\left|G_i^{(t)} \bigcup D_i^{(t)}\right|}, \tag{9}$$

where $G_i^{(t)}$ denotes the ith ground-truth object in the tth frame, D_i^t denotes the detected object for G_i^t, and N_{mapped}^t is the number of mapped object pairs in frame t.

Using the assignment sets, the Multiple Object Detection Precision ($MODP$) for each frame t was computed as

$$MODP(t) = \frac{(Mapped\ Overlap\ Ratio)}{N_{mapped}^{(t)}}. \tag{10}$$

This gave us the precision of detection in any given frame and we normalized the measure by taking into account the total number of relevant evaluation frames. If $N_{mapped}^t = 0$, then $MODP$ was forced to a value for zero for that frame. The Normalized $MODP$ ($N\text{-}MODP$) that gives the detection precision for the entire sequence was computed as

$$N\text{-}MODP = \frac{\sum_{t=1}^{N_{frames}} MODP(t)}{N_{frames}}. \tag{11}$$

Stiefelhagen et al. [40] provide a comparison between $SFDA$ and $MODA/MODP$.

3.3.3 *Multiple Object Tracking Accuracy* ($MOTA$)

To extract the accuracy aspect of the system output track, we computed the number of missed detects, false positives, and switches in the system output track for a given reference ground-truth track. Multiple Object Tracking Accuracy ($MOTA$) was defined as

$$MOTA = 1 - \frac{\sum_{t=1}^{N_{frames}} \left(c_m(m_t) + c_f(fp_t) + c_s(ID\text{-}SWITCHES_t)\right)}{\sum_{t=1}^{N_{frames}} N_G^{(t)}}, \tag{12}$$

where, after computing the mapping for frame t, m_t is the number of misses, fp_t is the number of false positives, and $ID\text{-}SWITCHES_t$ is the number of ID mismatches in frame t considering the mapping in frame $(t-1)$. Therefore, during tracking, if there was a track split or merge, we would still consider the contribution of the new track but penalized it by counting it as an $ID\text{-}SWITCH$. The values used for the weighting functions in this evaluation were $c_m = c_f = 1$ and $c_s = \log_{10}$. We started the $ID\text{-}SWITCH$ count from 1 because of the log function.

3.3.4 *Multiple Object Tracking Precision* ($MOTP$)

To obtain the precision score, we calculated the spatiotemporal overlap between the reference tracks and the system output tracks. It is worth noting that, while computing the precision score, we accounted for the contribution of split and merged tracks as well. The Multiple Object Tracking Precision ($MOTP$) was defined as

$$MOTP = \frac{\sum_{i=1}^{N_{mapped}} \sum_{t=1}^{N_{frames}^{(t)}} \left[\frac{\left|G_i^{(t)} \cap D_i^{(t)}\right|}{\left|G_i^{(t)} \cup D_i^{(t)}\right|}\right]}{\sum_{t=1}^{N_{frames}} N_{mapped}^{(t)}}, \tag{13}$$

where N_{mapped} refers to the mapped system output objects over an entire reference track taking into account splits and merges and N^t_{mapped} refers to the number of mapped objects in the $t\text{th}$ frame.

Stiefelhagen et al. [40] provide a comparison between ATA and $MOTA/MOTP$.

4 Baseline Algorithms

This section describes the baseline algorithms along with the results on their respective testing sets. These baselines are provided as a reference for future performance comparison.

4.1 Face Detection and Tracking

Detecting faces in images is an essential step to intelligent vision-based human-computer interaction. It is the first step in many research efforts in face processing, including face recognition, pose estimation, and expression recognition. Numerous techniques have been proposed to detect faces in images and video [41], [42].

The baseline face detection algorithm we present here was one of the biggest milestones in the area of real-time face processing in video. It was the work of Viola and Jones [43] that was later extended by Lienhart and Maydt [44].

Detection. The Viola-Jones face detector detects close-to-frontal horizontally aligned faces using a cascade of pretrained classifiers of increasing complexity applied to rectangular Haar-like binary wavelet features efficiently computed from video frames using the "integral image" preprocessing technique. More details of the algorithm can be found in [43], [44].

The Intel Open Source Computer Vision Library (OpenCV) [45] implementation (Version 1.0rc1) of the Haar face detection algorithm was used as the baseline for this task. The face detector was *not* trained on the training set of the Broadcast News corpus. Instead, the trained classifier cascade available with the OpenCV distribution was used.

There were three user selectable parameters for the OpenCV implementation of the Haar face detector as follows:

- The scaling factor—specified the factor by which the search window is scaled in a subsequent iteration.
- The grouping factor—signified the minimum number of neighboring face rectangles that should be joined into a single "face." Smaller groups were rejected for a lack of sufficient evidence.
- *The preprocessing flag*—when set, the algorithm worked with edge data (Canny edge detector [46]) instead of the raw data. This made the algorithm run faster due to the reduction in information.

The optimal values for each of these parameters were found using an experimental design strategy [47] described in Appendix A. The final set of values was scaling factor = 1.2, grouping factor = 3, and preprocessing flag = TRUE.

Tracking. The tracking algorithm used in our face and vehicle baselines (Section 4.3) used the detected objects in each frame to track them throughout the given sequence. Once an algorithm detected an object, it must be identified with the same object ID across frames to be scored as correct. A distance-based algorithm was used to track the object throughout the given sequence. The method is described in Algorithm 1.

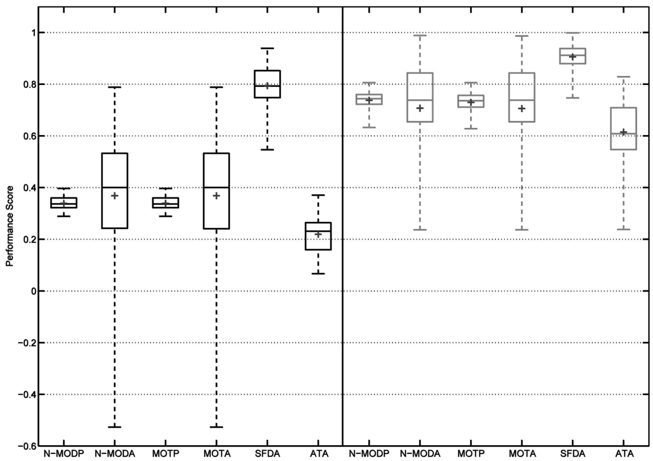
Algorithm 1 *Face and Vehicle Tracking Algorithm.*

```
Require: Three consecutive frames (current, previous, and
    second previous) with the detected objects specified
    (the number of detected objects in current frame is $n$
    and numbers of detected objects in the previous and
    second previous frames are $m_1$ and $m_2$ consecutively);
    two predefined thresholds ($threshold1$ and $threshold2$).
Ensure: The temporal propagated object $IDs$ from the
    previous and second previous frames to the current
    frame;
    new $IDs$ for new objects.
1:  Calculate the centroids $(c_{x_k}, c_{y_k})$, $k = 1, \dots, M$,
    $M = n + m_1 + m_2$, of the detected objects in all three
    frames, using the formulas $c_x = x + \frac{w}{2}$, $c_y = y + \frac{h}{2}$,
    where $w$ is the width of the object, and $h$ is the height of
    the object
2:  Find the minimum euclidean distance $(d_{min_i})$ for
    $i\text{th-detected}$ object of the previous frame to the detected
    objects in the current frame using a greedy approach.

    $d_{min_i} = min_j \sqrt{(c^{pf}_{x_i} - c^{cf}_{x_j})^2 + (c^{pf}_{y_i} - c^{cf}_{y_j})^2}, j = 1, \dots, n,$

    where $(c^{pf}_{x_i}, c^{pf}_{y_i})$ is centroid of the $i\text{th-detected}$ object of
    the previous frame, $(c^{cf}_{x_j}, c^{cf}_{y_j})$ is centroid of the
    $j\text{th-detected}$ object of the current frame;
3:  if $d_{min_i} \leq threshold1$ then
4:      Assign the $ID$ of the previous object to the
        corresponding object of the current frame with the
        minimum distance;
5:      if the minimum distance is the same for two
        different objects in current frame then
6:          Assign $ID$ to the object with the closest area to
            the area of the previous object;
7:      end if
8:  end if
9:  Repeat steps 2-8 $m_1$ times;
10: Repeat the above steps $m_2$ times for the detected
    objects of the second previous frame and the detected
    objects of the current frame, with additional conditions
    (different distance threshold $(threshold2)$ and
    constraint that two or more objects cannot have the
    same $ID$);
11: Assign new $IDs$ to all unassigned objects in the current
    frame.
```

Results. We utilized a graphical method of display called *boxplots* to present the performance scores of an algorithm. Boxplots provide an efficient way to summarize statistical measures such as median, upper and lower quartiles, and minimum and maximum data values. The box contains the middle 50 percent of the data. The upper edge of the box indicates the 75th percentile of the data set and the lower edge indicates the 25th percentile. The horizontal line and the "+" within the box represent the median and the mean of the data samples, respectively. The extremities of the dotted line outside the box signify minimum and maximum data values. Boxplots have the advantage that they display a variable's location and spread at a glance. The most important advantage is that, by using a boxplot for each



(a) (b)

Fig. 6. Comparing the performance of the baseline algorithm with a state-of-the-art face detection and tracking algorithm on the broadcast news corpus for several different metrics defined in Sections 3.2 and 3.3 (+ indicates mean of the corresponding performance metric score). The state-of-the-art algorithm was the best performing system in the VACE Phase II evaluations. (a) Baseline algorithm. (b) State-of-the-art algorithm.

system side-by-side on the same graph, one can quickly compare algorithms. When the boxes of two algorithm scores overlap one another, it suggests that the two algorithms are similar in their performance on the given data set.

Fig. 6a shows the boxplots of the detection and tracking scores for the baseline algorithm. For detection, the specific measures that we observed were the $N\text{-}MODP$, $N\text{-}MODA$, and $SFDA$ whose mean values were 0.34, 0.37, and 0.79. While the $SFDA$ measures the detection performance comprehensively and had a high score, the $N\text{-}MODP$ and $N\text{-}MODA$ were lower. Specifically, the $N\text{-}MODP$ showed that the overall spatial overlap (on the mapped objects) was 0.34 and on some clips as high as nearly 0.40. The $N\text{-}MODA$, which shows how accurate the counts of the objects are, measured 0.37 overall and, on some clips, as high as 0.80. It has to be noted that the $SFDA$ score was thresholded at 20 percent spatial overlap. For the tracking performance, the specific measures we used were $MOTP$, $MOTA$, and ATA. The $MOTP$ and $MOTA$ were similar to their detection counterparts but were slightly lower. The average ATA score was 0.22.

Fig. 6b compares the performance of the baseline algorithm against a state-of-the-art face detection and tracking algorithm on this data set. Algorithm results for face detection and tracking in boardroom meeting videos using the $SFDA/ATA$ metrics were presented in [2]. From the plots in Fig. 6, it can be observed that the performance of the baseline algorithm is comparable to the state of the art, which makes it an appropriate choice for this task.

4.2 Text Detection and Tracking

Text embedded in video frames often carries important information such as time, place, name, topics, and other relevant information. These semantic cues can be used in video indexing and video content understanding. To extract textual information from video, which is often referred to as *Video Optical Character Recognition*, the first essential step is to detect and track the text region in the video sequence. There are several published efforts addressing the problem of text area detection in video [48].

The baseline algorithm that we present here was the work of Crandall et al. [49]. The approach consisted of three steps.

4.2.1 Detection and Localization

Detection was accomplished by analyzing local texture features using 8 × 8 blockwise Discrete Cosine Transform (DCT) and finding blocks of the image that had texture consistent with text. For each block, the absolute values of a subset of the DCT coefficients were computed and regarded as the *text energy* of that block.

Once individual blocks of a frame had been classified, the text instances were grouped by finding the minimum bounding rectangles using an iterative procedure. After this, heuristics based on rectangle dimensions were applied to discard nontext regions. We also discarded rectangles whose length or width was less than 8 pixels.

4.2.2 Binarization

Binarization is the process of separating character strokes from the background by classifying individual pixels as text or background. A series of steps was followed to achieve this starting with the following:

- *Preprocessing.* Where, if the localized text region was slanted, a rotation transformation was applied to make it horizontal. A linear interpolation step was used to double the resolution, after which a gray-level histogram equalization was performed to improve contrast between text and background.
- *Logical level thresholding.* Where the motivation was to reduce noise in the binary output. This step utilized the perceptually uniform $L^*a^*b^*$ color space.
- *Character candidate filtering.* Where connected component analysis was performed on both the output images from the previous steps.
- *Choice of binarization polarity.* Where, as noted earlier, the logical level thresholding was applied on both the original localized text region and its inverse. When the average gray-scale value of the text was higher than the background, logical level thresholding binarized the background and, when the average gray-scale value of the text was lower than the background, it resulted in the binarization of the text.

4.2.3 Tracking

We used the motion vectors of MPEG-compressed video to predict text motion with very little computational cost to the tracker. In effect, the computation cost had already been paid by the MPEG encoder.

Given a localized text region in one frame, the next frame was searched for all macroblocks whose motion vectors pointed back to any part of the text region. Several constraints were then applied to the motion vectors to select only those that were likely to be reliable. Very small motion vectors (less than 2 pixels in magnitude) were likely to be noise and were therefore ignored. Motion vectors from relatively featureless macroblocks were also discarded

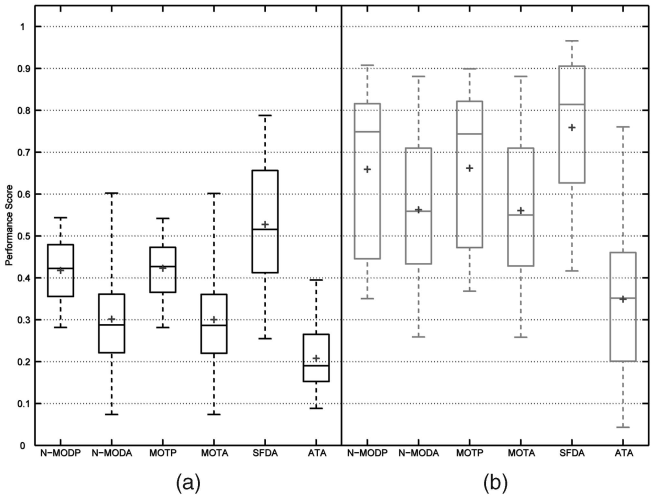


Fig. 7. Comparing the performance of the baseline algorithm with a state-of-the-art text detection and tracking algorithm on the broadcast news corpus for several different metrics defined in Sections 3.2 and 3.3 (+ indicates mean of the corresponding performance metric score). The state-of-the-art algorithm was the best performing system in the VACE Phase II evaluations. (a) Baseline algorithm. (b) State-of-the-art algorithm.

because they were not likely to be accurate. This was determined by applying a Sobel edge detector on each macroblock and eliminating macroblocks that contained less than four edge pixels. The magnitude and direction of the remaining motion vectors were then clustered. The vectors in the largest cluster, which corresponds to the approximate motion of the text block, were then averaged to yield a single motion vector for the text region.

Results. Fig. 7a shows the baseline text detection and tracking performance in the broadcast news domain. For both the detection and the tracking scores, the precision was low. This was a result of overestimation of the bounding box size by the baseline algorithm. The false positives that were introduced by the algorithm were removed by a basic postprocessing, as described in Appendix B. The average baseline algorithm performance for both detection and tracking was 0.42 in terms of precision and 0.30 for accuracy.

Fig. 7b compares the performance of the baseline algorithm against a state-of-the-art text detection and tracking algorithm on this data set. Algorithm results for text detection and tracking on a subset of the test data set using the $SFDA/ATA$ metrics were presented in [1]. The average state-of-the-art algorithm performance for both detection and tracking was 0.66 in terms of precision and 0.56 for accuracy. The baseline algorithm performance showed similar characteristics except that the mean was lower. This further alludes to the issue that the errors introduced were mainly from missed detects. The drop was consistent when we looked at the $SFDA$ and ATA scores as well.

4.3 Vehicle Detection and Tracking

Detecting and tracking vehicles in video data has important ramifications for surveillance and traffic monitoring. In surveillance, by automatically extracting information about the presence of a vehicle, one could focus on the specific parts of a video that contain vehicles, rather than searching a long video for these specific instances. In traffic monitoring, the detection and tracking of vehicles potentially allows for automated solutions to queue management, incident detection and reporting, and traffic flow analysis. At a minimum, the detection and tracking information provided could help an expert find the times and places of interest for further analysis. There are many reported works concerning moving vehicle detection. Sun et al. [50] presented a review of vision-based on-road vehicle detection systems where the camera is mounted on the vehicle. References [51], [52], [53], [54], [55], [56], [57] are some of the recent works on detecting and tracking vehicles from fixed cameras such as in traffic/driveway monitoring systems.

The baseline algorithm described here involved three major steps.

4.3.1 Background Subtraction

In this step, we used a simple running average background learning method. For every pixel in each of the selected video frames, the average background image was updated using the update equation, as shown below:

$$\begin{aligned} BI_{color-channel}(x,y) &= (1-a) \times BI_{color-channel}(x,y) \\ &\quad + a \times CI_{color-channel}(x,y), \end{aligned} \tag{14}$$

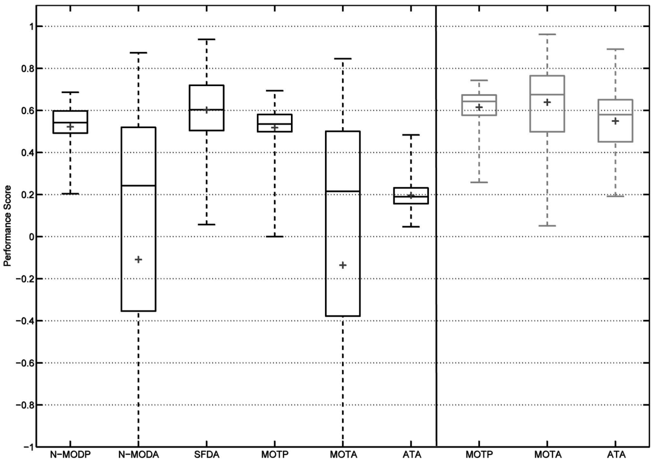
where *color channel* is red, green, and blue, BI indicates accumulated background image, CI indicates current image, $0 \leq a \leq 1$ ($= 0.0004$, in our case) is the learning rate, and (x, y) indicates the 2D image coordinates.

Hardware-level automatic gain control in the video cameras used to collect the source video produces a large global illumination change in the video from time to time. In order to counteract these inconsistencies in the data, we implemented an automatic gain adjust function. It was accomplished by taking the average gray-scale intensity from the initial background image and from the current video image. The difference between these two average gray-scale intensities was added to every pixel in the current video image, thus making the average gray-scale intensity for the current image the same as the initial background image.

Once the background image was learned, the pixels of interest (those which were in motion) were obtained by a simple background subtraction method followed by a binary thresholding ($T_0 = 52$, after manual training) on the difference image.

4.3.2 Foreground Blob Filtering

The resulting binary image obtained from the previous step had noise and included nonvehicle objects. Filtering had to be performed to extract only the vehicles. We made some basic assumptions with regard to vehicles and viewpoints. First, vehicles were assumed to be the largest moving objects in the scene with respect to the 3D coordinates. Second, the camera was assumed to be looking at the scene from above at an angle. Using these two assumptions, the filtering operation should only return the largest blobs from the scene but take into account the 3D view of the scene when determining what constituted a "large" blob. Specifically, objects near the camera will look bigger in comparison to those farther away, so object blobs must be filtered differently depending on their respective positions



(a) (b)

Fig. 8. Comparing the performance of the baseline algorithm with a state-of-the-art vehicle detection and tracking algorithm on the surveillance video corpus for several different metrics defined in Sections 3.2 and 3.3 (+ indicates mean of the corresponding performance metric score). The state-of-the-art algorithm was the best performing system in the CLEAR 2006 evaluations. (a) Baseline algorithm. (b) State-of-the-art algorithm.

in the image. This filtering by size operation was integral to differentiating between vehicle and nonvehicle blobs.

The foreground image was divided into regions by linking all pixels to their adjacent pixels in the 4-neighborhood, where the pixel was considered in the foreground; no linking was done if the pixel was in the background. In this way, blobs of pixels were formed, where they were labeled foreground and connected to at least one other foreground neighbor pixel. Rectangles aligned with the image axes and completely containing the blob were then created.

The threshold value for the size thresholding step was described using a simple quadratic equation, meant to take into account the camera view:

$$T = c \cdot y^2 + d \cdot y + T_0, \tag{15}$$

where y is the y-axis value corresponding to each region rectangle and $c\ (= 0.04)$, $d\ (= 2.3)$, and $T_0\ (= 120)$ are values used for weighting the threshold value based on image location. The values of c, d, and T_0 were obtained through manual training. For the algorithm, y can be chosen to the minimum, maximum, or centroid y value of the region rectangles. The size thresholding operation for vehicle detection was then performed using the total pixel count of the identified object as the size of the blob.

This filtering by size operation and the background subtraction done in the first part of the algorithm together formed the moving vehicle detection step.

4.3.3 Tracking

For tracking the detected vehicle blobs, we used the same algorithm utilized for the face tracking task (Section 4.1).

Results. Fig. 8a shows the performance of the baseline vehicle detection and tracking algorithm in the Surveillance domain. The mean detection and tracking precision scores ($N\text{-}MODP$ and $MOTP$) were both 0.52. The mean detection and tracking accuracy scores ($N\text{-}MODA$ and $MOTA$) were -0.11 and -0.14, respectively. The accuracy score tended to be lower than the precision since false positives tend to be punished harshly by the metrics, but the baseline algorithm strives for a balance between good detections at the expense of having more false positives. The $SFDA$ and ATA were both respectable for a baseline algorithm, 0.60 and 0.20, respectively, when scored with a spatial threshold value of 20 percent.

Fig. 8b compares the performance of the baseline algorithm against a sample state-of-the-art vehicle-tracking algorithm on the test data set.[5] Results from multiple state-of-the-art algorithms for vehicle tracking on this data set were presented in [40]. From the figure, one can see that the scores from the baseline are lower than the scores from the sample algorithm and this was consistent even for individual clips in the test corpus. The mean $MOTP$ of the sample algorithm was 0.10 higher than the baseline algorithm—a significant difference. The accuracy of the sample algorithm was better than the baseline (as noted above, this was mostly due to false positives in the baseline).

5 Protocol

This section describes the USF-DATE scoring tool used for computing the performance by comparing the algorithm output with the annotated ground-truth data. We also present other evaluation logistics such as the evaluation settings used for different tasks and the input/output details. A complete description of the evaluation process, such as the usage of the scoring tool, the details of the ViPER XML format, the attribute tags, and the file naming conventions for both ground truth and system output, is provided in the companion documentation, which can be found in the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10-1109/TPAMI.2008.57.

5.1 USF-DATE Scoring Tool

The *USF Detection and Tracking Evaluation (USF-DATE)* tool was completely developed in C++ using a gcc compiler version 3.4.2. All of the metrics described earlier ($SFDA$, ATA, $MODA$, $MODP$, $MOTA$, and $MOTP$) were supported by the tool. In addition to the metrics, the tool computed a set of auxiliary information such as the IDs of the ground-truth objects that were missed, the system output objects that were treated as false positives, and the spatial overlap between mapped object pairs on a per frame basis. A visualization option superimposed the reference annotation and the system output on the original video, which helped in a cursory analysis of the performance. These features aided the developers in investigating the failure instances of their algorithm for debugging purposes.

5.2 Evaluation Settings

The reference was richly annotated with a variety of information, as described in Section 2.2. The additional set of attributes was used in deciding whether a particular object should be evaluated or not. The specific settings used for the

[5] CLEAR 2006 evaluations particularly measured the vehicle tracking performance. Hence, we compare the results of the baseline with the tracking scores alone.

TABLE 3
Evaluation Settings for Face, Text, and Vehicle Tasks

TASK	EVALUATION SETTINGS

Face Detect & Track	VISIBLE = TRUE
	AMBIGUITY = 0
	SYNTHETIC = FALSE
	OCCLUDED = FALSE
	HEADGEAR = FALSE

Text Detect & Track	TEXT-TYPE = Overlay
	READABILITY = 2
	OCCLUSION = FALSE
	LOGO = FALSE

Vehicle Detect & Track	PRESENT = TRUE
	OCCLUSION = FALSE
	MOBILITY = MOBILE
	AMBIGUITY = FALSE

face, text, and vehicle tasks are shown in Table 3. These settings were used to get the performance scores of both the baseline and the state-of-the-art algorithms (Section 4).

All other annotated regions were treated as "Don't Care," where the system output was neither penalized for missing nor given credit for detecting the unscored region. It has to be noted that each of these attributes can be selectively specified to be included in evaluation through the scoring tool USF-DATE.

5.3 System Input/Output

The evaluation source video was in multiple sequences varying in duration from 1-4 minutes. The corresponding ground-truth annotations were in XML format with the attributes specified in the task definitions (Section 2.2). The system output was also to be in XML format conforming with the ViPER schema to be scored by the scoring tool USF-DATE. The algorithm had to emit the information specified in the task definitions (Section 2.2).

6 Conclusions

We presented a framework for evaluating the performance of object detection and tracking algorithms for face, text, and vehicle in video. This work employs a significantly large development and evaluation corpus that can be used to support machine learning approaches and statistically differentiate differences in system performance. The corpus includes extensive human-annotated ground truth that has been calibrated for consistency. A set of comprehensive metrics was presented, which permit system performance differences to be discerned via a minimal number of measures. We provided a balanced set of measures, where comprehensive comparisons can be made with particular measures ($SFDA/ATA$), and failures can be analyzed by splitting the scores for accuracy and precision ($MODA/MODP$ and $MOTA/MOTP$). The approach presented in this paper supports direct measurement of detection and tracking technologies, facilitates iterative algorithm development, and provides important diagnostic feedback.

This framework includes the necessary infrastructure (source video, task definitions, metrics, ground truth, and scoring tools) to perform formal evaluations of face, text, and vehicle detection and tracking tasks. In addition, we furnished baseline algorithms for these specific tasks and measured their performance over the entire test set of 50 clips in their respective domains.

This set of data and tools will provide a lasting resource for research and objective comparison of approaches and algorithmic improvements. Moreover, the availability of common detection and tracking data sets and metrics permits developers to directly compare the performance of their algorithm with other approaches. This will speed the adoption of successful approaches, minimize redundant research efforts, and accelerate progress. A further benefit to such a framework is that the current state-of-the-art can be reliably calibrated and tracked over time. In conclusion, we believe that the videos, reference annotations, performance metrics, evaluation protocol, scoring software, and baseline algorithms provide researchers an invaluable resource to advance research on the topic of object detection and tracking.

Appendix A
Face Training

This section describes an experimental design technique [47] used to identify the optimal values for the three user-selectable parameters of the OpenCV implementation of the Haar face detection algorithm used in this paper (Section 4.1).

Our task in this process was to analyze the following aspects of the algorithm:

1. the optimal parameter values that would maximize the performance,
2. the significance of each parameter, and
3. the correlation between parameters (called as factor *interaction* in statistical learning).

An experiment was then designed with the following combinations:

- scale factor (Factor A) at three levels—1.1, 1.2, and 1.4,
- grouping factor (Factor B) at two levels—2 and 3, and
- preprocessing flag (Factor C) at two levels—TRUE and FALSE.

Thus, it was a three-factor mixed-level experiment. For a given parameter setting, the algorithm, being deterministic, will produce the same output when run multiple times. Thus, there was no random error involved in the process. This resulted in a one-replicate design experiment.

To accomplish the tasks described earlier, the following strategy was adopted:

1. Perform an ANalysis Of VAriance (ANOVA) on the performance values for various settings to identify the significance of individual parameters and their interactions.
2. Based on the results of the ANOVA, build a regression model with the significant factors.
3. Follow it up with an optimization step to compute the optimal parameter settings.

In order to test the generality of the solution, the above process was repeated on six training videos. For each parameter setting, the algorithm output was obtained, and

TABLE 4
Sample Data Showing Performance Values for Different Parameter Settings

		Preprocessing flag at 0	
		Grouping	
		2	3
Scale	1.1	0.669158	0.785116
	1.2	0.766816	0.842430
	1.4	0.807907	0.796853

		Preprocessing flag at 1	
		Grouping	
		2	3
Scale	1.1	0.669633	0.785496
	1.2	0.767765	0.842430
	1.4	0.807907	0.795713

the score was computed by using the $SFDA$ metric. Table 4 shows a sample data for one of the videos.

Being a one-replicate experiment, there was a need to build a reduced model to estimate the error. By observing the experimental data, we inferred that the level of factor C (preprocessing flag) was not significant to the performance of the algorithm. From Table 4, one can observe that the level of factor C causes a difference in the score, which is significant only in the fourth decimal. Hence, we used the sum of squares of factor C as our initial estimate for the error.

Through the ANOVA, it was found that the factors A and B (scale and grouping factors) and the interaction between them were the only significant sources of variation. This observation was consistent across all six training videos. Based on these results, the final model was built by combining sum of squares of C, AC, BC, and ABC to get a better estimate for the sum of squares of the error.

A regression model was then built with the significant factors as the model parameters. Since a linear model was not able to accurately capture the variation in the underlying distribution space, a quadratic model $(z = \beta_0 + \beta_1 X_A + \beta_2 X_B + \beta_3 X_A X_B + \beta_4 X_A^2)$ was built by introducing a quadratic term for the scale factor. The average R^2 value for this model was 0.88, which justified that this model was adequate. The optimal values for each of the six videos were obtained using this model.

A final regression model was built based on these optimal values. The motivation behind this step is to find the values that can be generalized across more videos. The values for scale and grouping factors after this optimization step were 1.20 (after rounding to the first decimal) and 3, respectively. With the preprocessing flag set to be TRUE for faster execution, the above values were used in the test set.

Appendix B

Text Postprocessing

Although the DCT method could detect text blocks in video frames, many text-like background blocks with high contrast and sharp edges were also marked as text block mistakenly due to their large text energies. It was critical to remove these false positives. Some feature-based postprocessing approaches have been proposed to solve this problem by analyzing the differences in texture features, color information, or motion energy between text blocks and background blocks. However, all of these features were computed within the blocks, a result of which was quite limited information extraction, especially when candidate blocks were small or video resolution was low.

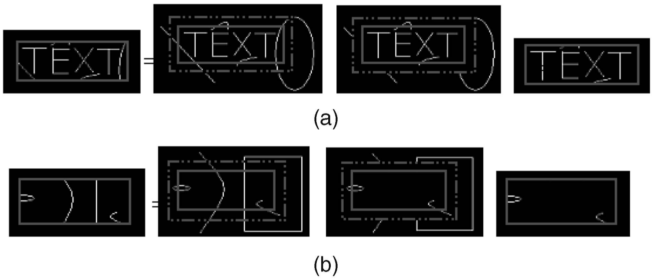


Fig. 9. Steps involved in removing edges (solid gray is original block, dashed gray is extended block, and white are edges).

In this work, based on the investigation of the relation between the candidate blocks and their neighbor areas, we propose a new postprocessing method that can remove false positives effectively by erasing background edges in both text blocks and background blocks. The basic idea of this approach was based on the assumption that a good text block should contain the entire text it detected, that is, all text edges should be bounded by the text block. On the other hand, the edges that come from neighbor areas are considered as background edges and should be erased from text blocks.

Fig. 9 shows a visualization of specific steps in the algorithm. First, the original text block was expanded to a larger block. This was done by examining all of the edge information in the original block area. Then, the Canny edge detector was applied to the expanded block. The resulting edges were then selectively erased if the edge intersected the expanded block. These *background* edges were removed by utilizing an edge tracking technique.

After removing the background edges, we then determined if a candidate block was a false positive by using the equation below:

$$\frac{|\text{Remaining edge pixels}|}{|\text{Pixels in the block}|} < T_1, \qquad (16)$$

where T_1 is a predefined threshold. Equation (16) indicates that if there are too few remaining edges in a block, then this is a background block. Examples of this technique in action can be seen in Figs. 10 and 11.

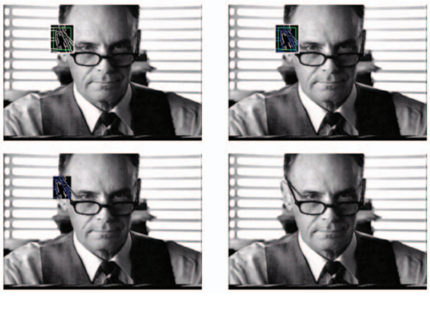


Fig. 10. Example of spurious edges removed near the face corner.

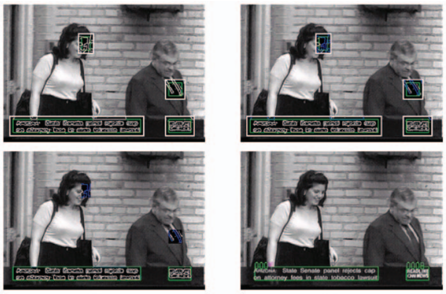


Fig. 11. Example of spurious edges removed on the face and tie area.

Acknowledgments

The work presented here was carried out over a period of seven years, from 2001 to 2007, at multiple institutions. The authors would like to thank Vladimir Mariano, Junghye Min, and Jinhyeong Park at the Pennsylvania State University for developing the initial framework for evaluation and David Mihalcik and David Doermann at the University of Maryland for the video data annotation and the support of the Video Performance Evaluation Resource (ViPER) tool. The authors are indebted to the guidance and support of Stephen Dennis, Thomas Drayer, John Prange, Dennis Moellman, and Paul Matthews from the US Department of Defense, who defined the goals and made performance evaluation a part of the VACE research program. They are grateful to Rajeev Sharma, Harish Raju, and Shubha Prasad (Video Mining Inc.), who headed the annotation team that generated the ground truth for the VACE Phase II evaluations. Their sincere thanks to Rainer Stiefelhagen and Keni Bernardin (University of Karlsruhe) for their efforts and cooperation in the development of the CLEAR metrics. They would like to thank R. Travis Rose (NIST) for his help in organizing and editing this manuscript. The authors are indebted to the following key VACE contractors for their insightful discussions during the design of this framework: Michael C. Nechyba (Pittsburgh Pattern Recognition Inc.), Ram Nevatia and Isaac Cohen (University of Southern California), Prem Natarajan (BBN), and Greg Myers (SRI). They are thankful to David Crandall for his help in the development of the text detection and tracking baseline algorithm.

DISCLAIMER: Certain commercial equipment, instruments, software, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by NIST nor is it intended to imply that the equipment, instruments, software, or materials are necessarily the best available ones for the purpose.

References

[1] V. Manohar, P. Soundararajan, M. Boonstra, H. Raju, D. Goldgof, R. Kasturi, and J. Garofolo, "Performance Evaluation of Text Detection and Tracking in Video," *Proc. Seventh Int'l Workshop Document Analysis Systems,* pp. 576-587, 2006.

[2] V. Manohar, P. Soundararajan, H. Raju, D. Goldgof, R. Kasturi, and J. Garofolo, "Performance Evaluation of Object Detection and Tracking in Video," *Proc. Seventh Asian Conf. Computer Vision,* pp. 151-161, 2006.

[3] M. Sezgin and B. Sankur, "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation," *J. Electronic Imaging,* vol. 13, no. 1, pp. 146-168, 2004.

[4] M.D. Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer, "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 12, pp. 1338-1359, Dec. 1997.

[5] A. Hoover, G. Jean-Baptiste, X. Jiang, P.J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D.W. Eggert, A. Fitzgibbon, and R.B. Fisher, "An Experimental Comparison of Range Image Segmentation Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 7, pp. 673-689, July 1996.

[6] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 10, pp. 1615-1630, Oct. 2005.

[7] S. Sarkar, P. Phillips, Z. Liu, I. Vega, P. Grother, and K. Bowyer, "The HumanID Gait Challenge Problem: Data Sets, Performance, and Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 2, pp. 162-177, Feb. 2005.

[8] C. Wilson, R.A. Hicklin, M. Bone, H. Korves, P. Grother, B. Ulery, R. Micheals, M. Zoepfl, S. Otto, and C. Watson, "Fingerprint Vendor Technology Evaluation 2003: Summary of Results and Analysis Report," Technical Report NISTIR 7123, Nat'l Inst. Standards and Technology, 2004.

[9] R. Cappelli, D. Maio, D. Maltoni, J.L. Wayman, and A.K. Jain, "Performance Evaluation of Fingerprint Verification Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 1, pp. 3-18, Jan. 2006.

[10] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss, "The FERET Evaluation Methodology for Face Recognition Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 10, pp. 1090-1104, Oct. 2000.

[11] D.M. Blackburn, M. Bone, and P.J. Phillips, "Facial Recognition Vendor Test 2000: Evaluation Report," technical report, Nat'l Inst. Standards and Technology, http://www.frvt.org/DLs/FRVT_2000.pdf, 2001.

[12] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the Face Recognition Grand Challenge," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 947-954, 2005.

[13] D. Young and J. Ferryman, "PETS Metrics: On-Line Performance Evaluation Service," *Proc. Joint IEEE Int'l Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance,* pp. 317-324, Oct. 2005.

[14] *PETS Metrics,* http://petsmetrics.net, 2008.

[15] *Computers in the Human Interaction Loop (CHIL),* http://chil.server.de, 2008.

[16] *Augmented Multiparty Interaction (AMI),* http://www.amiproject.org, 2008.

[17] *Evaluation du Traitement et de l'Interprétation de Séquences Vidéo (ETISEO),* http://www.silogic.fr/etiseo, 2008.

[18] *Cognitive Agent that Learns and Organizes (CALO),* http://caloproject.sri.com, 2008.

[19] *NIST Rich Transcription Meeting Recognition Evaluation (RT),* http://www.nist.gov/speech/tests/rt/, 2008.

[20] *Proc. Text REtrieval Conf. VIDeo Retrieval Evaluation (TRECVID),* http://www-nlpir.nist.gov/projects/trecvid/, 2008.

[21] J. Garofolo, R.T. Rose, and R. Steifelhagen, "Eval-Ware: Multimodal Interaction," *IEEE Signal Processing Magazine,* vol. 24, no. 2, pp. 154-155, 2007.

[22] *CLassification of Events, Activities and Relationships (CLEAR),* http://www.clear-evaluation.org, 2008.

[23] V. Manohar, M. Boonstra, V. Korzhova, P. Soundararajan, D. Goldgof, R. Kasturi, S. Prasad, H. Raju, R. Bowers, and J. Garofolo, "PETS versus VACE Evaluation Programs: A Comparative Study," *Proc. Ninth IEEE Int'l Workshop Performance Evaluation of Tracking and Surveillance,* pp. 1-6, 2006.

[24] S. Antani, D. Crandall, A. Narasimhamurthy, V.Y. Mariano, and R. Kasturi, "Evaluation of Methods for Detection and Localization of Text in Video," *Proc. Int'l Workshop Document Analysis Systems,* pp. 507-514, 2000.

[25] X. Hua, L. Wenyin, and H. Zhang, "An Automatic Performance Evaluation Protocol for Video Text Detection Algorithms," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 14, no. 4, pp. 498-507, 2004.

[26] J. Nascimento and J. Marques, "Performance Evaluation of Object Detection for Video Surveillance," *IEEE Trans. Multimedia,* vol. 8, no. 4, pp. 761-774, 2006.

[27] R.B. Fisher, "The PETS04 Surveillance Ground-Truth Data Sets," *Proc. IEEE Performance Evaluation of Tracking and Surveillance Workshop,* May 2004.

[28] R. Collins, X. Zhou, and S.K. Teh, "An Open Source Tracking Testbed and Evaluation Web Site," *Proc. IEEE Int'l Workshop Performance Evaluation of Tracking and Surveillance),* pp. 17-24, Jan. 2005.

[29] J. Black, T.J. Ellis, and P. Rosin, "A Novel Method for Video Tracking Performance Evaluation," *Proc. IEEE Performance Evaluation of Tracking and Surveillance Workshop,* Oct. 2003.

[30] L.M. Brown, A.W. Senior, Y. Tian, J. Connell, A. Hampapur, C. Shu, H. Merkl, and M. Lu, "Performance Evaluation of Surveillance Systems under Varying Conditions," *Proc. IEEE Performance Evaluation of Tracking and Surveillance Workshop,* Jan. 2005.

[31] K. Smith, D. Gatica-Perez, J. Odobez, and S. Ba, "Evaluating Multi-Object Tracking," *Proc. IEEE Empirical Evaluation Methods in Computer Vision Workshop,* June 2005.

[32] M. Liberman and C. Cieri, "The Creation, Distribution and Use of Linguistic Data: The Case of the Linguistic Data Consortium," *Proc. First Int'l Conf. Language Resources and Evaluation,* 1998.

[33] D. Doermann and D. Mihalcik, "Tools and Techniques for Video Performance Evaluation," *Proc. Int'l Conf. Pattern Recognition,* vol. 4, pp. 167-170, 2000.

[34] J.G. Fiscus, J. Ajot, and J. Garofolo, "The Rich Transcription 2007 Meeting Recognition Evaluation," *Proc. Multimodal Technologies for Perception of Humans, Joint Proc. Second Int'l Evaluation Workshop Classification of Events, Activities, and Relationships and the Spring 2007 Rich Transcription Meeting Evaluation,* R. Stiefelhagen, R. Bowers, and J. Fiscus, eds., 2007.

[35] J.R. Munkres, "Algorithms for the Assignment and Transportation Problems," *J. SIAM,* vol. 5, pp. 32-38, 1957.

[36] M.L. Fredman and R.E. Tarjan, "Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms," *J. ACM,* vol. 34, no. 3, pp. 596-615, July 1987.

[37] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity.* Prentice Hall, 1982.

[38] R.T. Rockafellar, *Network Flows and Monotropic Optimization.* John Wiley & Sons, 1984.

[39] D.E. Knuth, *The Stanford GraphBase: A Platform for Combinatorial Computing.* ACM, 1993.

[40] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, and P. Soundararajan, "The CLEAR 2006 Evaluation," *Multimodal Technologies for Perception of Humans,* pp. 1-44, Springer, 2006.

[41] E. Hjelmasa and B. Low, "Face Detection: A Survey," *Computer Vision and Image Understanding,* vol. 83, no. 3, pp. 236-274, 2001.

[42] M. Yang, D. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 1, pp. 34-58, Jan. 2002.

[43] P. Viola and M. Jones, "Robust Real-Time Object Detection," *Int'l J. Computer Vision,* 2002.

[44] R. Lienhart and J. Maydt, "An Extended Set of Haar-Like Features for Rapid Object Detection," *Proc. Int'l Conf. Image Processing,* pp. 900-903, 2002.

[45] *The Intel Open Source Computer Vision Library,* http://www.intel.com/technology/computing/opencv/, 2008.

[46] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 6, pp. 679-698, 1986.

[47] D.C. Montgomery, *Design and Analysis of Experiments,* sixth ed. John Wiley & Sons, 2005.

[48] K. Jung, K. Kim, and A. Jain, "Text Information Extraction in Images and Video: A Survey," *Pattern Recognition,* vol. 37, no. 5, pp. 977-997, 2004.

[49] D. Crandall, S. Antani, and R. Kasturi, "Extraction of Special Effects Caption Text Events from Digital Video," *Int'l J. Document Analysis and Recognition,* vol. 5, nos. 2-3, pp. 138-157, Apr. 2003.

[50] Z. Sun, G. Bebis, and R. Miller, "On-Road Vehicle Detection: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 5, pp. 694-711, May 2006.

[51] R. Cucchiara, A. Prati, M. Piccardi, and N. Scarabottolo, "Real-Time Detection of Moving Vehicles," *Proc. 10th Int'l Conf. Image Analysis and Processing,* pp. 618-623, 1999.

[52] Z. Zhu, G. Xu, B. Yang, D. Shi, and X. Lin, "VISATRAM: A Real-time Vision System for Automatic Traffic Monitoring," *Image and Vision Computing,* vol. 18, no. 10, pp. 781-794, 2000.

[53] Z. Kim and J. Malik, "Fast Vehicle Detection with Probabilistic Feature Grouping and Its Application to Vehicle Tracking," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* pp. 524-531, 2003.

[54] M. Taj, E. Maggio, and A. Cavallaro, "Multi-Feature Graph-Based Object Tracking," *Multimodal Technologies for Perception of Humans,* pp. 190-199, 2006.

[55] Y. Zhai, P. Berkowitz, A. Miller, K. Shafique, A. Vartak, B. White, and M. Shah, "Multiple Vehicle Tracking in Surveillance Videos," *Multimodal Technologies for Perception of Humans,* pp. 200-208, 2006.

[56] W. Abd-Almageed and L. Davis, "Robust Appearance Modeling for Pedestrian and Vehicle Tracking," *Multimodal Technologies for Perception of Humans,* pp. 209-215, 2006.

[57] X. Song and R. Nevatia, "Robust Vehicle Blob Tracking with Split/Merge Handling," *Multimodal Technologies for Perception of Humans,* pp. 216-222, 2006.

Rangachar Kasturi received the BE (electrical) degree from Bangalore University, India, in 1968 and the MSEE and PhD degrees from Texas Tech University in 1980 and 1982, respectively. He was a professor of computer science and engineering and electrical engineering at the Pennsylvania State University from 1982 to 2003 and was a Fulbright Scholar in 1999. His research interests are in document image analysis, video sequence analysis, and biometrics. He served as the editor-in-chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (from 1995 to 1998) and the *Machine Vision and Applications* (from 1993 to 1994) journals. He is an author of the textbook *Machine Vision* (McGraw-Hill, 1995). He is the 2008 president of the IEEE Computer Society. He was the president of the International Association for Pattern Recognition (IAPR) from 2002 to 2004. He is a fellow of the IEEE and the IAPR.

Dmitry Goldgof received the MS degree in computer engineering from Rensselaer Polytechnic Institute in 1985 and the PhD degree in electrical engineering from the University of Illinois at Urbana-Champaign in 1989. He is currently a professor and an associate chair of the Department of Computer Science and Engineering and a member of the H. Lee Moffitt Cancer Center and Research Institute, wherein, from 2002 to 2003, he held the position of professor in bioinformatics and cancer control. Previously, he held visiting positions in the Department of Computer Science at the University of California, Santa Barbara, and in the Department of Computer Science at the University of Bern, Switzerland. His research interests include motion and deformation analysis, image analysis and its biomedical applications, bioinformatics, and pattern recognition. He has graduated 12 PhD and 35 MS students, edited four books, published more than 60 journal and more than 130 conference proceedings papers, and was granted a US patent. He served as an IEEE Distinguished Visitor from 2004 to 2006 and on the Board of Governors of the IEEE Systems, Man, and Cybernetics Society in 2007. One of his papers was selected by the International Medical Informatics Association for the 2000 Yearbook, containing "the best of medical informatics." He is an associate editor for the *IEEE Transactions on Systems, Man and Cybernetics B* and for the *International Journal of Pattern Recognition and Artificial Intelligence.* He has been involved in numerous professional society activities, has served as a North American editor for the *Image and Vision Computing Journal*, as a member of the editorial board of *Pattern Recognition*, and as an associate editor for the *IEEE Transactions on Image Processing.* He received the Annual Pattern Recognition Society Award in 1993 and 2002. He was a local arrangement chair for ICPR '08. He is a fellow of the IEEE "for contributions to computer vision and biomedical applications."

Padmanabhan Soundararajan received the bachelor's degree in electronics and communication engineeering from Mysore University, India, in 1995 and the PhD degree in computer science and engineering from the University of South Florida, Tampa, in 2004. From 1995 to 1998, he was a project assistant at the Indian Institute of Science, Bangalore. He is currently working at Nielsen Media Research. His research interests include video tracking and recognition systems, perceptual organization, statistical techniques in pattern recognition, and performance evaluation of vision systems. He is a member of the IEEE and the IEEE Computer Society.

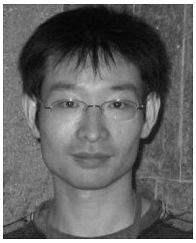
Vasant Manohar received the BE degree (honors) in computer science from the Birla Institute of Technology and Science, Pilani, India, and the MS degree in computer science from the University of South Florida (USF) in 2003 and 2006, respectively. He is currently a PhD candidate in the Computer Science and Engineering Department at USF. His general research interests include computer vision, image processing, and pattern recognition. His specific topics of research include video-based face recognition, nonrigid motion analysis, and empirical evaluation techniques for object detection and tracking in video. He is a member of Sigma Xi and Tau Beta Pi. He is a student member of the IEEE.

John Garofolo has been with the US National Institute of Standards and the Technology Information Technology Laboratory since 1987. In the late 1980s, his work focused on the evaluation of early speech-to-text continuous speech transcription systems (formerly referred to as automatic speech recognition) in the context of a number of DARPA programs. His work on corpus and evaluation-driven research and development helped to significantly speed progress in this technology. In the 1990s, he extended his work to the evaluation of speech understanding systems in the DARPA ATIS and Communicator programs and, in the late 1990s, to information retrieval applied to speech in the TREC Spoken Document Retrieval Track. He saw multimodality as a key challenge for the future and spearheaded the construction of a massively instrumented multimodal/multichannel meeting room. The data collected in the room has been used in a variety of speech and computer vision evaluations and the project inspired two European programs focused on multimodality. In 2004, he began working with the Video Analysis and Content Extraction Program under what is now IARPA to port approaches used in speech technology evaluation to the computer vision domain. He developed an extensive corpus and metrics-based evaluation framework for video object detection and tracking technologies, as well as video text recognition technologies in collaboration with the University of South Florida. He is currently the manager of the NIST Speech Technologies Group and oversees a number of human language and computer vision technology evaluation activities. He continues to reach to bring his knowledge of evaluation-driven research and development to new technology communities.

Rachel Bowers, biography and photo are not available.

Matthew Boonstra received the BS degree in computer engineering and the MS degree in computer science from the University of South Florida in 2003 and 2007, respectively. He is a PhD candidate at the University of South Florida. His research interests include artificial intelligence, machine learning, and computer vision. He is a student member of the IEEE and the IEEE Computer Society and a Tau Beta Pi alumnus.

Valentina Korzhova received the MS degree in computer science from the University of South Florida (USF) in 2006, where she is currently a PhD candidate. She works as a research assistant at USF, specializing in image processing and pattern recognition. Her additional interests include algorithm optimization and mathematical modeling with applications in medicine. She has published more than 28 scientific works in several conference proceedings and journals. She is a student member of the IEEE.



Jing Zhang received the MS degree in electronics and communication from Xi'an Jiaotong University, China, in 2004. From 2005 to 2006, he was a PhD student at the University of Hong Kong. He is currently working toward the PhD degree in the Computer Science and Engineering Department at the University of South Florida. His research interests include text detection in videos and graph recognition. He is a student member of the IEEE.