18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Video-to-Text Information Fusion Evaluation for Level 5 User Refinement

Erik Blasch
Air Force Research Laboratory
Rome, NY, 13441
erik.blasch.1@us.af.mil

Dan Shen, Genshe Chen
Intelligent Fusion Technology
Germantown, MD 20876
{dshen, gchen}@intfusiontech.com

Arslan Basharat, Roddy Collins
Kitware
Clifton Park, NY, 12065
arslan.basharat@kitware.com

Haibin Ling
Temple University
Philadelphia, PA 19122
hbling@temple.edu

Riad Hammoud
BAE Systems
Burlington, MA 01803
riad.hammoud@baesystems.com

Alex Aved, James Nagy
Air Force Research Laboratory
Rome, NY, 13441
{alexander.aved, james.nagy.2}@us.af.mil

Abstract— *Video-to-Text (V2T) fusion is an example of coordinating low-level information fusion (LLIF) with high-level information fusion (HLIF) through semantic descriptions of physical information. Using hard (e.g., video) and soft (i.e., text) data fusion affords Level 5 User Refinement of object characterization, target tracking, and situation assessment. Building on our previous video-to-text (V2T) Fusion2014 paper, we extend the method for evaluation of eight tracking methods compared for extraction of semantic information including target number, category, attribute, and direction. Using the CMUSphinx speech-to-text system for semantic parsing of user call-outs, preliminary results show the integration of video tracking and text analysis is better with the compressive tracker (CT) and the Tracking-Learning-Detection (TLD) method. The feature analysis of the CT and TLD demonstrate the ability to associate user call-out text-based semantic descriptors with video exploitation. The results are presented in a visualization tool for rapid production to aid user refinement (HLIF) and object assessment (LLIF) functions.*

Keywords: *Information Fusion, Level 5 User Refinement, High-Level Information Fusion, Semantic Label, L1 tracker, Hard-soft fusion*

I. INTRODUCTION

Numerous efforts have been established to link hard (e.g., video) and soft (i.e., text) data fusion in support of high-level information fusion (HLIF) [1]. Figure 1 demonstrates the standard levels of information processing including Level 0 (e.g., registration) and Level 6 (e.g., mission/context management). Low-level information fusion (LLIF) includes Level 1 (L1) object tracking and identification [2, 3, 4, 5, 6] for which we explore various tracking methods [7, 8, 9]. HLIF includes situation (L2) and threat assessment (L3) as well as sensor (L4), user (L5), and mission management (L6) [10, 11, 12]. Inherent in HLIF is context assessment/management [13] over the environment [14], sensors [15], and targets [16]. A fusion system gives the user a perspective that results in situation awareness [17] and situation assessment [18] for resource management [19] and knowledge management [20] for product reporting.

Linking *physics and human information fusion (PHIF)* has been explored in various applications such as multi-intelligence [21, 22], scene content [23, 24], narratives [25], and data and decisions [26]. Recently we have applied PHIF to various imagery tracking methods [27, 28, 29, 30]. Emerging issues include deep learning, scalable methods for dynamic analysis, and big data analytics. To bind HLIF with LLIF requires context [31] and information management [32].

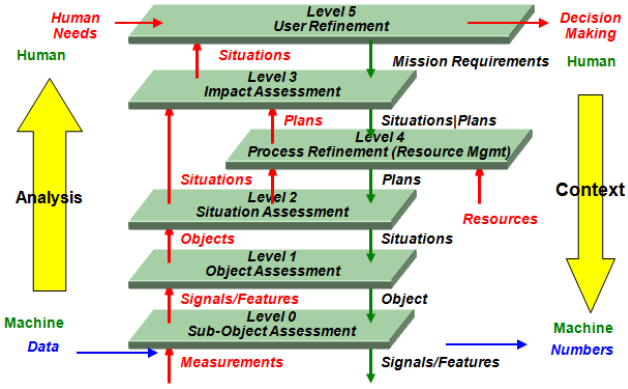


Figure 1 – Information Fusion Levels

As shown in Figure 2, a user typical refines the outputs of object assessment by providing semantic descriptions of information from video processing. The semantic analysis is in the form of textual descriptions that requires hard-soft fusion. Both the semantic representation from a user and a physical representation from the machine of a scene include contextual information to refine estimates of the situation.

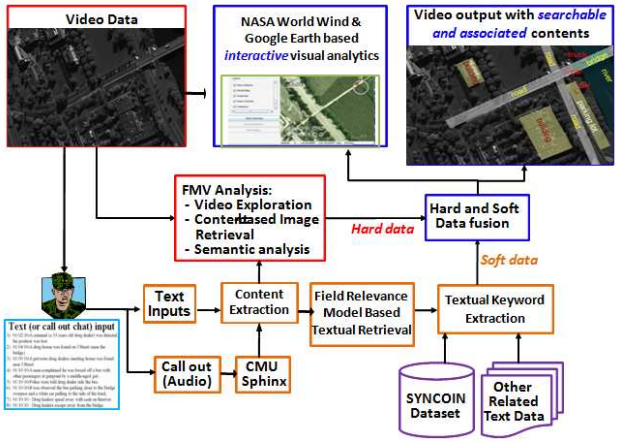


Figure 2 – HLIF-LLIF (Hard-soft) Fusion System.

A concept that bridges video and text analysis is content-based image retrieval (CBIR) or query by image content, as shown at the right in Figure 3. CBIR assesses the content (e.g., color, features) of the image versus the metadata accompanying the image which is shown as a *meta image*. These features are assessed for image semantic descriptions [33] and a survey is found in [34]. A slight misnomer in CBIR (as with information fusion) is that a *querying technique* implies that a user supplies a known image of semantic importance (e.g., car exemplar) for comparison. Once the user has chosen the image exemplar, automatic reasoning is done to compare images with similar features for a match; however user refinement is needed such as scaling [35]. Two related methods include *semantic retrieval* (e.g., text based requests matched to a known image) and *relevance feedback* where a user refines the image search by validating correct results as a tagged image.

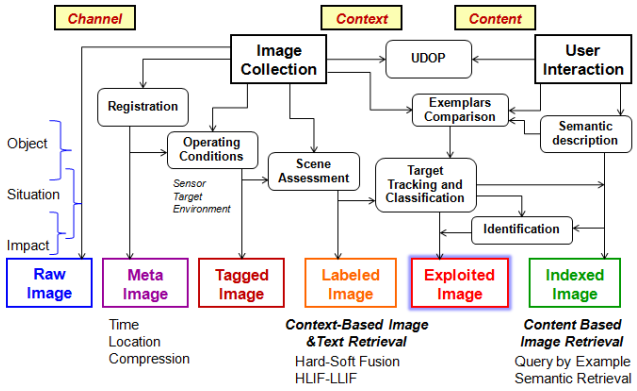


Figure 3 – Image Annotation types.

For our purposes, we are interested in a combination of a Meta, Labeled, and an Indexed Image (shown at the bottom of Figure 3). We highlight the concept as a Labeled Image from HLIF and an Exploited Image from LLIF. Note that from regular full motion video channels, images could include no (raw), collected (meta), or manual (tagged) information. Using information fusion, the *Labeled Image* includes metadata and feature extraction from a machine; while an *Exploited image* incorporates the user's mission needs. To do this, we use user call-outs about the image (i.e., soft, text, HLIF analysis) matched to that of feature extraction (e.g., hard, classifications, LLIF assessments). The distinction made here is that our concept differs from traditional CBIR as the mission, scenario, or priority comes from context-based image and text (CBITR) retrieval.

The combination of video and text fusion is evident in many applications such as multimedia production (e.g., video archiving and understanding [36, 37]), target classification for law enforcement (e.g., audio-imagery fusion [38]), and medical diagnosis (e.g., transcriptions with annotated imagery). We seek semantic representations as video-based hidden Markov Models (HMM) [39] or patterns [40] could be used in multi-level fusion architecture.

To detail the process, Sect. II discusses the user call-outs as a function of HLIF using speech-to-text methods. Sect. III presents the eight tracking methods tested. Sect. IV is the experiment results and Sect. V presents the user interface for "User Refinement". Conclusions are presented in Sect. VI.

II. Object Text Recognition

A. Speech-to-Text with CMUSphinx

The CMU Sphinx is a complete state-of-the-art hidden Markov model (HMM) based open source speech recognition system [41]. Designed at Carnegie Mellon University, CMUSphinx is one of the most versatile recognition systems in use today. Previous developments were used for news transcription [42]. As an HMM-based system, like most other speech recognition systems, it functions by first learning the characteristics (or parameters) of a set of sound units, and then using what it has learned about the units, finds the most probable sequence of sound units for a given speech signal. The process of learning about the sound units is called *training*. The process of using the knowledge acquired to deduce the most probable sequence of units in a given signal is called *decoding*, or simply recognition.

Accordingly, we will need those components of the CMUSphinx system that we can use for training and for recognition including the *CMUSphinx trainer* and a *CMUSphinx decoder*. For our speech-to-text call-outs, a call-out message is exploited such as: "*one white pick-up truck, turns left, travels north, center of screen*". CMUSphinx recognizes 'white' as 'like', which results in semantic confusion. Therefore, we need to perform training to calibrate the speech recognition toolkit.

B. Training using Audio Classifications

The CMUSphinx system comes with various acoustic models for several languages that were optimized for robust speech recognition. For example, acoustic models for microphone, broadcast, and speech over a telephone are provided for US English. Applications for direct use include command-and-control, large vocabulary, and text-based discourse [43].

When higher accuracy is needed, CMUSphinx provides ways for *adaptation*. Adaptation supports different languages (e.g., UK English), recording environments (e.g., close-distance microphone or a handset channel), or when a slightly different tone of voice is present (e.g., speaker under stress). Using the adaptation feature allows for rapid support of a new dictionary model (e.g., target analysis) [43]. For our analysis, we need to train our own model to ensure LLIF-HLIF semantic associations for target tracking are viable.

The CMUSphinx training function uses a set of sample speech signals to learn the model sound units as a *training database*. The database affords acoustic model statistical analysis. However, the sound units (or sequences) have to be designated from the training database as a file called the *transcript file*. The transcript file includes the word sequence, non-speech sounds, and tags to associate the desired sequence with the corresponding speech signal as a dictionary [43]. Two dictionaries are used: (1) *language dictionary* in which legitimate words are mapped sequences of sound units (or sub-word units) and (2) *filler dictionary* from which non-speech sounds are mapped to sound units.

After training, the *decoder* checks the training results by testing the acoustic model against the database and reference transcriptions. During the testing stage, both the order of words in the language and model quality estimates are produced [43].

It is noted that the *Database* should have a variety of speakers, recording conditions, and linguistic sentences. Using the test images, a set of call-outs were conducted to compile the CMUSphinx trained library. The testing database size was small relative to the key words used in normal target tracking over the video sequences, so additional sentences were added for training. The additional information included: other targeting information not in the images, discussions an operator might have with other members, and non-mission related activities [9]. The CMUSphinx allowed for a convenient method for compiling the database for speech-to-text operator call-outs that relate to the corresponding features available from video tracking [30].

C. Text Analytics

For the video examples, we are interested in semantic descriptions of labeling target movement and feature classification (which could include identification information of allegiance) to support video and text fusion [9]:

- Direction {north, east, south, west)
- Attribute {color, size, histogram}
- Category {person, object, vehicle}
- Label {number, priority}

For the microtext analytics of the call-outs, we use natural language processing (NLP) for entity extraction and entity resolution as detailed in [9, 44, 45, 46]. The entity analysis prioritized the salient words used in the call-outs corresponding to video (as well as time correlation). Additionally, we can explore text-based activity [47] and sentiment analysis [48]. Next, we present tracking methods used in the video-to-text fusion analysis.

III. VIDEO TRACKING METHODS

Building on our previous results in video tracking [49, 50, 51, 52], we sought to evaluate various tracking methods in support of V2T analysis. Previously we highlighted the L1 particle filtering method. In this study, we compared PF without feature analysis. Both the compressive tracker (CT) and the tracking-learning-detection (TLD) method afford feature analysis to align with the speech to text call-outs. A review of the methods are briefing discussed below.

A. T1: Compressive Tracker (CT)

Compressive tracking [53] is a low computational complexity model based on features extracted in the compressed domain. By applying these feature extracted in preprocessing, the surrounding background is separated from the target object via a naive Bayes classifier. In the appearance model, features are selected by an information-preserving and non-adaptive dimensionality reduction from the multi-scale image feature space based on compressive sensing theories. The framework of compressive tracker is presented in table below.

Algorithm 1. Compressive Tracking

Input: video frames

1.Sample a set of image patches, $D^{\gamma} = \{\mathbf{z} | \|\mathbf{l}(\mathbf{z}) - \mathbf{l}_{t-1}\| < \gamma\}$ where $\mathbf{l}_{t-1}$ is the tracking location at the (t-1)-th frame, and extract the features with low dimensionality.

2.Use classifier B to each feature vector $\mathbf{v}(\mathbf{z})$ and find the tracking location $\mathbf{l}_t$ with the maximal classifier response.

3.Sample two sets of image patches $D^{\alpha} = \{\mathbf{z} | \|\mathbf{l}(\mathbf{z}) - \mathbf{l}_t\| < \alpha\}$ and $D^{\zeta,\beta} = \{\mathbf{z} | \zeta < \|\mathbf{l}(\mathbf{z}) - \mathbf{l}_t\| < \beta\}$ with $\alpha < \zeta < \beta$. (γ, α, ζ and β are search radius of the set to detect the object location).

4.Extract the features with these two sets of samples and update the classifier parameters.

Output: Tracking location $\mathbf{l}_t$ and classifier parameters.

A random matrix R projects data from high dimensional image space $\mathbf{x} \in \mathbb{R}^m$ to a low dimensional space $\mathbf{v} \in \mathbb{R}^n$: $\mathbf{v} = R\mathbf{x}$, where $n<<m$. For each sample $\mathbf{z} \in \mathbb{R}^m$, its low-dimensional representation is $\mathbf{v} = (v_1, \cdots, v_n)^{\mathrm{T}} \in \mathbb{R}^n$. All elements in $\mathbf{v}$ are independently distributed and a naive Bayes classifier is modeled:

$$B(\mathbf{v}) = \log\left(\frac{\prod_{i=1}^{n} p(v_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(v_i|y=0)p(y=0)}\right) = \sum_{i=1}^{n} \log\left(\frac{p(v_i|y=1)}{p(v_i|y=0)}\right) \quad (5)$$

where the uniform prior us assumed $p(y=1) = p(y=0)$, and $y \in \{0,1\}$ is a binary variable which represents the labels of the samples. The conditional distribution $p(v_i|y=1)$ and $p(v_i|y=0)$ in the classifier $B(\mathbf{v})$ are assumed to be Gaussian distributed with four parameters $\mu_i^1, \sigma_i^1, \mu_i^0, \sigma_i^0$ where

$$p(v_i|y=1) \sim N(\mu_i^1, \sigma_i^1),\ p(v_i|y=0) \sim N(\mu_i^0, \sigma_i^0) \quad (6)$$

The scalar parameter above are incrementally updated

$$\mu_i^1 \leftarrow \lambda\mu_i^1 + (1-\lambda)\mu^1$$

$$\sigma_i^1 \leftarrow \sqrt{\lambda(\sigma_i^1)^2 + (1-\lambda)(\sigma^1)^2 + \lambda(1-\lambda)(\mu_i^1 - \mu^1)^2}, \quad (7)$$

where $\lambda > 0$ is a learning parameter,

$$\sigma^1 = \sqrt{\frac{1}{n}\sum_{k=0|y=1}^{n-1}(v_i(k) - u^1)^2}\ , \text{and}$$

$$\mu^1 = \frac{1}{n}\sum_{k=0|y=1}^{n-1} v_i(k).$$

The above equations can be easily derived by maximal likelihood estimation.

B. T2: Tracking-Learning-Detection Tracker (TLD)

The TLD tracker [54] is a framework designed for long-term tracking of an unknown object in a video stream. Its block diagram is shown in Figure 4. The components of the framework are characterized as follows: *Tracker* estimates the object's motion between consecutive frames under the assumption that the frame-to-frame motion is limited and the object is visible. The tracker is likely to fail and never recover is the object moves out of the camera view. *Detectors* treats every frame as independent and performs full scanning of the image to localize all appearances that have been observed and learned in the past. As any other detector, the detector makes two types of errors: false positive and false negative. *Learning* observes performance of both, tracker and detector, estimates detector's errors and generates training examples to avoid

these errors in the future. The learning component assumes that both the tracker and the detector can fail. By virtue of the learning, the detector generalizes to more object appearances and discriminates against background.

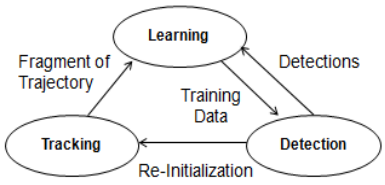


Figure 4 – Block Diagram of TLD System.

C. T3: Robust Fragments Tracker (Frag)

The Frag tracker [55] applies a recognition-by-parts approach to object tracking. The template object is represented by multiple image fragments or patches. The patches are arbitrary and are not based on an object model. Every patch votes on the possible positions and scales of the object in the current frame, by comparing its histogram with the corresponding image patch histogram. A robust statistic is then minimized in order to combine the vote maps of the multiple patches.

The Frag tracker overcomes several difficulties which cannot be handled by traditional histogram-based algorithms (e.g. mean shift). First, by robustly combining multiple patch votes, partial occlusions or pose change are able to be handled. Second, the geometric relations between the template patches take into account the spatial distribution of the pixel intensities - information which is lost in traditional histogram-based algorithm. Third, tracking large targets has the same computational cost as tracking small targets.

D. T4: Structured Output Tracking with Kernels (STRUCK)

In the STRUCK tracker [56], a framework for adaptive visual object tracking based on structured output prediction. By explicitly allowing the output space to express the needs of the tracker, the STRUCK tracker can avoid the need for an intermediate classification step. This method uses a kernelized structured output support vector machine (SVM), which is learned online to provide adaptive appearance tracking [57]. A budgeting mechanism is introduced for real-time application and preventing the unbounded growth in the number of support vectors which would otherwise occur during tracking. This algorithm is able to outperform state-of-the-art trackers on various benchmark videos. Moreover, additional features and kernels are easily incorporated into this framework and results in increased performance. However, the inevitable concern of this method is the high computational complexity.

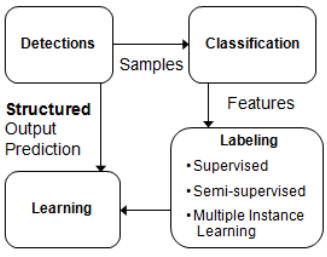


Figure 5 – The paradigms of the traditional and Struck trackers.

Figure 5 shows an adaptive tracking-by-detection paradigm [54]: given the current estimated object location, traditional approaches (shown on the right-hand side) generate a set of samples and, depending on the type of learner, produce training labels. Struck tracker (left-hand side) avoids these steps, and operates directly on the tracking output.

E. T5: Particle Filter Tracker (PF)

Particle Filter tracker methods [49-52] are a set of on-line posterior density estimation algorithms that estimate the posterior density of the state-space by directly implementing the Bayesian recursion equations. PF methods use a sampling approach, with a set of particles to represent the posterior density. The state-space model can be non-linear and the initial state and noise distribution can take any form required. PF methods provide a well-established methodology for generating samples from the required distribution without requiring assumptions about the state-space model or the state distribution. However, PF methods do not perform well when applied to high dimensional systems. They implement the Bayesian recursion equations directly by using an ensemble based approach. The samples from the distribution are represented by a set of particles; each particle has a weight assigned to it that represent the probability of that particle being sample from the probability density function.

Weight disparity leading to weight collapse is a common issue encounter in these filtering algorithms. However it can be mitigated by including a resampling step before the weights become too uneven. In the resampling step, the particles with negligible weights are replaced by new particles in the proximity of the particles with higher weights.

The objective of a particle filter is to estimate the posterior density of the state variables given the observation variables. The particle filter can be designed for a Hidden Markov Model (HMM), where the system consists of hidden and observable variables for sparse learning [58, 59]. The observable variables (observation process) are related to the hidden variables (state-process) by some functional form that is known. Similarly the dynamical system describing the evolution of the state variables is also known probabilistically. A generic particle filter estimates the posterior distribution of the hidden states using the observation measurement process. Consider a state-space shown in the diagram (Figure 6). The objective of the particle filter is to estimate the values of the hidden states $\boldsymbol{x}$, given the value of the observation process $\boldsymbol{y}$.

The particle filter aims to estimate the sequence of hidden sequences, x_k, based only on the observed data y_k for $k = 0,1,2,3,\cdots,K$. All Bayesian estimates of x_k follow from the posterior distribution $p(x_k|y_0, y_1, \cdots, y_k)$. In contrast, the importance sampling approach would model the full posterior $p(x_0, x_1, \cdots x_k|y_0, y_1, \cdots, y_k)$.

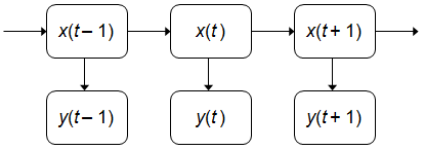


Figure 6 – The State-Space (The hidden states given by the vector $\boldsymbol{x}$, and the observation states given by the vector $\boldsymbol{y}$)

F. T6: On-line Boosting Tracker (BOOST)

The BOOST tracker [60] is a novel on-line AdaBoost feature selection algorithm for tracking. The distinct advantage of the method is its capability of on-line training. This allows adapt the classifier while tracking the object. Therefore appearance changes of the object (e.g. out of plane rotation, illumination changes) are handled quite naturally. Moreover, depending on the background the algorithm selects the most discriminating features for tracking resulting in stable tracking results. By using fast computable features (e.g., Haar-like wavelets, orientation histograms, local binary patterns) the algorithm runs in real-time.

The main idea of on-line boosting is the introduction of the *selectors*. They are randomly initialized and each of them holds a separate feature pool of weak classifiers [61]. A pool of classifiers has been applied to tracking [62, 63, 64]. When a new training sample arrives the weak classifiers of each selector are updated. The best weak classifier (having the lowest error) is selected by the selector where the error of the weak classifier is estimated from samples seen so far. The complexity is determined by the number of selectors.

The part which requires most of the processing time is the updating of weak classifiers. In order to speed up this process, we propose as a modification to use a single "global weak classifier" pool (see Figure 7) which is shared by all selectors instead of single pools for each of them. The advantage of this modification is that now for each sample that arrives, all weak classifiers need to be updated only once.

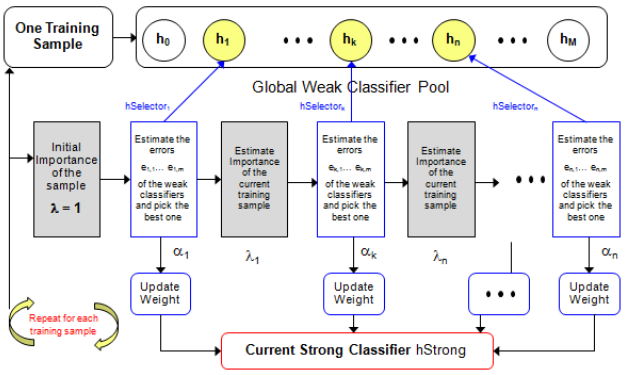


Figure 7 – Principle of on-line boosting for feature selection

Then the selectors sequentially switch to the best weak classifiers need to be updated only once. Then the selectors sequentially switch to the best weak classifier with respect to the current estimated λ and the importance weight is passed on to the next selector. This procedure is repeated until all selectors are updated. Finally, at each time step an updated strong classifier is available. In order to increase the diversity of the weak classifiers and to allow changes in the environment, the worst weak classifier of the shared feature pool is replaced with a new randomly chosen one.

G. T7: Semi-Supervised Boosting Tracker (Semi-BOOST)

Semi-BOOST tracker [65, 66] is a novel on-line semi-supervised boosting method which significantly alleviates the drifting problem in tracking applications. This allows user to limit the drifting problem while still staying adaptive to appearance changes. The main idea is to formulate the update process in a semi-supervised fashion as combined decision of a given prior and an on-line classifier. This comes without any parameter tuning.

Algorithm 2. On-line Semi-supervised Boosting for feature selection

Required: training (labeled or unlabeled) example $< x, y >, x \in \chi$
Required: prior classifier H^P(can be initialized by training on χ^L)
Required: strong classifier H (initialized randomly)
Required: weight $\lambda^c_{n,m}, \lambda^w_{n,m}$(initialized with 1)

for $n = 1,2,\cdots,N$ **do**//
  if $x \in \chi^L$ **then**
    $y_n = y, \lambda_n = \exp(-yH_{n-1}(x))$
  else
    $y_n = \text{sign}(p(x) - q(x)), \lambda_n = |p(x) - q(x)|$
  end if
  for $m = 1,2,\cdots,M$ **do** //update the selector h^{sel}_n
    $h_{n,m} = \text{update}(h_{n,m} < x, y >, \lambda)$
    //estimate errors
    if $h^{weak}_{n,m}(x) = y$ **then**
      $\lambda^c_{n,m} = \lambda^c_{n,m} + \lambda_n$
    else
      $\lambda^w_{n,m} = \lambda^w_{n,m} + \lambda_n$
    end if
    $e_{n,m} = \frac{\lambda^w_{n,m}}{\lambda^c_{n,m} + \lambda^w_{n,m}}$
  end for
  //choose weak classifier with the lowest error
  $m^+ = \text{argmin}_m(e_{n,m}), e_n = e_{n,m+}, h^{sel}_n = h_{n,m+}$
  if $e_n = 0$ or $e_n > \frac{1}{2}$ **then**
    exit
  end if
  $\alpha_n = \frac{1}{2} \cdot \ln(\frac{1-e_n}{e_n})$ //calculate voting weight
end for

H. T 8: Multiple Instance Learning Tracker (MIL)

The MIL framework [67] allows users to update the appearance model with a set of image patches, even though it is not known which image patch precisely captures the object of interest. This leads to more robust tracking results with fewer parameter tweaks. Weak classifiers are chosen sequentially to optimize the following criteria: $(\mathbf{h}_k, \alpha_k) = \text{argmax}_{\mathbf{h} \in \mathcal{H}, \alpha} J(\mathbf{H}_{k-1} + \alpha \mathbf{h})$ where $\mathbf{H}_{k-1}$ is the strong classifier made up of the first (k-1) weak classifiers, and $\mathcal{H}$ is the set of all possible weak classifiers. In batch boosting algorithms, the objective function J is computed over the entire training dataset.

Algorithm 3. On-line MILBoost

Input: Dataset $\{X_i, y_i\}_{i=1}^N$, where $X_i = \{x_{i1}, x_{i2}, \cdots\}, y_i \in \{0,1\}$
1. Update all M weak classifier in the pool with data $\{x_{ij}, y_i\}$.
2. Initialize $H_{ij} = 0$ for all i, j
3. **for** k = 1to K **do**
4. **for** m=1 to M **do**
5. $p^m_{ij} = \sigma(H_{ij} + h_m(x_{ij}))$
6. $p^m_i = 1 - \prod_j (1 - p^m_{ij})$
7. $\mathcal{L}^m = \sum_i (y_i \log(p^m_i) + (1 - y_i) \log(1 - p^m_i))$

8. **end for**
9. $m^* = \text{argmax}_m \mathcal{L}^m$
10. $\mathbf{h}_k(x) \leftarrow h_{m^*}(x)$
11. $H_{ij} = H_{ij} + \mathbf{h}_k(x)$
12. **end for**

Output: Classifier $\mathbf{H}(x) = \sum_k \mathbf{h}_k(x)$, where $p(y|x) = \sigma(\mathbf{H}(x))$

From these sampling of common visual trackers, we sought to compare them in a scenario in which a user is calling-out the activities for track association and refinement. Comparisons of some of these methods are popular in the literature such as the CT, MIL, Frag, to the Stuck with a learning SVM [68].

IV. Results

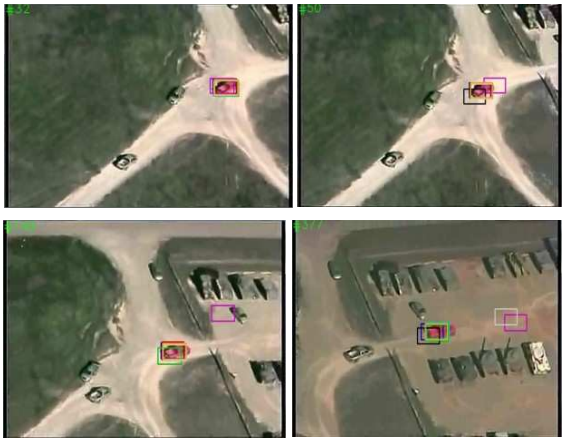
In the experiments, the platform of running all the trackers is Intel core i7-4500U 2.4GHz and 8 GB memory. To quantitatively compare robustness under challenging conditions, we manually annotated the target's bounding box in each frame for all the test sequences. The test sequences we selected are the classical sequences that afford speech-text analysis. Our example includes video tracking "person *jogging*" and "car *moving*". Next, we used the CMU Sphinx tool for speech recognition. The resulting speech-to-text chat was analyzed semantically to match the objects and activities. The video "car" is a very challenging airborne video sequence in VIRAT Video Dataset [69]. As can be seen in Figure 8, Tracker PF(pink) and S-BOOST(white) do not perform robustly under low-resolution and realistic conditions. As shown in Table 1 and 2, TLD and Compressive Trackers show their robustness in average tracking errors and tracking quality comparison; respectively as per the request of "car".

Table 1. Average tracking errors. The error is measured using the Euclidean distance of two center points, which has been normalized by the size of the target from the ground truth. The last row is the average error for each tracker over all the test sequences.

	PF	FRAG	STRUCK	BOOST	S-Boost	MIL	TLD	CT
jogging	0.1885	0.6383	0.8526	0.0570	0.8916	0.8211	**0.0056**	0.0085
pole	0.7520	0.0409	0.5728	0.0109	0.8591	0.0072	**0.0068**	0.0093
car	6.9842	0.3216	0.4685	0.4169	4.2561	0.5714	0.2034	**0.1026**
Average	2.6416	0.3336	0.6313	0.1616	2.0022	0.4665	**0.0719**	0.1204

The accuracy of the tracking methods is based on the scoring of the target center-point (pixel) to the track output distance. In general, the distance scoring for track accuracy could be important for pinpointing the target; however metadata information for geo-rectification affects the final result. However, using a semantic call-out helps determine if the target being tracked is correct which improves track quality. The various semantic descriptors can also be used in the track quality assessment (e.g., speed).

Track quality is not based on a center-point, but a box around the object of interest. While all the methods afford feature analysis and learning, the matching of the call-out to the target type (e.g., color) and movement (e.g. direction) is more robust in the CT and TLD. Current efforts are being explored to better understand the differences. One choice for the user is that the TLD method reduces the track error (at the center point) while the CT is better for semantic analysis with the call-out – shown in Table 2.



Legend: PF (pink), FRAG (green), STRUCK (cyan), BOOST (black), S-BOOST (white), MIL (orange), TLD (red). CT (blue)

Figure 8 – Tracking results of different algorithms in video "car".

Table 2. Average track quality. The quality is measured using the area coverage between the tracking box and the annotated call out.

	PF	FRAG	STRUCK	BOOST	S-Boost	MIL	TLD	CT
jogging	0.4141	0.1643	0.1339	0.1761	0.1294	0.5333	0.5369	**0.6836**
pole	0.3063	0.2791	0.3524	0.3939	0.0176	0.3422	**0.5449**	0.5263
car	0.0120	0.2941	0.2516	0.4224	0.0202	0.3618	0.4865	**0.6572**
Average	0.2441	0.2458	0.2459	0.3308	0.0557	0.4124	0.5227	**0.6223**

These methods were developed with a user-interface configured for operational testing. Using the system for testing affords not only measures of performance but measures of effectiveness [70, 71] such as timeliness for call-out to screen presentation. Likewise, the analysis supports and ontology [72,73] for semantic uncertainty understanding and awareness [74] for HLIF (user call out) to LLIF (track attribute) matching.

V. User Analysis Tool

The vsPlay user interface is designed for linking LLIF full motion video exploitation (i.e., Level 1 Fusion) with HLIF user semantic call-outs (i.e., Level 5 Fusion). Previous efforts included the JVIEW situation awareness tool [75]. There are three primary modules: manual event identification, object change detection, and moving target tracker. The change detection, tracking, and associated database functions are provided as part of a multi-intelligence system capability. The vsPlay primary user interface capabilities include exploitation aides such as tripwire, geofence selector, and time-space filtering. Basic full motion video (FMV) exploitation functions for analysis support include pause, play, fast-forward, and rewind. For general exploitation functions, there are capabilities to measure distance, image magnification, and video polarity change.

The screen layout (Fig. 9) for vsPlay includes a row of tabs, which contain drop down menu functions at the top of the screen. These drop-down menus include a number of icon functions and provide additional capabilities for the user to interact with the screen display layout and video based upon tab selection. To provide maximum situational awareness of the activity occurring within the video feed, an analyst should

set their screen layout to display the tracks pane (by selecting the show track list from the drop down menu), events pane (by selecting the show events list from the drop down menu), and the change detections pane (by selecting the Descriptors tab and ensuring the show alert list is activated). Each screen layout tab is described below:

- Video Tab - Controls the FMV feed such as start, stop, decrease speed;
- Tracks Tab – Provides on screen displays for the MTT such as track ID's, entity bounding boxes, object scores;
- Events Tab – Displays events that correlate with the FMV feed such as show all person/vehicle events;
- Descriptors Tab – Works with alerts to activate/deactivate or shows alerts;
- Regions Tab – Supports analyst ability to create/select/de-select or display regions of interest for activity or non-activity within the FMV feed;
- Call-out Tab – Lists the call-out from the speech to text in a text format to enable probabilistic graph matching; and.
- Tools Tab – Provides report generation, measuring/ruler, display change detection list functions.

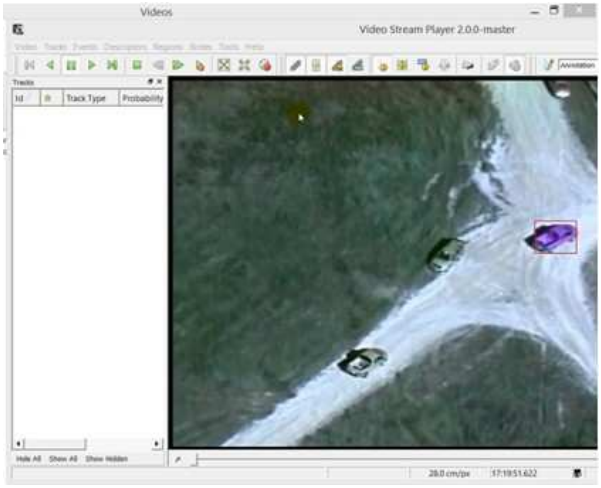


Figure 9 – The screen shot of a demo tracking in VsPlay interface.

Using the interface allows testing, analysis, and operational development. By investigating different information fusion functions, users are engaged in the development process for the adoption of tools for mission applications.

VI. CONCLUSIONS

In this paper, we evaluated a series of common video tracking methods for video-2-text (V2T) fusion in support of HLIF-LLIF coordination. The user is the primary stakeholder (e.g., Level 5 fusion) and can either manually do the task or make use of the information fusion tools such as video tracking and natural language processing. To support the adoption of such tools, testing is required in the work domain of the user. Thus, we explored the performance of V2T using representational data. Currently, the TLD and CT are shown to be the best choices in matching the call-out to the tracking results. The integration of voice and video tracking reduced the uncertainty of distinguishing the key targets through direction, attribute, category, and label designations.

Future efforts include high-performance cloud computing [76] using mapreduce [77] for real-time performance, coordination with wide-area motion imagery (WAMI) tracking [78, 79, 80] and advanced contextual tracking methods for target tracking and classification [81, 82]. Likewise, NLP tools can aid in the entity, event, and relationships of the textual analytics that can link the HLIF (semantic) and LLIF (object tracking) information.

ACKNOWLEDGEMENTS

Erik Blasch was supported under an AFOSR grant in Dynamic Data-Driven Applications Systems and support from AFRL. Additional technical thanks go to Steve Scott, Mike Hinman, Guna Seetharaman, Michael Schneider, Georgiy Levchuk, Hillary Holloway, Andrew Kaluzniacki, and Eric Jones, and William Pottenger.

REFERENCES

[1] E. Blasch, E. Bosse, and D. Lambert, *High-Level Information Fusion Management and Systems Design*, Artech House, Norwood, MA, 2012.
[2] E. Blasch, *Derivation of a Belief Filter for Simultaneous High Range Resolution Radar Tracking and Identification,* Ph.D. Thesis, Wright State University, 1999.
[3] S. Davey, D. Gray, R. Streit, "Tracking, association, and classification: A combined PMHT Approach," *Digital Signal Processing*, 12, 2, 372–382, 2002.
[4] E. Blasch, B. Kahler, "Multiresolution EO/IR Tracking and Identification" *Int. Conf. on Info Fusion,* 2005.
[5] D. Angelova, L. Mihaylova, "Joint target tracking and classification with particle filtering and mixture Kalman filtering using kinematic radar information, *Digital Signal Processing*, 16, 2, 180–204, 2006.
[6] M. Mei, G.-L. Shan, X. R. Li, "Simultaneous tracking and classification: A modularized scheme," *IEEE Transactions on Aerospace and Electronic Systems*, 43, 2, 581–599, 2007.
[7] C. Yang and E. Blasch, "Kalman Filtering with Nonlinear State Constraints," *IEEE Trans. Aerospace and Electronic Systems*, Vol. 45, No. 1, 70-84, Jan. 2009.
[8] O. Straka, J. Dunık, *et al*., "Randomized Unscented Transform in State Estimation of non-Gaussian Systems: Algorithms and Performance," *Int. Conf. on Info Fusion*, 2012.
[9] E. Blasch, J. Nagy, A. Aved, W. M. Pottenger, M. Schneider, R. Hammoud, E. K. Jones, A. Basharat, A. Hoogs, G. Chen, D. Shen, H. Ling, "Context aided Video-to-Text Information Fusion," *Int'l.. Conf. on Information Fusion*, 2014.
[10] E. Blasch, "Sensor, User, Mission (SUM) Resource Management and their interaction with Level 2/3 fusion" *Int. Conf. on Info Fusion*, 2006.
[11] E. Blasch, D. A. Lambert, P. Valin, M. M. Kokar, J. Llinas, S. Das, C-Y. Chong, and E. Shahbazian, "High Level Information Fusion (HLIF) Survey of Models, Issues, and Grand Challenges," *IEEE Aerospace and Electronic Systems Magazine,* Vol. 27, No. 9, Sept. 2012.
[12] P. Foo, G. Ng, "High-Level Information Fusion: An Overview," *J. Adv. Information Fusion*, Vol. 8 (1), June 2013.
[13] A. Steinberg, C. Bowman, *et al*., "Adaptive Context Assessment and Context Management," *Int'l Conf. on Information Fusion*, 2014.
[14] C. Yang and E. Blasch, "Fusion of Tracks with Road Constraints," *J. of. Advances in Information Fusion,* Vol. 3, No. 1, 14-32, June 2008.
[15] B. Kahler and E. Blasch, "Sensor Management Fusion Using Operating Conditions," *Proc. IEEE Nat. Aerospace Electronics Conf.*, 2008.
[16] J. M, Brown, D. Vernal, "Time-Dominant Fusion in a Complex World," *Trajectory Magazine*, Nov. 2014.
[17] M. R. Endsley, D. J. Garland, (Eds.) *Situation awareness analysis and measurement*. Mahwah, NJ: Lawrence Erlbaum Associates, 2000.
[18] E. Blasch, I. Kadar, J. Salerno, M. M. Kokar, S. Das, *et. al.*, "Issues and Challenges in Situation Assessment (Level 2 Fusion)," *J. of Advances in Information Fusion,* Vol. 1, No. 2, pp. 122 - 139, Dec. 2006.
[19] E. Blasch, I. Kadar, K. Hintz, J. Biermann, C. Chong, and S. Das, "Resource Management Coordination with Level 2/3 Fusion Issues and Challenges," *IEEE Aerospace and Electronic Systems Magazine,* Vol. 23, No. 3, pp. 32-46, Mar. 2008.
[20] A. Salfinger, D. Neidhart, W. Retschitzegger, *et al*, "SEM^2 Suite – Towards a Tool Suite for Supporting Knowledge Management and Situation Awareness Systems," *IEEE IRI*, 2014.

[21] R. T. Antony, J. A. Karakowski, "First-Principle Approach to Functionally decomposing the JDL Fusion Model: Emphasis on Soft Target Data," *Int'l Conf. on Information Fusion*, 2008.

[22] M. Pravia, O. Babko-Malaya, M. Schneider, J. White, C. Chong and A. Willsky, "Lessons learned in the creation of a data set for hard/soft information fusion," *International Conf. on Information Fusion*, 2009.

[23] E. Blasch, É. Dorion, *et al*., "Ontology Alignment in Geographical Hard-Soft Information Fusion Systems," *Int'l. Conf. on Info Fusion*, 2010.

[24] S. Acharya, M. Kam, "Evidence combination for hard and soft sensor data fusion," *Int'l. Conf. on Information Fusion*, 2011.

[25] J. L. Graham, D. L. Hall, J. Rimlan. "A synthetic dataset for evaluating soft and hard fusion algorithms," *Proc. SPIE*, 8062, 2011.

[26] A. Preece, D. Pizzocaro, D. Braines, D. Mott, G. de Mel, and T. Pham, "Integrating hard and soft Information Sources for D2D Using Controlled Natural Language," *Int'l Conf. on Information Fusion*, 2012.

[27] E. Blasch, G. Seetharaman, K. Palaniappan, H. Ling, and G. Chen, "Wide-Area Motion Imagery (WAMI) Exploitation Tools for Enhanced Situation Awareness," *IEEE App. Imagery Pattern Rec.Workshop*, 2012.

[28] E. Blasch, Z. Wang, H. Ling, K. Palaniappan, G. Chen, D. Shen, A. Aved, *et al*., "Video-Based Activity Analysis Using the L1 tracker on VIRAT data," *IEEE App. Imagery Pattern Rec. Workshop*, 2013.

[29] R. I. Hammoud, *et al*., "Multi-Source Multi-Modal Activity Recognition in Aerial Video Surveillance," *CVPR Workshop*, 2014.

[30] R. I. Hammoud, C. S. Sahin, E. P. Blasch, B. J. Rhodes, and T. Wang, "Automatic Association of Chats and Video Tracks for Activity Learning and Recognition in Aerial Video Surveillance," *Sensors*, 14, 19843-19860, 2014.

[31] J. Garcia Herro, L. Snidaro, and I. Visentini, "Exploiting context as a binding element for multi-level fusion," *Int'l Conf. on Info. Fusion*, 2012.

[32] E. Blasch, A. Steinberg, S. Das, J. Llinas, *et al*., "Revisiting the JDL model for information Exploitation," *Int'l Conf. on Info Fusion*, 2013.

[33] X. S. Zhou, T. S. Huang, CBIR: from low-level features to high-level semantics," *Proc, SPIE*, Vol. 3794, 2000.

[34] Y. Liu, D. Zhang, G. Lu, W. Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, 2007.

[35] S. Ezekiel, M. G. Alford, D. Ferris, E. Jones, A. Bubalo, *et al*., "Multi-Scale Decomposition Tool for Content Based Image Retrieval," *IEEE Applied Imagery Pattern Recognition Workshop*, 2013.

[36] A. Hoogs, J. Mundy, and G. Cross, "Multi-Modal Fusion for Video Understanding," *IEEE Applied Imagery Pattern Rec. Workshop*, 2001.

[37] A. Hoogs, J. Rittscher, G. Stein and J. Schmiederer, "Video Content Annotation Using Visual Analysis and a Large Semantic Knowledgebase," *IEEE CVPR*, 2003.

[38] T. Wang, Z. Zhu and R. Hammoud, "Audio-Visual Feature Fusion for Vehicles Classification in a Surveillance System," *IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR), 2013.

[39] M. T. Chan, A. Hoogs, J. Schmiederer, M. Petersen, "Detecting Rare Events in Video Using Semantic Primitives with HMM," *International Conference on Pattern Recognition*, 2004.

[40] A. Basharat, A. Gritai, M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," *IEEE CVPR*, 2008.

[41] P. Lamere, P. Kwok, W. Walker, E. B. Gouvêa, R Singh, "Design of the CMU sphinx-4 decoder," Interspeech, 2003.

[42] K Seymore, S Chen, *et al*., "The 1997 CMU Sphinx-3 English broadcast news transcription system," *DARPA Speech Rec. Workshop*, 1998.

[43] http://cmusphinx.sourceforge.net/wiki/tutoriallm

[44] T. Wu and W. Pottenger, "A semi-supervised active learning algorithm for information extraction from textual data: Research Articles," *Journal of the American Society for Information Science and Technology - Intelligence and Security Informatics*, vol. 56, no. 3, pp. 258-271, 2005.

[45] M. C. Ganiz, C. George, and W. M. Pottenger, "Higher Order Naïve Bayes: A Novel Non-IID Approach to Text Classification," *IEEE Tr. on Knowledge and Data Engineering*, vol. 23 (7): 1022-1034, July, 2011.

[46] C. Nelson, H. Keiler H., W.M. Pottenger, Modeling Microtext with Higher Order Learning, *AAAI Spring Symposium*, 2013.

[47] A. Panasyuk, *et al*., "Extraction of Semantic Activities from Twitter Data," *Semantic Technol. for Intelligence, Defense, and Security*, 2013.

[48] B. Liu, *et al*., "Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier," *IEEE Intl Conf. on Big Data*, 2013.

[49] H. Ling, L. Bai, *et al*., "Robust infrared vehicle tracking across target pose change using L1 regularization," *Int'l Conf. on Info. Fusion*, 2010.

[50] X. Mei, H. Ling, *et al*., "Minimum Error Bounded Efficient L1 Tracker with Occlusion Detection," *IEEE Comp. Vision and Pattern Rec.*, 2011.

[51] X. Zhang, W. Li, W. Hu, H. Ling, *et al*., "Block covariance based L1 tracker with a subtle template dictionary," *Pattern Recognition*, 2012.

[52] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Efficient Minimum Error Bounded Particle Resampling L1 Tracker with Occlusion Detection," *IEEE Trans. on Image Processing* (T-IP), Vol. 22 (7), 2661 – 2675, 2013.

[53] K. Zhang, L. Zhang, and M.-H. Yang, "Real-Time compressive tracking," *European Conf. on Computer Vision*, pp. 864-877, 2012.

[54] Z. Kalal, K. Mikolajczyk, J. Matas, "Tracking-Learning-Detection," *IEEE T. PAMI*, Vol. 6, No. 1, Jan. 2010.

[55] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 798–805, 2006.

[56] S. Hare, A. Saffari, P. Torr, "Struck: Structured output tracking with kernels," in *ICCV,* 2011.

[57] X. Jia, H. Lu, M. H., Yang, "Visual tracking via adaptive structural local sparse appearance model," in *CVPR,* 2012.

[58] S. Yan, H. Wang, "Semi-supervised learning by sparse representation," in *SIAM Int'l Conf. on Data Mining*, 2009.

[59] T. Zhang, B. Ghanem, S. Liu, N. Ahuja, "Robust visual tracking via multi-task sparse learning," in *CVPR,* 2012.

[60] H. Grabner, M. Grabner, H. Bischof, "Real-time tracking via on-line boosting," in *BMVC,* 2006.

[61] N. C. Oza, "Online bagging and boosting," *International Conf. on Systems, Man, and Cybernetics*, Vol. 3, pp. 2340-2345, 2005.

[62] S. Avidan, "Ensemble Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 29, Issue 2, pp. 261-271, 2007.

[63] I. Visentini, L. Snidaro, G. L. Foresti, "Selecting the classifiers by F-Score for real-time video tracking," *Int. Conf. on Info. Fusion*, 2010.

[64] I. Visentini, L. Snidaro, G. H. Foresti "Cascade online boosting," *J. of Real-Time Image Processing*, Vol. 5, Issue 4, pp. 245-257, 2010.

[65] H. Grabner, C. Leistner, H. Bischof, "Semi-supervised on-line boosting for robust tracking," *ECCV,* 2008.

[66] P.K. Mallapragada, R. Jin, A. K. Jain, Y. Liu, "Semiboost: Boosting for semi-supervised learning," *Trans.on PAMI* 31, 2000–2014, 2009.

[67] B. Babenko, M.H. Yang, S. Belongie, "Visual tracking with online multiple instance learning," in *CVPR,* 2009.

[68] X. Li, C. Shen, A. Dick, A. van den Hengel, "Learning compact binary codes for visual tracking," in *CVPR,* 2013.

[69] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.C. Chen, J. T. Lee, S., Mukherjee, J. K. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, D. Song, A. Fong, A. R. Chowdhury, M. Desai, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR*, 2011.

[70] E. Blasch, P. Valin, and E. Bossé, "Measures of Effectiveness for High-Level Fusion," *Int'l Conf. on Info Fusion*, 2010.

[71] E. Blasch, R. Breton, and P. Valin, "Information Fusion Measures of Effectiveness (MOE) for Decision Support," *Proc. SPIE* 8050, 2011.

[72] P. C. G. Costa, K. B. Laskey, E. Blasch and A-L. Jousselme, "Towards Unbiased Evaluation of Uncertainty Reasoning: The URREF Ontology," *Int. Conf. on Info Fusion*, 2012.

[73] E. Blasch, K. B. Laskey, A-L. Jousselme, V. Dragos, P. C. G. Costa, and J. Dezert, "URREF Reliability versus Credibility in Information Fusion (STANAG 2511)," *Int'l Conf. on Info Fusion*, 2013.

[74] E. Blasch *et al*., "DFIG Level 5 (User Refinement) issues supporting Situational Assessment Reasoning," *Int. Conf. on Info Fusion,* 2005.

[75] E. Blasch, "Enhanced Air Operations Using JView for an Air-Ground Fused Situation Awareness UDOP," *AIAA/IEEE Digital Avionics Systems Conference*, 2013.

[76] B. Liu, Y. Chen, E. Blasch, K. Pham, D. Shen, and G. Chen, "A holistic cloud-enabled robotics system for real-time video tracking application," *Int'l Workshop on Enhanced Cloud Fusion*, w/ *Future Info. Tech*, 2013.

[77] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *IEEE conference on Symposium on Operating Systems Design & Implementation*, vol.6, 2013.

[78] P. Liang, *et al*., "Multiple Kernel Learning for Vehicle Detection in Wide Area Motion Imagery," *Int. Conf. on Info Fusion*, 2012.

[79] J. Gao, H. Ling, *et al*., "Pattern of life from WAMI objects tracking based on visual context-aware tracking and infusion network models," *Proc. SPIE*, Vol. 8745, 2013.

[80] P. Liang, H. Ling, E. Blasch, G. Seetharaman, D. Shen, G. Chen, "Vehicle Detection in Wide Area Aerial Surveillance using Temporal Context," *Int'l Conf. on Info Fusion*, 2013.

[81] E. Blasch, J. Garcia Herrero, L. Snidaro, J. Llinas, G. Seetharaman, K. Palaniappan, "Overview of contextual tracking approaches in information fusion ," *Proc. SPIE*, Vol. 8747, 2013.

[82] E. Blasch, J. Dezert, B Pannetier, "Overview of Dempster-Shafer and Belief Function Tracking Methods," *Proc. SPIE*, Vol. 8745, 2013.