A Cloud Infrastructure for Target Detection and Tracking Using Audio and Video Fusion

Kui Liu
Intelligent Fusion Technology
Germantown, MD 20876
kui.liu@intfusiontech.com

Bingwei Liu
Binghamton University
Binghamton, NY 13902
bingweiliu11@gmail.com

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

Dan Shen, Zhonghai Wang
Intelligent Fusion Technology
Germantown, MD 20876
{dshen, zwang}@intfusiontech.com

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

Genshe Chen
Intelligent Fusion Technology
Germantown, MD 20876
gchen@intfusiontech.com

Abstract

This paper presents a Cloud-based architecture for detecting and tracking multiple moving targets from airborne videos combined with the audio assistance, which is called Cloud-based Audio-Video (CAV) fusion. The CAV system innovation is a method for user-based voice-to-text color feature descriptor track matching with an automated hue feature extraction from image pixels. The introduced CAV approach is general purpose for detecting and tracking different valuable targets' movement for suspicious behavior recognition through multi-intelligence data fusion. Using Cloud computing leads to real-time performance as compared a single machine workflow. The obtained multiple moving target tracking results from airborne videos demonstrate that the CAV approach provides improved frame rate, enhanced detection, and real-time tracking and classification performance under realistic conditions.

Keywords: Audio and video fusion, cloud infrastructure, target detection, target tracking and classification.

1 Introduction

The detection of moving objects is critical in many defense and security applications, where target detection is usually performed for target tracking and automatic target recognition, classification, and identification [1, 2, 3, 4, 5]. Target tracking, as low-level information fusion (LLIF), is utilized in high-level information fusion (HLIF) situation awareness for user monitoring and area surveillance [6]. Many challenges exist in HLIF such as semantic, knowledge, and information representation and management [7, 8]. For both robust target detection and tracking, there is a need to improve performance which can be aided by other sensors.

Many videos used for surveillance applications are outdoor videos whose quality may be degraded by various noisy sources, such as atmospheric turbulence, sensor platform scintillation, etc. To deal with the video quality from areal images for movement detection approaches include a layered sensing approach [9], image fusion [10, 11], and use of descriptors [12]. Moving objects may be very small occupying only a few pixels, which makes target detection very challenging such as in Wide-Area Motion Imagery [13]. Under this circumstance, existing approaches may generate significant amount of false alarms of detecting things that are not targets.

Target tracking has been extensively investigated as described in survey papers [14, 15]. For this paper, we extend optical flow methods []16, 17, 18]. Many developments in video tracking are from indoor videos with large objects; however, research with outdoor scenes must account for lighting, perspective, and obscuration variation [19]. As one of the major techniques, optical flow based approaches have been widely used for target detection. There are two classic methods of optical flow computation in computer vision: Gunnar Farneback (GF) method [20] and Lucas-Kanade (LK) method [21,22,23]. Both GF and LK are based on the two-frame difference algorithms. Since the LK method needs the construction of a pyramid model in sparse feature scale space and iterative computational updating at successively finer scales, we focus on the GF method for a dense optical flow computation in our framework.

The paper develops a framework that utilizes information from an optical flow generator (OFG) [24] and a color-based *active-learning histogram matcher* (AHM) to increase the detection robustness for real-time applications. The OFG and AHM processes are deployed in such a way that they act in a complementary manner to rule out the erroneous data that may get detected by each process individually. Together, OFG/AHM capture the potential and valuable moving targets. For AHM, the Hue component of color in HSV (Hue-Saturation-Value) space is used for tracking. A template Hue histogram associated with a typical color is initialized at the start of the detection framework. From the start of detection to the start of tracking, the template Hue histograms keep actively calibrated according to the slight color changes of the targets. The histogram within a window is used as the target tracking feature which robustly increases target detection and tracking [25, 26] of which other templates could apply [27].

The use of *multisensory data* can increase target detection and tracking such as video combined with audio information [28] or signals of opportunity [29]. In some applications, a user audibly calls-out things in the image, which can be used with the tracking results for which limited demonstrations have been reported in the literature. Audio information can come from two sources: the target and/or an observer. Target classification based on audio sources is possible within a certain distance [30]. Audio information extracted from user's speech is combined to generate the features of targets if interest (size, color and/or moving direction). Recent efforts have utilized speech to text, call-out information, and chat for full motion video target detection [31, 32]. The use of the semantic information from audio results provides context to the track [33]. Collection and retrieval [34] of audio-video data is enhanced with cloud technology.

Cloud technology has become increasingly important for multisensory and image processing systems [35, 36, 37]. A Cloud-based framework uses Cloud computing [38], is built using Xenserver [39]. A web portal in the Cloud is provided for the user. From this web portal, the user chooses algorithms, datasets and system parameters such as desired processing frame rate, maximum frame duration, etc. A *controller* in the Cloud will then decide the amount of computing resources to be allocated to the task in order to achieve the user's requirement of performance. Inside the Cloud, there are several Virtual Machines (VMs) running on each physical machine. Each of these VMs are capable of running various detection, registration, and tracking algorithms. One registration algorithm is usually run by several threads in one or more VMs. Another contribution is that a non real-time algorithm achieves real-time performance based on the application of a Cloud computing infrastructure. The processing speed is drastically improved by parallel implementation and elastic resource usage of Cloud computing. Rather than colors, it is worth noting that the proposed CAV infrastructure is general purpose in the sense that it can be used for detecting and tracking different valuable targets' movement based on many object features such as size, frequency, and moving direction.

The rest of the paper is as follows. Sect. 2 provides the framework. Sect. 3 discusses the cloud infrastructure, AHM, and target track generation. Sect. 4 provides tracking and classification results and Sect. 5 conclusions.

2 Overview of the CAV Framework

Until recently, typical experiments conducted for video tracking applications reported in the literature mostly involve constant lighting conditions and relatively stable image collections. However, in practice, image information captured under realistic lighting conditions degrades considerably with a moving background and unstable image collections. As a result, target detection and tracking becomes quite challenging and many false alarms are generated by existing approaches. The Cloud-based Audio-Video (CAV) fusion system consists of Cloud-based image alignment, online target color calibration, optical flow detection and histogram matching for target detection and tracking. Key elements are:

Histogram Matching: a template Hue histogram associated with specific pixel color in a window is used as a matching feature to locate or redetect the target location estimate when its tracking is hampered by the environment. The candidate window with the histogram is generated from the optical flow field between two aligned consecutive frames. Any optical flow blob with a similar histogram will be considered as a strong potential candidate. The template Hue histogram will be tuned according to the strong potential candidates. The goal of the online target color calibration is to adapt our subsequent color processing to the color characteristic of the light source under which images are captured. This technique has been previously used quite successfully for hand detection in [40] in order to cope with unknown color characteristics of various light sources encountered in practice, calibration is performed from the beginning when the system is turned on or the tracked target is missing till the target is located or redetected. It involves updating the histogram in HSV color space to represent the color characteristics of the target being captured in an online or on-the-fly manner. The calibration is achieved easily by a simply weighted linear addition between the template histogram and the strong candidate histogram. Representative color pixels of the target are collected within the window and the histogram is calibrated and used as a feature to perform target locating or redetection. Thus the developed solution is designed to be robust to different background and lighting conditions. More details of the online color calibration can be found in [41].

Cloud Architecture: CAV utilizes a Cloud infrastructure to run image alignment or registration processes in parallel and thus a non real-time algorithm can perform at real-time frame rate. The well-known attractive features of Cloud computing include on-demand scalability of highly available and reliable pooled computing resources [42], secure access to metered services from anywhere, and displacement of data and services from inside to outside the organization.

Tracking and Registration: the target track generation based on the image alignment and registration enables further analysis and estimation to the target behavior, such as pose estimation [43]. The image alignment is a process of transforming different sets of data into one coordinate system, since the homography matrix generated from the image alignment can further achieve the rotation and translation matrices. With the rotation and translation matrices, the coordinates in the previous frames are projected into the current frames and form a sequential track of the target. The image alignment consumes most of the computation in the CAV framework, thus this step is adaptively allocated in a Cloud infrastructure.

3 CAV Infrastructure

3.1 Cloud infrastructure based image alignment

Fig. 1 depicts a high level view of a host in the Cloud system. XenServer [39] serves as the Virtual Machine (VM) Manager, or hypervisor, of each physical server in the Cloud. All components of the same task have access to shared storage in the Cloud. The user only interacts with the system through the Web graphical user interface (GUI). The user's computing requests from audio processing are passed to controller for further processing. The *Controller* assigns an appropriate number of jobs to VM workers for each request. Each VM worker runs one or more the assigned tasks (detector, tracker and register) and sends the results to the controller [44]. The web GUI will then display the processing result in real-time once the backend processing starts. The *Monitor* uses a database to store real-time performance of all tasks in the system [45]. It can also monitor the Cloud's performance such as average CPU (central processing unit) load and memory usage. The user can choose what metrics to be displayed on the web GUI and can call other visual analytic tools such as a rendering on a ground plane, road network [46] or 3-dimensional trajectory [47].

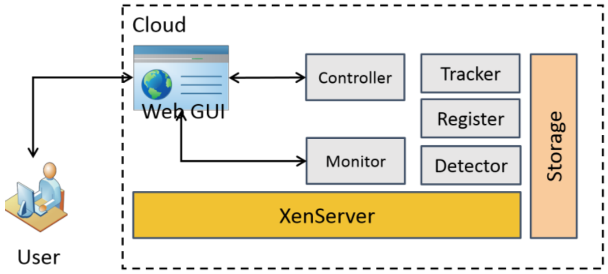


Figure 1. Cloud Infrastructure

3.1.1 Interactive Tracking module

The interactive tracking module provides the options to choose various detector, register, and tracker algorithms and the desired processing frame rates as parameters before the system initializes. Each combination of detector, register, tracker and frame rate consists of a configuration of the system. Once a configuration is chosen by the user, the system will start to process the video sequence once the processing command is received from the web GUI [48]. A target on the initial frame can also be manually identified by the user as an alternative to a detector that selects the pixel boundary representing the target.

In target detection and tracking scenarios, the key components of the CAV system perform the following tasks:

- *User*. The user chooses configuration of the system, can verbally designate color of the object, and sends commands to the system to initiate a task.
- *Web GUI*. The web GUI communicates with the user and by receiving input commands, displays processing results and presents analytical system performance.
- *Controller*. The Controller receives commands from the web GUI, makes decisions on how many resources are needed to satisfy the required performance, assigns jobs and tasks to virtual machines in the Cloud, combines processing results of different components into resultant frames, calculates processing speed in real-time, and informs the web GUI processing results.
- *Monitor*. The Monitor collects performance metrics such as processing speed and system load, and provides a web GUI query service when there is a need to display the metrics.
- *Virtual Machines (VMs).* Each VM can act as a detector, register or tracker in the system. The actual roles a VM will perform are decided by the controller.

3.1.2 Voice Command Tracking

The voice command tracking module is able to receive commands from the user's speech in real-time. Fig. 2 shows the overall module data flow. The system starts and enters "Play Video", displaying the video sequence on the web GUI. The web GUI keeps checking if the current frame is the last frame. If it is, the web GUI stops at the last frame, otherwise it listens to any voice signal from the user. The user can trigger a voice command at any time before the last frame is displayed. When speech signals are received from the user's microphone, the system runs speech recognition to identify the command. If a command is recognized, the command is sent to a detector running in the Cloud, entering "detect".

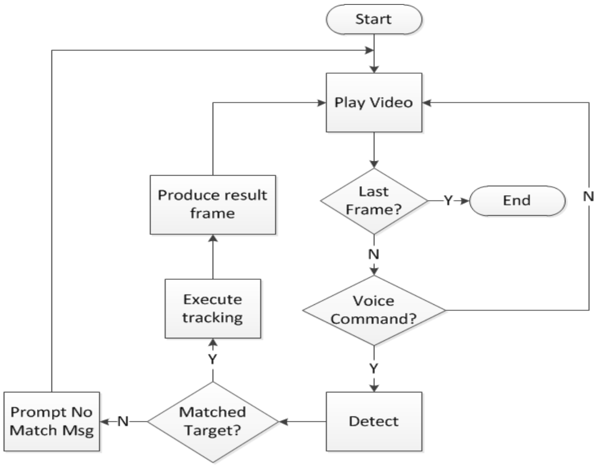


Figure 2. Voice Command Tracking Module Data Flow

The *Matcher* utilizes the speech command converted to text from which the text color semantic designation is used with a look-up-table to get the physical hue representation. The semantic hue values are then matched with the hue values from detector processing of the image pixels. If there is a good match, a target tracker will take over the execution, in coordinating with multiple registers determined by the Controller. The Controller will then combine processing results from trackers and registers and produce the resultant annotated video sequence. The web GUI will display processed frames once the first processed frame is available. If there is no matched target in the current frame, the web GUI will prompt a message to the user and continue to display the next frame.

Fig. 10 (*located at the end of the article*) shows the web GUI when the user triggers a voice command to track the car in the video sequence. As can be seen, there are three windows in the GUI (see Fig. 10). The top left window displays the original frames when no voice command is received. Once a voice command is triggered, the frames in this window will be replaced by processed frames with rectangle box on the target. The top right window is the optical flow window, synchronized with the current video frame. The window on the bottom right is the histogram of current frame used by the detector to detect the target. The details of the three windows will be extended in the following sections.

3.1.3 System Monitoring Module

Fig. 3 shows an instance of the System Monitoring Module (SMM) or the dashboard on the web GUI. On the dashboard page, the user chooses one or more metrics, provided by the monitor, to be displayed. The selected charts are then created on the fly by the web GUI. Once all charts are created, the web GUI starts to query with the SMM to retrieve all data for each chart. In Fig. 3, the charts provide real-time monitoring metrics the user chooses, updated every five seconds. For example, the top left chart depicts the Cloud host average CPU utilization, which is an indication of the fulfillment of service level agreement (SLA). The top right chart illustrates the real-time processing performance, the frame per second (fps) processing rate. As a baseline, the fps of all algorithms running on a single machine (CPU) are provided. The bottom left chart shows the number of register threads/processes running when executing the task.

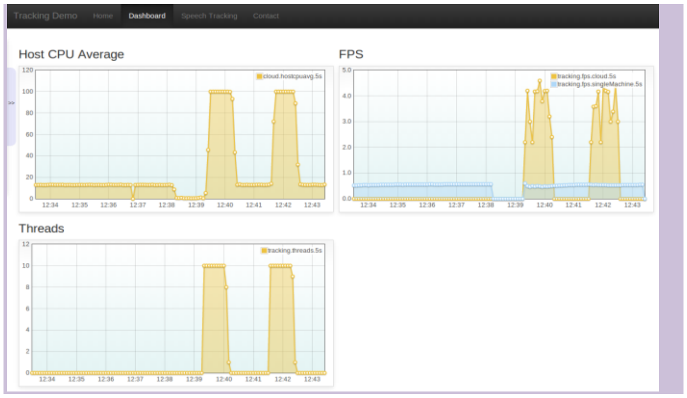


Figure 3. System Monitoring Web GUI

3.2 Active-learning Histogram Matcher (AHM)

The existing feature descriptors including SIFT (Scale-invariant feature transform) [49], SURF (Speeded Up Robust Features) [50] and Harris Corner [51] create real-time challenges due to their computational complexity. To have a more computationally efficient matching, an active-learning histogram is adopted in the proposed CAV system. In this paper, the color depth of Hue component in HSV is set to 0 to 255 and they are divided into 16 levels with the same step width, which means each level has 16 color bits.

As can be seen in Fig. 4, the two bargraphs respectively represent two optical colors (e.g., silver and red) pixel distributions in Hue component space. The color distribution of the template histograms is considered as 16-component vectors. (e.g., silver [0, 20, 45, 38, 75, 18, 0, 0, 0, 0, 0, 0, 0, 0, 0] and red [60, 28, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 110]). Note that the silver and red template hue histograms are examples of the general purpose matching in the sense that it is applicable to other target *features histograms* such as attributes (e.g., size, shape, color), kinematics (e.g., pose/direction, velocity), and identification (e.g., allegiance, clustering) to match the target.

Leveraging proven computer vision methods for parameter updates, the Exponentially Weighted Moving Average (EWMA) is used for histogram matching:

$$C_{Template} = \alpha * C_{Template} + \beta * C_{Candidate} \qquad (1)$$

As can be seen in equation (1), the update to the template hue component histogram is performed iteratively in each frame by the weighted linear addition. The given $C_{Template}$ is used to represent the current template histogram, $C_{Candidate}$ can be considered for the strong candidate histogram, and α and β are two weighting factors of the linear additive calibration. In this paper, we choose α as 0.9 and β as 0.1 to update the current template Hue component histogram. The candidate histogram windows are generated from the optical flow field between two aligned consecutive frames.

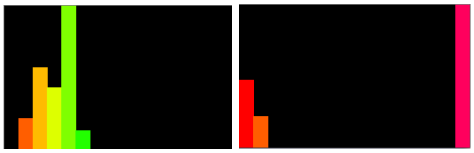


Figure 4. Template Hue histogram used for target color matching (e. g., silver and red)

As shown in Fig. 5, three white blobs are generated from the optical flow field after image alignment and the bounding rectangles are considered as candidate windows. Fig. 6 depicts the original frame of the image. The candidate windows with the most similar hue histogram are selected as the true target and these candidate hue histograms are used to calibrate the current template histogram. As can be seen in Fig. 5, there are three strong candidate windows in this frame where the right-hand side candidate is matched with the red template, the left-hand side matched with the silver template, while the one in the middle was rule out because of the low likelihood of the histogram.

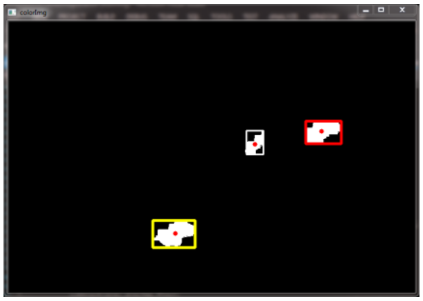


Figure 5. Optical flow field generation Candidate windows

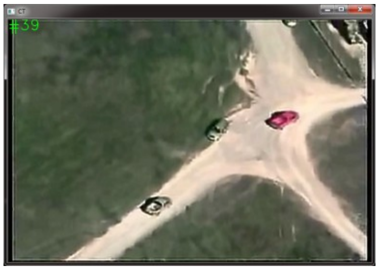


Figure 6. Original frame in VideoSequence1

In the CAV framework, the vectors consist of 16 consecutive numbers of the bars in the histogram are considered as the color feature distribution of the windows. A Euclidean distance method or the Dynamic Time Warping (DTW) algorithm can serve as the histogram matcher. The DTW was selected as this algorithm is capable of generating the dynamic distance between an unknown signal and a set of reference signals in a computational efficient manner. The details of the DTW algorithm are discussed in [52, 53, 54]. The sample signal comes from the current template histogram vector.

Fig. 7 depicts a flowchart which describes how the CAV system works. In Fig. 7, a video image frame available in the Cloud is selected at the beginning. A user audio speech, or voice command, calls out the color desired of the target and the audio signal is converted into a text input which is used to describe the features of the targets. For example, the color text designation of the target is converted into a hue representation using the gamut colorspace naming convention. In the CAV system, color distribution of the histogram is used to recognize and locate the targets of interest. Once the command has been given, the detector as shown in the red boundary of Fig 7 is initialized. The first step in the detector is image alignment (Homography Matrices generation) and registration. The registration step is the most computation-consuming step thus it is applied in Cloud and performed in a computational efficient manner. Based on the homography matrices and the registered frames, the optical flow field is generated.

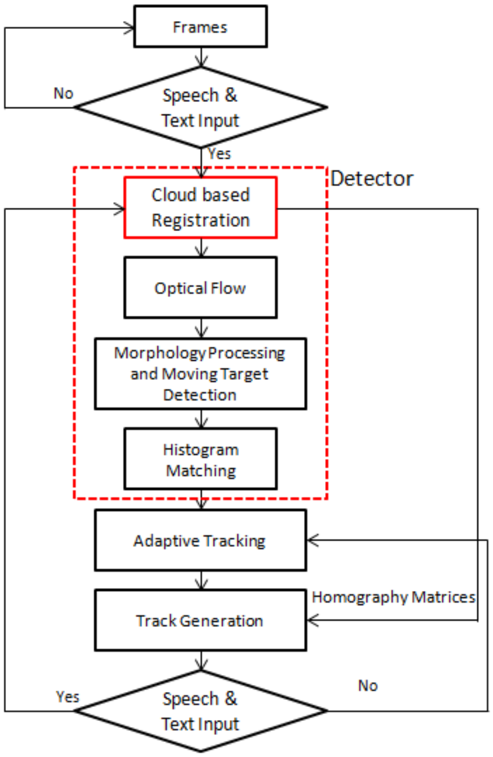


Figure 7. Flow diagram of the proposed CAV target detection and tracking system.

In Fig. 7, morphology processing which includes a Closing and Opening operations using a 3*3 square structuring element applied on the optical flow field. The leftover blobs of the morphology processing are considered as strong candidate contours of the target. Then the histogram templates of colors are used to match and recognize the valuable targets among the candidates. Once the target is recognized, the tracker will be initialized. In the current CAV system, either a L1-tracker [55] or a Compressive tracker [56] are options used to track the targets of interest. From homography matrices generated from the Cloud infrastructure for image alignment, the prior object tracks are projected in the current frame. At last step, in case that the system cannot correctly locate the target, the speech command or text input is used to redefine or relocate the targets of interest.

3.3 Target Track generation

As the homography matrices of the frames are generated from the Cloud infrastructure for image alignment, the rotation and translation matrices derived from homography matrices facilitate the integration of all image sequences possible. Thus, prior tracks of the objects are projected into the current fame. The Homography matrix **H**, generated from image alignment consists of 9 components.

$$\mathbf{H}=\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (2)$$

For homography matrix estimation, the *Direct Linear Transformation Algorithm* [57] is used. In the projective plane $\mathbb{P}^2$, the homography matrix estimation can be simplified as that given a set of points x_i and a corresponding set $\mathrm{x}_i^{'}$, compute the geometrical transformation which takes each x_i to $\mathrm{x}_i^{'}$. The transformation gives $\mathbf{H}\mathrm{x}_i \rightarrow \mathrm{x}_i^{'}$. For the 3-vector $\mathrm{x}_i^{'}$, the equation may be expressed in terms of the vector cross product as $\mathrm{x}_i^{'}\times\mathbf{H}\mathrm{x}^{'}=0$. Let $\mathrm{x}_i=(x_i, y_i, w_i)^{\mathrm{T}}$ and $\mathrm{x}_i^{'}=(x_i^{'}, y_i^{'}, w_i^{'})^{\mathrm{T}}$, the cross product is given as:

$$\mathrm{x}_i' \times \mathbf{H}\mathrm{x}' = \begin{pmatrix} y_i'\mathrm{h}^{3\mathsf{T}}\mathrm{x}_i - w_i'\mathrm{h}^{2\mathsf{T}}\mathrm{x}_i \\ w_i'\mathrm{h}^{1\mathsf{T}}\mathrm{x}_i - w_i'\mathrm{h}^{3\mathsf{T}}\mathrm{x}_i \\ x_i'\mathrm{h}^{2\mathsf{T}}\mathrm{x}_i - y_i'\mathrm{h}^{1\mathsf{T}}\mathrm{x}_i \end{pmatrix} \quad (3)$$

which can be written in the form

$$\begin{bmatrix} 0^{\mathsf{T}} & -w_i'\mathrm{x}_i^{\mathsf{T}} & y_i'\mathrm{x}_i^{\mathsf{T}} \\ w_i'\mathrm{x}_i^{\mathsf{T}} & 0^{\mathsf{T}} & -x_i'\mathrm{x}_i^{\mathsf{T}} \\ -y_i'\mathrm{x}_i^{\mathsf{T}} & x_i'\mathrm{x}_i^{\mathsf{T}} & 0^{\mathsf{T}} \end{bmatrix} \begin{pmatrix} \mathrm{h}^1 \\ \mathrm{h}^2 \\ \mathrm{h}^3 \end{pmatrix} = 0 \quad (4)$$

It has the form $\mathrm{A}_i\mathbf{h} = 0$, where A_i is a 3×9 matrix, **h** is a 9-vector which makes up the entities of homography matrix **H.**

Algorithm Direct Linear Transformation

Input: $m(m>4)$ pairs of Correspondence points$\{\mathrm{x}_i \leftrightarrow \mathrm{x}_i^{'}\}$

1. Normalization of x: Compute a similarity transformation **P**, consisting a translation and scaling. **P** takes the points x_i to a new set $\tilde{\mathrm{x}}_i$ such that the centroid of the points $\tilde{\mathrm{x}}_i$ is the coordinate origin $(0,0)^{'}$, and the average distance from the origin is $\sqrt{2}$.
2. Normalization of $\mathrm{x}_i^{'}$: Computer a similarity transformation $\mathbf{P}^{'}$,

transform points x_i' to $\tilde{x}_i'$.

3. For each correspondence $x_i' \leftrightarrow \tilde{x}_i'$, the matrix A_i is computed through,

$$A_i h = \begin{pmatrix} y_i'\hat{w}_i' - w_i'\hat{y}_i' \\ w_i'\hat{x}_i' - x_i'\hat{w}_i' \end{pmatrix}$$

4. Assemble the m 2×9 matrices into a $2m$×9 matrix A.
5. Apply singular value decomposition (SVD) to matrix A. The unit singular vector corresponding to the smallest singular value is the solution h.
6. Let vector x represent the measured image coordinates; $\hat{x}$ represent estimated value of the points and $\bar{x}$ the ground truth coordinate of the points. A vector can be defined $(\hat{x}_i', \hat{y}_i', \hat{w}_i')^T = \hat{x}_i' = \mathbf{H}\bar{x}_i$. The matrix $\tilde{\mathbf{H}}$ is generated from $\tilde{h}$ as

$$\begin{bmatrix} 0^T & -w_i'x_i^T & y_i'x_i^T \\ w_i'x_i^T & 0^T & -x_i'x_i^T \end{bmatrix} \begin{pmatrix} h^1 \\ h^2 \\ h^3 \end{pmatrix} = 0$$

7. De-normalization: Set $\mathbf{H} = (\mathbf{P}')^{-1}\ \tilde{\mathbf{H}}\mathbf{P}$.

Output: 2D homography matrix **H** such that $x_i' = \mathbf{H}x_i$.

Since the air-ground transformations between frames are not only affine, but also projective, the geometric transformation can be represented by a matrix form $\mathbf{H} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}$.

Thus we have:

$$\begin{bmatrix} \mathbf{x}_{\text{new}} \\ \mathbf{y}_{\text{new}} \\ \mathbf{1} \end{bmatrix} = \frac{\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{1} \end{bmatrix}}{\begin{bmatrix} g & h & 1 \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{1} \end{bmatrix}} \qquad (5)$$

Equation (5) is an iterative process where

$$\mathbf{x}_{\text{new}} = \frac{a\mathbf{x} + b\mathbf{y} + c}{g\mathbf{x} + h\mathbf{y} + 1}, \text{ and } \mathbf{y}_{\text{new}} = \frac{d\mathbf{x} + e\mathbf{y} + f}{g\mathbf{x} + h\mathbf{y} + 1} \qquad (6)$$

where $\mathbf{x}_{new}$ and $\mathbf{y}_{new}$ respectively represent the track generated in the current frame which consist of the registered previous track **x** and **y**. In the Homography matrix **H**, $\begin{bmatrix} a & b \\ d & e \end{bmatrix}$ represents the rotation matrix **R**, while $\begin{bmatrix} c \\ f \end{bmatrix}$ represents the translation matrix **t**. Rotation and translation matrices are called the matrix of extrinsic parameters. They are used to describe the camera motion around a static scene. The matrix [g h] gives the projective parameters, which describe the radial distortion and slight tangential distortion.

4 Experiments and Results

As can be seen in Fig. 8, the registered tracks are displayed in the current frame, which record the motion curves of the objects of interest. In the CAV system, two registration algorithms are applied: RANSAC (Random Sample Consensus) and LMEDS (Least Median of Squares). The tracks of RANSAC are shown in yellow and cyan, while the ones used LMEDS are shown in red and pink. Normally at the beginning of the work flow, one of these algorithms is selected to align the image sequences.

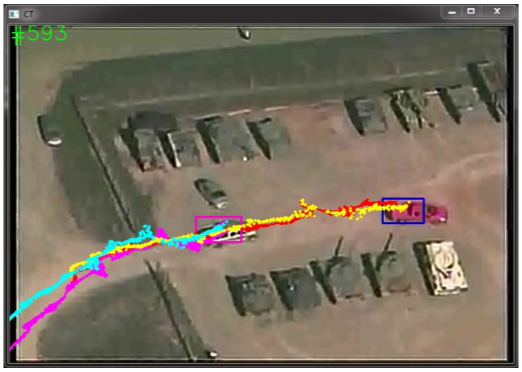


Fig. 8 Frame with the registered track in VideoSequence1

In our experiment, the very recent virtual infrastructure machine Xenserver [39] was applied. In this work, 2 virtual machines (8 CPUs) in Xenserver were utilized. The platform of CPU is Intel core i7-860 2.8GHz and 4G memory. The CAV algorithms were implemented in these two systems for comparison. The dataset we used to verify the speedup performance and the increase of the robustness are two videos which were from the VIRAT Video Dataset [58]. The size of the airborne frame is 720×480. The VIRAT dataset is designed to be realistic, natural, and challenging for video surveillance domains in terms of its resolution, background clutter, diversity in scenes, and human activity/event categories than existing action recognition datasets. Other data sets have been applied for indoor settings [36].

Table 1 shows the processing time for the algorithm applied on different numbers of threads. In the serial version, the running time kept at 0.546 ms; and in the Xenserver version, as expected, the processing time reaches minimum 0.121ms at 10 threads which represents the local maximum in the Speed-up Performance chart as can be seen in Fig. 9.

Table 1. Multi-platform frame Processing Time (ms)

Thread Numbers	Xenserver	CPU
1	0.921	0.546
5	0.225	
10	0.121	
15	0.124	
20	0.131	

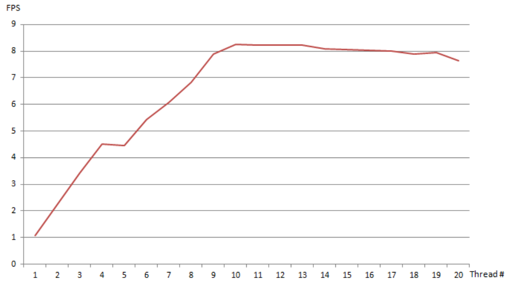


Fig. 9 Speed-up Performance based on the frames per second and the number of threads

It is worth noting that we use the most challenging airborne video sequence in the VIRAT Video Dataset for the proposed method as we have explored others [59]. For fair comparison of detection robustness, we tuned the parameters for best performance. The detector we chose to compare is a well-developed target detection method based on optical flow and principle component analysis [17]. The datasets are available at https://sites.google.com/site/dannyblueliu/.Since the image registration is included, our detector is implemented in C++ and runs at 9 frames per second (FPS) on Xenserver with 8 CPUs.

Table 2 gives the success rate of the objects of interest detection by the CAV system using the Principle Components Analysis (PCA) and Optical Flow method [17]. A joint PCA and Horn-Schunck optical flow approach for motion detection. Instead of considering the original optical flow, the two eigenvalues of the covariance matrix of local optical flows are analyzed. Since the first eigenvalue represents the major motion component and the second eigenvalue represents the minor motion component or turbulence, they are more useful to detect true motions with effectively suppressing false alarms. Table 2 records the quantitative results based on different challenging video sequences. The success rate is based on the ratio between the number of frames which the objects of interest get detected and the frame number of the video sequence. We note that the proposed Cloud based algorithm always achieves better results in all the sequences in terms of the success rate. This is because the AHM adaptively updates the histogram template of the objects of interest and thus achieves more robust performance (e.g. timeliness, accuracy) in detection.

Table 2. The Success Rate (%) of detection for proposed method and PCA & Optical Flow method

Sequence	CAV Cloud based Method	PCA & Optical Flow Method
VideoSequence1	85	64
VideoSequence2	74	61
VideoSequence3	82	69
VideoSequence4	69	62

5 Conclusion

In this paper, a Cloud-based Audio-Video (CAV) system for target detection and tracking was introduced. The CAV method is one of the first attempts at detecting and tracking multiple moving targets from airborne video using Cloud technology and audio information. The audio information from a user was converted to text as a feature for active-learning histogram matcher (AHM). It was shown that applying the Cloud computing infrastructure led to a much faster and real-time performance of detection and tracking as compared to situations when the whole workflow is applied in only one machine. Also, the obtained tracking results for the multiple moving targets tracking present in the airborne video datasets indicate that the CAV system demonstrates robust detection and tracking in real-time and under realistic conditions. Future work will include multi-mode tracking, image quality assessment [60], integration with text data for multimedia analysis [61], and multiple user applications viewing the same imagery for distributed applications [62].

Acknowledgement

The authors appreciate the reviewer comments to improve the paper. The research was supported in part by an AFOSR grant in Dynamic Data-Driven Applications Systems and an AFRL grant FA8750-13-C-0110.

References

[1] E. Blasch, L. Hong "Simultaneous Feature-based Identification and Track Fusion," *IEEE Conf. on Dec. Control,* pp. 239-245, 1998.

[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] T. Connare, *et al*., "Group IMM tracking utilizing Track and Identification Fusion," *Proc. Workshop on Estimation, Tracking, and Fusion; A Tribute to Y. Bar Shalom,* 2001.

[4] E. Blasch and B. Kahler, "Multiresolution EO/IR Tracking and Identification" *Int. Conf. on Info Fusion,* 2005**.**

[5] E. Blasch, C. Yang, I. Kadar, "Summary of Tracking and Identification Methods," *Proc. SPIE*, Vol. 9119, 2014.

[6] E. Blasch, E. Bosse, and D. Lambert, *High-Level Information Fusion Management and Systems Design*, Artech House, Norwood, MA, 2012.

[7] E. P. 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 Mag.,* Vol. 27, No. 9, Sept. 2012.

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

[9] O. Mendoza-Schrock, J. A. Patrick, *et al*., "Video Image Registration Evaluation for a Layered Sensing Environment," *Proc. IEEE Nat. Aerospace Electronics Conf (NAECON)*, 2009.

[10] Y. Wu, *et al*., "Multiple Source Data Fusion via Sparse Representation for Robust Visual Tracking," *Int. Conf. on Info Fusion*, 2011.

[11] Z. Liu, E. Blasch, Z. Xue, R. Langaniere, and W. Wu, "Objective Assessment of Multiresolution Image Fusion Algorithms for Context Enhancement in Night Vision: A Comparative Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(1):94-109, 2012..

[12] C Wang, S Li, Y Shen, Y Song, "Evaluation of Feature Detectors and Descriptors for Motion Detection from Aerial Videos," *ICPR*, 2014.

[13] P. Liang, G. Teodoro H. Ling, E. Blasch, G. Chen, L. Bai."Multiple Kernel Learning for Vehicle Detection in Wide Area Motion Imagery," *Int'l. Conf. on Info Fusion*, 2012.

[14] A. Mitiche and P. Bouthemy, "Computation and analysis of image motion: a synopsis of current problems and methods," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 29-55, 1996.

[15] W. Hu, T. Tan, L. Wang, S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Tr. on Systems, Man, and Cybernetics – Part C*, vol. 34, no. 3, pp. 334-352, Aug. 2004.

[16] K. Liu, H. Yang, B. Ma, and Q. Du, "A joint optical flow and principal component analysis approach for motion detection," *IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing*, pp. 1178-1181, Mar. 2010.

[17] K. Liu, Q. Du, H. Yang, B. Ma, "Optical Flow and Principal Component Analysis-Based Motion Detection in Outdoor Videos," *EURASIP Journal on Advances in Signal Processing*, 2010.

[18] K. Liu, H. Yang, B. Ma, and Q. Du, "Fast Motion Detection from Airborne Videos Using Graphics Computing Unit," *Journal of Applied Remote Sensing*, vol. 6, no.1, Jan. 2011.

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

[20] G. Farneback, "Two-Frame Motion Estimation Based on Polynomial Expansion," *Scandinavian Conf. on Image Analysis*, pp. 363-370, 2003.

[21] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *International Joint Conference on Artificial Intelligence*, pp. 674-679, Aug. 1981.

[22] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185-203, 1981.

[23] B. Horn and B. Schunck, “Determining optical flow: a retrospective”, *Artificial Intelligence*, vol. 59, no. 1-2, pp. 81-87, 1993.

[24] K. Lee, H. S. Ryu, J. H. Lee, K. J. Park, C-W. Shin, J. Woo, “Event-based image processing apparatus and method,” *US Patent 20130335595 A1*, Dec. 19, 2013.

[25] Z. Ji, W. Wang, “Robust Object Tracking via multi-task dynamic sparse model,” *Int’l Conf. on Image Processing*, 2014.

[26] R. V. Babu, P. Parate, K. A., Acharya, “Robust tracking with interest points: A sparse representation approach,” *Image and Vis. Comp.*, 2015.

[27] Y. Liu, H. Zhang. Z. Su, X. Lou, “Visual Tracking with Multi-Level Dictionary Learning,” *IEEE Int’l Conf. on Digital Home*, 2014.

[28] E. Blasch, ”NAECON08 Grand Challenge Entry Using the Belief Filter in Audio-Video Track and ID Fusion,” *Proc. IEEE Nat. Aerospace Electronics Conf (NAECON)*, 2009.

[29] C. Yang, T. Nguyen, *et al.*, “Field Testing and Evaluation of Mobile Positioning with Fused Mixed Signals of Opportunity,” *IEEE Aerospace and Electronic Systems Magazine,* Vol. 29, No. 4, April 2014.

[30] 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.

[31] R. I. Hammoud, C. S. Sahin, E. P. Blasch, B. J. Rhodes, “Multi-Source Multi-Modal Activity Recognition in Aerial Video Surveillance,” *Int. Computer Vision and Pattern Recognition* (ICVPR) Workshop, 2014.

[32] R. I. Hammoud, C. S. Sahin, *et al.*, “Automatic Association of Chats and Video Tracks for Activity Learning and Recognition in Aerial Video Surveillance,” *Sensors*, 14, 19843-19860, 2014.

[33] E. Blasch, J. Nagy, A. Aved, W. M. Pottenger, M. Schneider, R. Hammoud, E. K. Jones, A. Basharat, *et al.*, “Context aided Video-to-Text Information Fusion,” *Int’l.. Conf. on Information Fusion*, 2014.

[34] J. R. Smith, C-S. Li, ”Adaptive synthesis in progress retrieval of audio-visual data,” *IEEE Int’l Multimedia Expo*, 2000.

[35] R. Wu, Y. Chen, E. Blasch, B. Liu, *et al.*, “A Container-based Elastic Cloud Architecture for Real-Time Full-Motion Video (FMV) Target Tracking,” *IEEE App. Imagery Pattern Rec. Workshop*, 2014.

[36] B. Liu, E. Blasch, Y. Chen, A. J. Aved, A. Hadiks, D. Shen, G. Chen, “Information Fusion in a Cloud Computing Era: A Systems-Level Perspective,” *IEEE Aerospace and Electronic Systems Magazine*, Vol. 29, No. 10, pp. 16 – 24, Oct. 2014.

[37] E. Blasch, P. BiBona, M. Czajkowski, K. Barry, *et al.*, “Video Observations for Cloud Activity-Based Intelligence (VOCABI),” *IEEE National Aerospace and Electronics (NAECON)*, 2014.

[38] M. Menzel and R. Ranjan, “CloudGenius: decision support for web server cloud migration,” *ACM International conference on World Wide Web*, pp.979-988, New York, NY, Apr. 2012.

[39] “http://www.citrix.com/products/xenserver/overview.html”

[40] K. Liu, N. Kehtarnavaz, “Real-time robust vision-based hand gesture recognition using stereo images,” *Journal of Real-Time Image Processing*, pp.1-9, Feb. 2013.

[41] K. Liu, N. Kehtarnavaz and M. Carlsohn, “Comparison of two real-time hand gesture recognition systems involving stereo cameras, depth camera, and inertial sensor,” *Proc. SPIE,* Vol. 9139, 2014.

[42] 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,” *Future Information Technology, Lecture Notes in Electrical Engineering*, Vol. 276, pp 455-468, 2014.

[43] H. Ling, L. Bai, *et al.*, “Robust Infrared Vehicle Tracking Across Target Pose Change using L_1 regularization,” *Int’l Conf. on Info Fusion*, 2010.

[44] B. Liu, E. Blasch, Y. Chen, D. Shen and G. Chen, “Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier,” *IEEE International Conf. on Big Data*, 2013.

[45] B. Liu, Y. Chen, D. Shen, G. Chen, K. Pham, E. Blasch and B. Rubin, “An adaptive process-based cloud infrastructure for space situational awareness applications,” *Proc. SPIE,* , vol. 9085, 2013.

[46] 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.

[47] E. Blasch, “Enhanced Air Operations Using JView for an Air-Ground Fused Situation Awareness UDOP,” *AIAA/IEEE DASC*, 2013.

[48] B. Liu, Y. Chen, D. Shen, *et al.*, “Cloud-based space situational awareness: initial design and evaluation,” *Proc. SPIE*, vol.8739, 2013.

[49] D. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *Int’l J. of Computer Vision*, vol. 60, no.2, pp. 91-110, Nov. 2004.

[50] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, “SURF: Speeded Up Robust Features,” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, Jun. 2008.

[51] C. Harris and M. Stephens, “A combined corner and edge detector,” *Proceedings of the Alvey Vision Conference*, pp. 147–151, 1988.

[52] L. Wang, M. Liao, M. Gong, R. Yang and D. Nister, “High-quality real-time stereo using adaptive cost aggregation and dynamic programming,” *IEEE Int’l Symp. on 3D Data Processing, Vis, and Transmission*, 2006.

[53] P. Senin, *Dynamic Time Warping Algorithm Review*, Information and Computer Science Department, University of Hawaii at Manoa, 2008.

[54] K. Liu, C. Chen; R. Jafari, N. Kehtarnavaz, “Fusion of Inertial and Depth Sensor Data for Robust Hand Gesture Recognition,” *IEEE Sensors Journal*, vol. 14, no. 6, pp. 1898-1903, Jun. 2014.

[55] C. Bao, Y. Wu, H. Ling and H. Ji, “Real time robust L1 tracker using accelerated proximal gradient approach,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1830-1837,. 2012.

[56] K. Zhang, L. Zhang, and M. Yang, “Real-Time Compressive Tracking,” *European Conf. on Computer Vision*, vol. 7574, pp. 864-877, Oct. 2012.

[57] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Mar 25, 2004

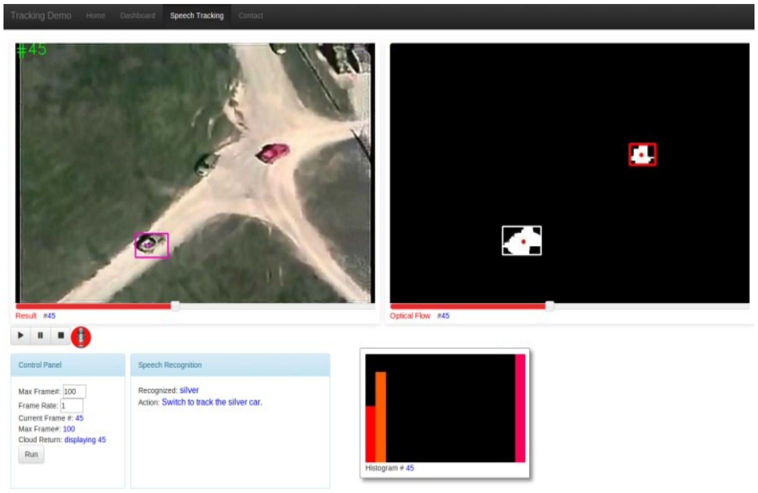
[58] http://www.viratdata.org/

[59] 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.

[60] E. Blasch, X. Li, *et al.*, “Image Quality Assessment for Performance Evaluation of Image fusion,” *Int’l Conf. on Information Fusion,* 2008.

[61] E. Blasch, S. K. Rogers, J. Culbertson, A. Rodriguez, *et al.*, “QuEST in Information Fusion,” *IEEE Nat. Aerospace and Elect. Conf.*, 2014.

[62] Y. Zheng, W. Dong, *et al.*, “Qualitative and quantitative comparisons of multispectral night vision colorization techniques,” *Optical Engineering*, Vol. 51, Issue 8, Aug. 2012.



Max Frame#: 100
Frame Rate: 1
Current Frame #: 45
Max Frame#: 100
Cloud Return: displaying 45

Recognized: silver
Action: Switch to track the silver car.

Figure 10. Voice Tracking Web GUI