

Towards an Integrated Aerial and Ground Surveillance System

Subhabrata Bhattacharya, Markus Quaritsch*, Bernhard Rinner*, Mubarak Shah

Computer Vision Lab,
University of Central Florida

{subh, shah}@cs.ucf.edu

*Institute of Networked and Embedded Systems,

Klagenfurt University, AUSTRIA

<firstname.lastname>@uni-klu.ac.at

Abstract—This paper introduces an integrated surveillance system capable of tracking multiple objects across aerial and ground cameras. To this end, we propose a set of methodologies that deal with tracking problems in urban scenarios where cameras mounted on quad-rotor unmanned helicopters could be used in conjunction with ground cameras to track multiple subjects persistently. We track moving objects from a moving aerial platform using a three staged conventional technique consisting of ego-motion-compensation, blob detection, and blob tracking. A hierarchical robust background subtraction followed by a motion correspondence algorithm is applied to track objects from the ground surveillance camera. Using metadata available at the airborne camera and the calibration parameters of the ground camera, we are able to transform the object’s position in both cameras’ local coordinate system to a generic world coordinate system. Trajectories obtained in terms of the generic world coordinate system are then merged assuming temporal continuity. A false candidate trajectory is eliminated using a similarity measure based on color intensity of the object that generated it. Our system has been tested in 3 real-world scenarios where it has been able to merge trajectories successfully in 80% of the cases.

I. INTRODUCTION

There has been an exponential increase in the awareness of surveillance in relation to safety and security in recent years. Video cameras and data storage, which form the basic necessity of these surveillance systems, are also affordable with low operational cost. However, cameras have finite spatial resolution, limiting their fields of view, thereby rendering them inefficient in real world surveillance scenarios, e.g. tracking an object. In order to solve this problem, Pan-tilt-zoom (PTZ) cameras were introduced whose fields of view could be adjusted remotely according to the changing regions of interest. While a network of several such PTZ cameras is a seemingly viable solution to this problem, the intricacies involved in control and automation of these cameras are prohibitive. This becomes the primary reason to explore the applicability of low-flying cameras in the context of such surveillance scenarios. Cameras mounted on aerial platform (more specifically, quad rotor unmanned helicopters) have wider coverage of a scene as the platform exhibits unconstrained 3-dimensional motion, as opposed to fixed cameras with no movement or PTZ cameras with limited panning movement. Moreover, with the decreasing cost of consumer electronics and significant technological

advances in unmanned aerial system design, building such a system is practically feasible. The interested reader is referred to [1] for more information on remote controlled helicopters equipped with video cameras that are widely used in commercial aerial surveillance systems.

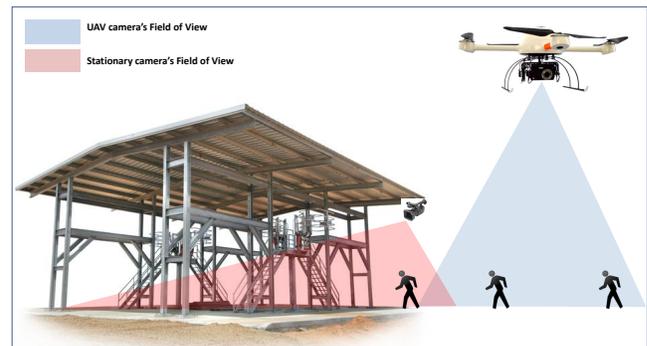


Fig. 1. A typical urban scenario under aerial and ground surveillance: Object is in the field of view of the aerial surveillance camera for a period of time and can be tracked. As it enters the building it escapes the field of view of the aerial camera, but is in the field of view of the ground camera and can be tracked again. Our objective is to establish correspondence between the two instances of the same person.

Consider a typical urban scene where a object is approaching a building from a distance. A camera mounted on a UAV is flying at an altitude just appropriate to distinguish between two or more objects based on color information. At a fixed altitude, the aerial camera has the same degrees of freedom as that of the object on the ground, hence the target is never lost from the field of view of the camera as long as the object is outside the building. However, as soon as the object enters the building, the aerial camera loses its track. Since the building has its own set of fixed surveillance cameras, any moving object that is in the field of view of these cameras, could be tracked as illustrated in figure 1. Although the tracks outside and inside the building are generated by the same object, it continues to be an extremely challenging problem in computer vision. Huttenlocher and Zabih in their technical report [2] have already discussed the gravity of the problem. The primary reason behind this is the incoherence between the fields of view from which the images of the same object is being generated. Any method

that seeks a direct correlation between the two images based on only appearance is bound to fail. We propose a novel technique to solve this correspondence problem using a 3-D geometry based approach. We also extend this method to generate persistent trajectories across cameras with mutually exclusive fields of view.

II. RELATED WORK

Object tracking is a well known problem in Computer Vision. The most fundamental case in this subdomain is tracking objects using a single stationary camera, which is predominantly done by background subtraction techniques followed by blob tracking. Several interesting extensions to the single camera object tracking problem have been proposed to date. A detailed literature review of tracking objects in multiple cameras is presented in the authors' book [4] on multi-camera surveillance.

Another non-trivial extension to the object tracking problem is observed in [3], [6], [7], [8], [9], and [10], where the authors have elucidated several methodologies involved in the detection, classification, and tracking of objects in aerial videos. In [5], the authors have addressed the problem of trajectory association across non-overlapping fields of view from multiple aerial cameras by making the motion model of each object with respect to time explicit. The association is refined by computing the maximum likelihood estimate of the inter-camera homographies, using Expectation Maximization algorithm. The closest in relevance to this paper is Sheikh and Shah's research [11] in tracking across multiple airborne cameras where the authors assume transitive closure between the fields of view of more than two cameras in order to ensure a coherent correspondence. They also restrict the cameras to the same plane. Our approach, on the other hand, exploits telemetry and calibration data from the cameras to map the fields of view to geospatial coordinates.

While all these prior efforts are interesting in their own aspects and many of them deal with different aspects of the multi-camera surveillance problem, yet they all assume a certain degree of homogeneity across the cameras. This is because in all multi-camera problems, the Brightness Transfer Functions (BTF) of all given cameras lie in the same low dimensional subspace as shown by the authors *et. al.* in [4], which could be effectively used to compute appearance similarity. Such a low-dimensional subspace is extremely difficult to determine for cameras whose fields of views are completely non-coplanar. To the best of our knowledge, our attempt to establish correspondence across completely uncorrelated and incoherent fields of view is an entirely novel study in itself. Furthermore, visual surveillance research, using both multiple airborne cameras and multiple ground cameras, has reached maturity in isolation. This work is the first step towards an integrated surveillance system which would open new avenues for research.

The rest of this paper is organized as follows. Section III provides a detailed description of our experimental hardware and software we have used to track moving objects in both



Fig. 2. A quad-rotor microdrone model md4-200

aerial and ground surveillance cameras. In section IV, we introduce our approach to establish correspondence between trajectories observed by our ground and aerial surveillance systems and generate persistent tracks. In section V, we conclude our discussion with some interesting results and provide some pointers toward future research in this area coupled with some real world applications.

III. INTEGRATED SURVEILLANCE SYSTEM

Object tracking in an aerial surveillance system is different from that of a ground surveillance system. We use separate hardware and software platforms for each task. Our aerial surveillance hardware consists of a lightweight Autonomous Unmanned Micro Aerial Vehicle (AUMAV) called the Microdrone model md4-200 (refer figure III) with Vertical Take Off and Landing (VTOL) capabilities. The drone is made of carbon fiber and reinforced plastic which inherently enhances lift and shields against electromagnetic interferences typically observed in urban environments. Flight stability is monitored in realtime using a built-in Altitude and Heading Reference System comprising of accelerometer, gyroscopes, and Magnetometer. The drone's quad-rotor lift system, which is based on synchronized, brushless direct drives, reduces the noise level and recovers from stall and overload conditions even at flight time. The payload capacity of the drone is approximately 200g, which is just sufficient to carry a small digital video camera. The drone is equipped with an onboard GPS system and video camera. The video captured by the camera is transmitted using a video channel at 24 frames/sec whereas the drone's positional coordinates and the camera *roll*, *pitch*, and *yaw* parameters, which form the telemetry information, are transmitted uniformly at 10 frames/sec from the drone using a separate channel. Both video and the meta-data information are processed using a 2.4 Ghz Intel computer serving as a base-station. The ground surveillance hardware consists of a video camera coupled with the same base-station to process output from this camera. The software environments used to process the video feeds from both of these cameras are discussed in sections III-A and III-B respectively.

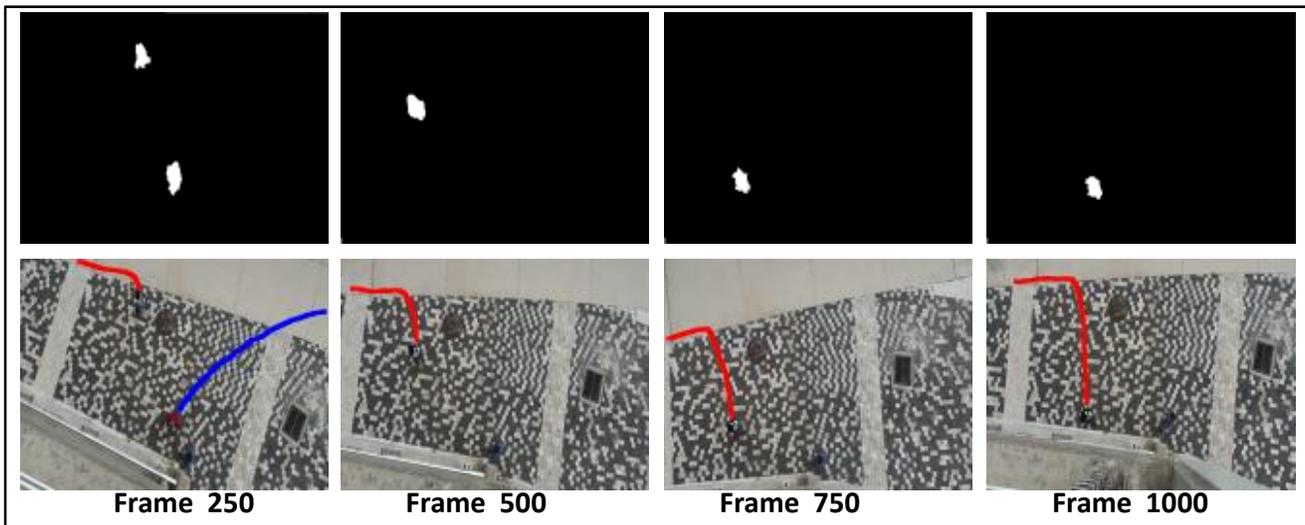


Fig. 3. Blob-detection followed by tracking in aerial surveillance system (each image is 250 frames apart): Top row shows blob detection results from UCF Harris sequence; Corresponding trajectories being output in the bottom row. It can be noted that the subject in red clothing is about to escape the field of view of the aerial camera (track shown in blue) and entering a shed in the first two images. Similarly, subject in white clothing approaching entry in the next frame and is about to escape the field of view of the aerial camera (track shown in red) and entering the shed in the last image.

A. Aerial Surveillance System

A fast OpenCV based implementation with some modifications to the original COCOA system [12] is used to retrieve tracks of moving objects from aerial video feed. The system processes videos at speed approximately equal to 12 frames/sec. A brief overview of the steps involved in the process is provided with some outputs.

In order to track objects effectively from an aerial camera, we initially have to eliminate ego-motion artifacts from the aerial sequence. In cases where the view is purely orthographic, the relationship between one image frame to the subsequent frame is affine and the transformation parameters (2×3 matrix) are computed using the direct registration technique proposed by Bergen *et. al.* in [13]. However, when the view changes from orthographic to oblique, a more refined projective estimation (Homography) is applied. This is a two step process in which interest points on a source image frame are computed using a method proposed by Shi and Tomasi in [14]. Since intra-frame motion is not significant, we compute corresponding candidate interest points in the subsequent target frame using pyramid optical flow as proposed in [15]. An iterative approach (RANSAC) is used to refine the selection of interest points that have strong correspondence across the source and target frames, which helps us to estimate the frame by frame homography matrix. This matrix contains the transformation parameters which are required to align one particular frame with respect to its previous frame.

Once a given number of consecutive frames are aligned with respect to the initial frame from the video input, we apply consecutive frame differencing on this given temporal window to obtain blobs. A battery of heuristics (mean gray area, blob compactness and eccentricity) are then applied

to filter most of the false blobs. In order to further refine the results of detection, area based thresholds are applied to reduce further false positives. The following voting scheme is applied to the i -th blob to filter misdetections from appearing in subsequent frames:

$$W_i = K \times mga_i + \frac{1}{L} \times C_i + \frac{1}{M} \times E_i, \quad (1)$$

where symbols denote the following:

- W_i is the weight assigned to each blob,
- mga_i is the mean gray area of the blob and is determined by taking the mean of pixels, in the temporal window of frames found within an N-connected neighborhood in i -th region,
- C_i compactness measure of i th blob,
- E_i eccentricity measure of the i th blob,
- K, L , and M are empirically determined constants.

Finally, a search across all blobs is done using the parameters observed in the detection stage, and similarities are derived from blob appearance, shape, and position of centers. If similarities between blobs of consequent frames are observed, tracks are generated accumulating similar blobs over the same temporal window of frames. In case the blobs disappear or reappear for short frame strides, tracks are obtained by linearly interpolating blob centers in the frames where the blobs had disappeared. A new track is generated for blobs that do not have any correspondence across previously detected blobs. Figure 3 elucidate the outcome of the blob-detection and corresponding blob-tracking processes respectively.

B. Ground Surveillance System

The motion detection algorithm used on the ground surveillance camera videos implements a robust background

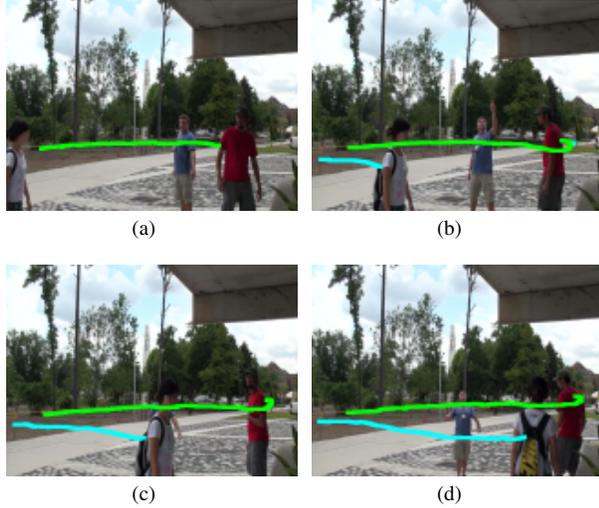


Fig. 4. Tracking in ground surveillance system (each image is 40 frames apart): (a) Subject in red clothing has entered the field of view of the ground camera (track shown in green) installed inside the shed mentioned in figure 3, (b), (c) and (d) Subject in white clothing moves within field of view of the ground camera (track shown in cyan).

subtraction proposed in [4]. This is a hierarchical method of carving moving foreground objects from the static background by first classifying each pixels into a foreground or background class using statistical distribution of gradient and intensity information. Foreground pixels, hence obtained from the intensity based subtraction, are grouped into blobs using connected component analysis. False regions are invalidated using gradient based subtraction, and this inference is used as a feedback to eliminate falsely classified pixels from the initial step. Finally, a frame level analysis is performed to remove further discrepancies that arise due to changes in illumination. Certain movements detected by the blob detection algorithm could be ignored using a threshold over detected blob-sizes. For example, the hand movement of a person as seen in figures 4(c) – 4(d) is ignored by the system as it is insignificant as compared to the full body motion.

After having the foreground blobs segmented, the next goal is to establish correspondence between these blobs over a sequence of frames. This is done by minimizing the deviation in speed and direction of motion exhibited by all detected blobs. The motion depicted by blobs provides a better idea of the motion exhibited by the whole object as compared to the individual points on the object which could be noisy. Figures 4(a) through 4(d) demonstrate the output of the ground surveillance system used in this paper with trajectories on the same scenario discussed in figure 3 .

IV. TRAJECTORY CORRESPONDENCE

Given a set of trajectories acquired by our UAV camera, our objective is to find the corresponding continuation using cues from the ground camera with the only assumption being that the temporal gap between each pair of trajectories is very small. For a formal definition of the problem, let us suppose that both the aerial and the ground cameras observe

two scenes with the same configuration as shown in section I for a fixed period of time. Let K_a and K_g be the number of objects observed by each cameras respectively. An object k , as observed by camera n (in this case $n = 1$ or 2), is denoted as O_k^n . Let us also denote the imaged location of each object under observation by $\mathbf{x}_k^n = (x_{k,t}^n, y_{k,t}^n)$. Therefore each trajectory is a set of points $\tau_k^n = (\mathbf{x}_{k,i}^n, \mathbf{x}_{k,i+1}^n, \dots, \mathbf{x}_{k,j}^n)$.

Since trajectories are recorded by cameras that do not share a co-planar relationship, the set of trajectories can not be associated unless a common transformation is applied to project the imaged locations O_k^n in the same plane. The next section gives a brief explanation of the technique we use to perform this transformation.

A. Transformation

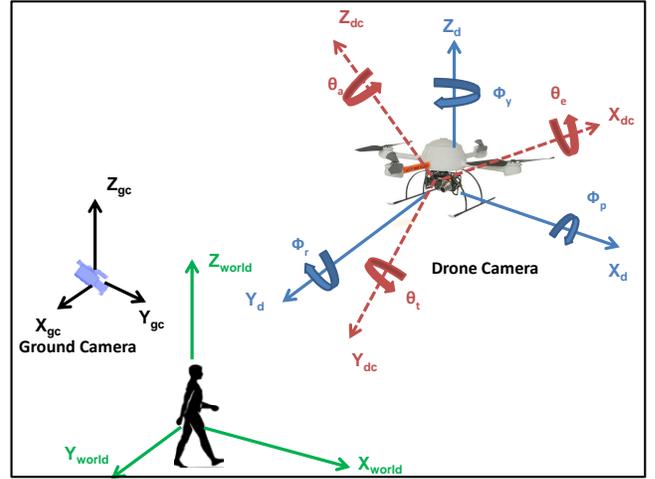


Fig. 5. The geometrical configuration of the aerial and ground cameras with respect to the world coordinates. Both the camera axes are represented in different colors (aerial:blue, ground:red). The different angles of rotation for both the drone and the camera attached to it are also shown. Since the drone's camera can only rotate about the Y-axis, only the twist angle needs to be taken into account; rest of the rotation parameters could be ignored.

The sensor telemetry or metadata is available to us from the microdrone. Recall that video feed is available at 24 frames/sec while the metadata is available for approximately every 3rd frame. Since telemetry does not change frequently within this period, interpolating the metadata parameters is a reasonable assumption. Similar to the drone metadata, we have calibration data for our ground camera. Both the metadata and the calibration data are used to transform the respective locations of the objects in image coordinates of the airborne camera and the ground camera respectively. Unlike the drone metadata, calibration data is constant for a given sequence. The following figure 5 illustrates the coordinate system of both the cameras and the relationship between the image coordinates of the drone camera and the world coordinates.

Ideally while building the sensor model, we need the following parameters from the metadata:

- Geodetic coordinates in Latitude and Longitude (translation along the y and x axes respectively T_y, T_x),

- Instantaneous altitude (translation along z-axis T_z),
- Camera orientation as *elevation*, *twist*, and *azimuth* (rotations around x, y, and z axes relative to the drone: $\theta_e, \theta_t, \theta_a$),
- Instantaneous angular displacement (pitch, roll, yaw) of the drone (rotations along x,y,z axes : ϕ_p, ϕ_t, ϕ_a), and
- focal length of the camera (f).

We assume that the drone is flying at a fixed altitude over the ground plane. Additionally, the microdrone camera has only one degree of freedom, elevation (θ_e) and azimuth (θ_a) could be ignored (refer to figure 5 for more detail).

Given telemetry ($T_x, T_y, T_z, \theta_e, \phi_p, \phi_t, \phi_a, f$), the geographical location (world 3D coordinates) of the object concerned ($\mathbf{x}_{k,t}^w = (x_{k,t}^w, y_{k,t}^w, z_{k,t}^w)$) at any given time t is related to the imaged location ($x_{k,t}^i$) as:

$$\mathbf{x}_{k,t}^n = \Pi_{sensor} \mathbf{x}_{k,t}^w, \quad (2)$$

where Π_{sensor} is the sensor model given by:

$$\Pi_{sensor} = T_x T_y T_z \theta_e \phi_p \phi_t \phi_a, \quad (3)$$

and,

$$\mathbf{x}_{k,t}^n = (x_{k,t}^n, y_{k,t}^n, -f). \quad (4)$$

From equation 2, the geo-spatial coordinates of the object could be retrieved using a simple ray-tracing function. For a non planar surface this ray-tracing function is called a terrain projection which exploits Digital Elevation Maps. As the surface under observation is very close to planar, we set $z_{k,t}^w = 0 \forall t \in R$.

Since translation along x and y axes are available in terms of latitude and longitude, we need to convert it into metric system, to be eventually used to match against the ground camera calibration data. These equations are used to perform the conversion:

$$X_{k,t}^w = (N(\phi) + h) \cos(\phi) \cos(\lambda), \quad (5)$$

$$Y_{k,t}^w = (N(\phi) + h) \cos(\phi) \sin(\lambda), \quad (6)$$

where,

$$N(\phi) = \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}}, \quad (7)$$

ϕ, λ , and h are latitude, longitude and height, a and e^2 are the semi-major axis and the square of the first numerical eccentricity of the ellipsoid respectively. The ground camera is calibrated using a method proposed by Drenk *et. al.* in [16], and the GPS coordinates of the ground camera is already known. We perform a simple experiment to verify the calibration of the ground camera and the telemetry of the aerial camera. Both these cameras are simultaneously deployed to cover a flat planner surface in such a manner that both their views coincide. Objects are captured walking by both the cameras, the input is processed by the aerial and ground surveillance systems, and tracks are generated. Objects in image coordinates from both the cameras are converted to the 3D world coordinate system and plotted

in figure 6(c). In figure 6(d), we plot the respective instantaneous speeds of individual objects. In almost all cases, we have observed that the positional coordinates from the cameras have aligned nearly perfectly. Also, object speeds as computed from both the cameras when plotted against each other demonstrate high correlation. The results from these two experiments reinforce our case to explore trajectory matching across these cameras in non-overlapping fields of view.

B. Matching

After we have obtained a set of trajectories from both the aerial and ground cameras, our next task is to match between the trajectories. The temporal information is used here as a discriminating cue. Let $\tau_k^w = (\mathbf{x}_{k,i}^w, \mathbf{x}_{k,i+1}^w, \dots, \mathbf{x}_{k,j}^w)$ be the trajectory defined by actual world locations of the of the object k from time $t = i$ to $t = j$ during which it was in the field of view of the aerial camera. Since we assume the temporal gap in the field of view switch between the aerial and ground cameras (also known as handover period) is very small, the object's exit from the aerial camera's field of view has a one to one mapping with his entry in the ground camera's field of view. We exploit this constraint to match trajectories from both the cameras and generate a complete trajectory of the form:

$$\tau_k^W = \{(\mathbf{x}_{k,i}^w, \mathbf{x}_{k,i+1}^w, \dots, \mathbf{x}_{k,j}^w), (\mathbf{x}_{k,j}^w, \mathbf{x}_{k,j+1}^w, \dots, \mathbf{x}_{k,l}^w)\},$$

where trajectory elements in $t = i$ to $t = j$ is generated from the aerial camera and the same in $t = j + 1$ to $t = l$ are generated from the ground camera.

C. Persistent Track Generation

In several real-world scenes, it is often observed that objects reappear in the fields of view of the respective aerial and ground cameras after short periods of disappearance. In those circumstances, it is highly desirable to re-associate the object back to its last appearance in the same field of view before it had disappeared. In order to accomplish this, we maintain a signature of the object in each trajectory observed by a particular camera. An object signature is derived using color histograms computed over a window of frames over the detected object in the input video.

In order to re-associate objects with their previous occurrences, object signature is obtained in the given field of view. After that, a nearest neighbor search over all previously observed signatures is performed and the new signature is re-associated with the trajectory that returned the topmost match in the search. If there is no match in the specified nearest neighbor radius, the signature along with its current trajectory is updated as an independent entry in the trajectory cache. For our experiments, we have computed color histogram on the detected object for 10 consecutive frames and performed exhaustive search for all possible signature matches. The primary limitation of this approach is in dealing with illumination changes and clothing similarity between objects.

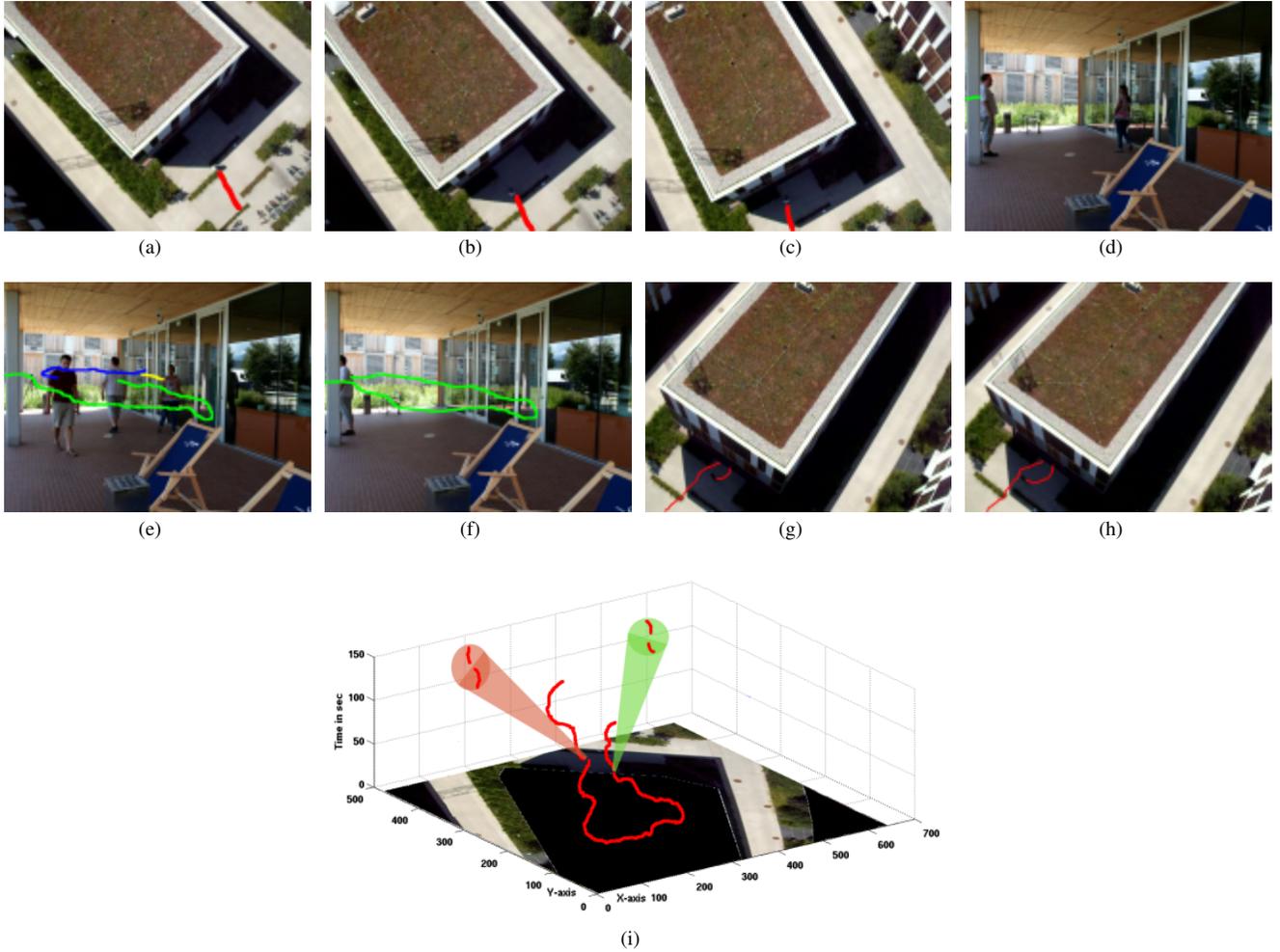


Fig. 7. Persistent trajectory generation (each image is 50 frames apart in the sequence): (a), (b), (c) Subject in white clothing gradually escapes the field of view of aerial camera (track shown in red), (d) The same object enters the field of view of the ground camera (e), (f) Subject moves within the field of view of ground camera (track shown in green), (g), (h) Track persistence constraint followed by signature matching applied to match trajectories between ground and aerial cameras and reassociate aerial camera tracks after 500 frames, dotted (not visible from the field of view of the aerial camera) tracks in red are generated from the tracks exported from the ground camera. (i) 3D visualization of the single persistent trajectory in sequence 1. This is generated by associating both aerial and ground trajectories after applying temporal continuity when the object escapes the field of view of one camera and enters that of the other. In this example there are two such instances (represented as zoomed insets). The x and y axes show the geospatial locations of the trajectories in the scene while the z-axis represents the time.

Sequence	GT	# of assoc.	GT	# of reassoc.
Seq 1	8	8	8	5
Seq 2	11	10	11	7
Seq 3	16	13	16	12

TABLE I

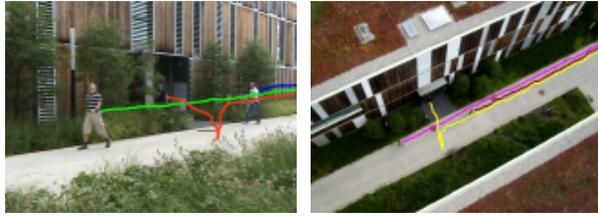
QUANTITATIVE ANALYSIS OF THE PERSISTENT TRAJECTORY GENERATION ALGORITHM (GT - GROUND TRUTH, REFER TO TEXT FOR DETAILS.)

Some results are shown from 2 independent sequences in figure 8(a) and 8(b). The table I shows a quantitative analysis of the number of objects for which the algorithm could correctly generate persistent trajectories. The table can be interpreted as follows : In column 2, the actual number of trajectory associations are specified (number of

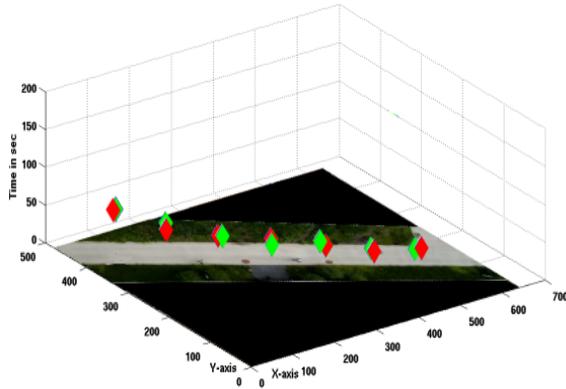
objects escaping the field of view of the aerial camera and entering the field of view of ground camera also known as ground truth). Column 3 contains the number of trajectories associated by the technique. Column 4 specifies the number of reassociations (actual number of objects then escaping the field of view of the ground camera and entering the field of view of the aerial camera). Finally, in column 5 the number of persistent trajectories generated is provided for each sequence. The dominant reason for the failure is due to shadow artifacts that degrade the discrimination capability of the color histogram based signature search.

V. CONCLUSION

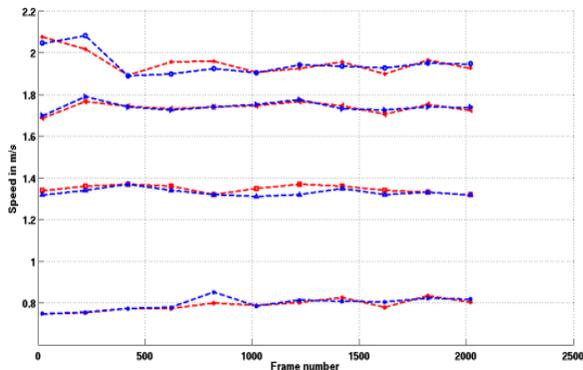
In this paper, we have proposed an approach to establish correspondence between moving trajectories across cameras with different fields of view separated in temporal domain.



(a) (b)



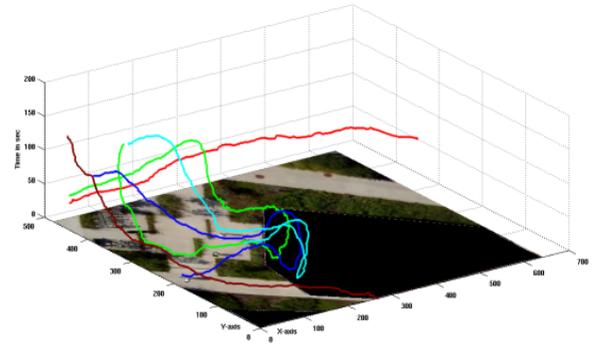
(c)



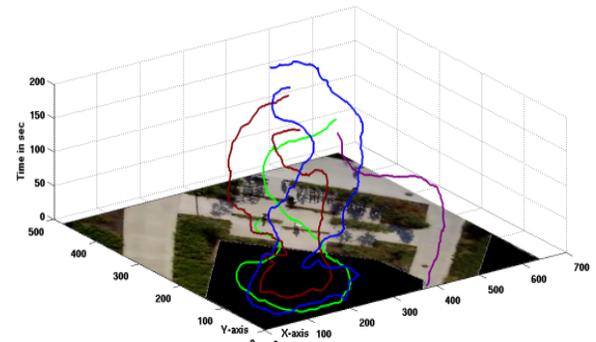
(d)

Fig. 6. Experiments depicting validity of metadata: (a) - (b) Trajectories of objects obtained from the ground and aerial surveillance systems respectively, after processing video recordings of the same scene. (c) Positions of a particular object as resolved into world coordinates by both the cameras from time $t = 0$ to $t = 10$ sec (z-axis). Positional coordinates (x,y axes) indicated by green diamonds are obtained by transforming the object's imaged location in the calibrated ground camera's coordinate system to the world coordinate system. Similarly, coordinates indicated by red diamonds are obtained after transforming the object's corresponding imaged location in the aerial camera's coordinate system. It could be observed how closely the points match up indicating the validity of the meta-data. (d) Graph showing the walking speeds (y-axis) of four independent objects strolling a cross the common field of view of both the cameras, at different segments of time. Each object's walking speed, calculated from individual surveillance systems, is represented by a pair of red and blue lines with different markers. The red curve corresponds to the speed observed by the ground surveillance system. The walking speeds also appear to match up quite well, as expected.

Our experiments demonstrate promising results with independent real world scenarios with an aerial and a ground camera surveillance system. There are some primary limitations of this approach which we intend to explore further.



(a)



(b)

Fig. 8. A subset of results from the persistent trajectory generation on sequence 2 and 3 is shown. The x,y axes corresponds to the world coordinate axes and the z-axis corresponds to time. Several trajectories are being shown.

Firstly, the technique proposed here is capable of associating trajectories correctly if the handover period is small. This is not a necessary criterion in real-world scenarios. Secondly, the color histogram based object signature encounters problems in cases that involve inter-object occlusion in addition to strong natural illumination changes (shadows, clouds etc.). Also, it is our general observation that the telemetry information is mostly reliable for shorter intervals of time (3-4 minutes). This restricts us to test the system on longer sequences. Therefore one possible interesting direction would be to explore the multiview correspondence problem with more semantic interpretation of the scene, e.g. person entering building, than purely low-level visual information. Another possible extension to this work would be to have more cameras in ground to cover a wider field of view inside a building (as observed in supermarkets with multiple exits). Programmable UAV could exploit this framework and enhance their automatic localization capabilities near the object of interest.

VI. ACKNOWLEDGEMENT

This research was funded in part by US Governments VACE program and by Lakeside Labs with funding from the European Regional Development Fund and the Corinthian Economic Promotion Fund (KW-20214-17095-24772).

REFERENCES

- [1] "ARC Helicopter.Com is Your Source for ARC Helicopter News & Information", <http://www.rchelicopter.com/>, Sep 2009.
- [2] D.H. Huttenlocher and R. Zabih, "Aerial and groundbased video surveillance at Cornell university", in Proc. DARPA Image Understanding Workshop, (Monterey, CA), 1998, pp. 77-83.
- [3] R. Kumar, H. Sawhney, S. Samarasekera, S. Hsu, Hai Tao, Yanlin Guo, K. Hanna, A. Pope, R. Wildes, D. Hirvonen, M. Hansen, P. Burt, "Aerial video surveillance and exploitation", Proceedings of the IEEE Volume: 89, Issue: 10, Oct 2001.
- [4] O. Javed, and M. Shah, "Automated Multi-camera Surveillance Algorithms and Practice", Springer Series: The International Series in Video Computing , Vol. 10, 2008.
- [5] Y. Sheikh, X. Li, and M. Shah, "Trajectory Association across Non-overlapping Moving Cameras in Planar Scenes", IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, USA 2007.
- [6] J. Kang, I. Cohen, G. Medioni, and C. Yuan, "Detection and tracking of moving objects from a moving platform in presence of strong parallax". In Proc. of IEEE International Conference on Computer Vision. (2005)
- [7] I. Cohen and G. Medioni, "Detecting and Tracking Moving Objects in Video from an Airborne Observer", In Proc. IEEE Image Understanding Workshop, pp. 217-222, 1998
- [8] J.Xiao, H. Cheng, H. Feng, and C. Yang "Object Tracking and Classification in Aerial Videos", Proc. of SPIE, the Intl. Soc. for Optical Engg., Automatic target recognition No18, Orlando FL, ETATS-UNIS (2008), vol. 6967, 2008, pp. 696711.1-696711.9
- [9] R. Pless, T. Brodsky, and Y. Aloimonos, "Detecting independent motion: The statistics of temporal continuity". IEEE PAMI, vol 22(8), 2000, pp. 768-773.
- [10] S. Zhang, "Object Tracking in Unmanned Aerial Vehicle (UAV) Videos Using a Combined Approach" In Proc. of ICASSP '05, vol.2, March 18-23, 2005, pp. 681 - 684
- [11] Y. Sheikh and M. Shah, "Object Tracking Across Multiple Independently Moving Cameras", In proc. of IEEE Intl. Conf. on Computer Vision, 2005.
- [12] S. Ali and M. Shah , "COCOA - Tracking in Aerial Imagery", SPIE Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications, Orlando, 2006.
- [13] J. R. Bergen, P. Anandan, Keith J. Hanna, and R. Hingorani, "Hierarchical Model-Based Motion Estimation", In Proc. of ECCV, 1992, pp.237-252.
- [14] J. Shi, and C. Tomasi, "Good features to track", In Proc.of CVPR, 1994, pp. 593-600.
- [15] J. Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm", http://robots.stanford.edu/cs223b04/algo_tracking.pdf. Sep 2009.
- [16] V. Drenk, F. Hildebrand, M. Kindler, and D. Kliche, "A 3D video technique for analysis of swimming in a flume", In Scientific Proceedings of the XVI/International Symposium on Biomechanics in Sports, 1999 pp. 361-364.