A CLUSTERING APPROACH FOR DETECTING MOVING OBJECTS CAPTURED BY A MOVING AERIAL CAMERA

Joseph DeGol

University of Illinois at Urbana-Champaign
Department of Computer Science
Urbana, IL 61801 USA

Myra Nam

MIT Lincoln Labratory
Intelligence and Decision Technologies
Lexington, MA 02420 USA

ABSTRACT

We propose a novel approach to motion detection in scenes captured from a camera onboard an aerial vehicle. In particular, we are interested in detecting small objects such as cars or people that move slowly and independently in the scene. Slow motion detection in an aerial video is challenging because it is difficult to differentiate object motion from camera motion. We adopt an unsupervised learning approach that requires a grouping step to define slow object motion. The grouping is done by building a graph of edges connecting dense feature keypoints. Then, we use camera motion constraints over a window of adjacent frames to compute a weight for each edge and automatically prune away dissimilar edges. This leaves us with groupings of similarly moving feature points in the space, which we cluster and differentiate as moving objects and background. With a focus on surveillance from a moving aerial platform, we test our algorithm on the challenging VIRAT aerial data set [1] and provide qualitative and quantitative results that demonstrate the effectiveness of our detection approach.

Index Terms— Aerial video, slow motion detection, clustering, and graph representation.

1. INTRODUCTION

We address the problem of motion detection from video captured by a moving camera. In particular, we focus on data captured from a camera on board an aerial vehicle flying over regions of interest. The captured scene may contain large camera motions from aircraft instability and may have a large number of objects moving at various rates throughout the scene. Our goal is to segment the moving objects in the scene by leveraging the differences between each object's motion and background motion induced by the moving camera. There are many applications of this research including video surveillance, activity analysis, and robot and drone navigation [2, 3, 4].

Motion detection is a fundamental computer vision problem. It has been well studied with early work often leveraging the assumption that the scene was captured with a stationary camera [5, 6, 7, 8, 9]. This assumption simplifies the problem because object motion becomes the only motion in the scene. For video captured from mobile platforms such as robots and mobile phones, however, the stationary camera assumption can become invalid. Relaxing the stationary camera assumption presents many challenges. In a scene captured by a moving camera, the motion no longer comes just from object motion, but also from camera motion and scene geometry. In addition, the effects of each motion type vary depending on the velocity of the camera and objects, and the distance of the camera from the scene.

Traditional approaches for motion detection from video captured by a moving camera include background subtraction [10, 11, 12]. In addition, geometric, shape, and camera constraints have also been shown to be useful for motion detection, leveraging strong parallax by a perspective angle from a mid-range aerial view [13, 14, 15, 3, 16]. The most related work to this paper is the work by [17, 18]. They leverage long term trajectories for motion segmentation. However, these approaches explore close range activities with large moving objects in the scene.

For scenes captured from an aerial vehicle, camera motion is the dominant factor due to the large distance between the camera and the scene being captured; this makes object motion detection challenging for several reasons. Because the camera is mounted to an aerial vehicle, it is subjected to vibration, instability, and sometimes-violent motion. Moreover, due to the large distance between the camera and scene, these vibration and instability effects are only magnified. Not only does this cause groups of successive blurred frames, but it also induces large motions on the scene [1]. Couple these large motions with the comparatively slow pace of human and vehicle movements, and the motion of objects becomes difficult to discern, which is addressed as a challenging problem in [19]. In addition, the large distance between the camera and scene makes the geometry negligible, severely limiting the effectiveness of epipolar constraint based methods [3, 2]. Figure 1 provides detailed depictions of the aforementioned challenges.

Given a scene captured from a camera on board a moving aerial vehicle, the proposed method is capable of detecting

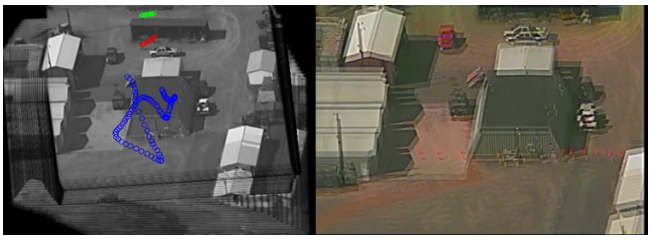


Fig. 1: A Depiction of the Challenges. Left: the dynamic camera path (blue) has high frequency jitter and undesirable motions. It makes it difficult to differentiate slow object motion (red and green) from dynamic camera motion. Right: the sporadic motion of the camera causes motion blur and double image effects.

minor object movement in the scene. For data captured from a moving aerial vehicle, we propose a novel framework that builds on the intuition that long-term trajectories of moving objects in a scene can be differentiated from long-term trajectories of stationary objects with induced camera motion. Having long windows of trajectories allows us to discriminate slowly moving objects from stationary scenery. The contributions of our paper are:

- Robustness against large camera motion in aerial video. Video stabilization is not required as a preprocessing step.
- No requirement of object detection. Without incorporating any prior knowledge on moving objects, we detect object-level motion in an unsupervised fashion.
- High tolerance for detecting slowly moving objects. In the aerial video domain, we leverage long-term range trajectories to detect motion in a robust way.

We propose a novel method to motion detection that leverages camera motion and keypoint trajectories over long windows of frames in order to cluster trajectories into individual object motions and a single background motion. Having long windows of trajectories allows us to discriminate slowly moving objects from stationary scenery; something that has proven difficult with other methods [19]. We define an attributed graph where each node is a motion trajectory and each weighted edge indicates the level of similarity between two given trajectories. Using automatic edge pruning, we disconnect dissimilar trajectories and identify clusters from the remaining connected components. Each cluster then corresponds to trajectories of the background or moving objects in the scene. We offer both qualitative and quantitative results for the challenging and relatively unexplored VIRAT aerial data set [1]. The proposed method provides an early attempt at overcoming some of the new challenges this data set offers. Next, we detail the method in Section 2. We continue in Section 3 with a discussion of the results and future work on the VIRAT aerial data set.

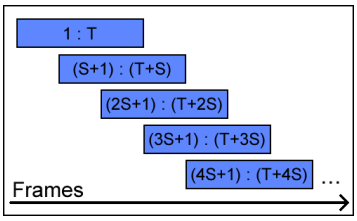


Fig. 2: Partitioning the sequence into windows. We partition the video frames by using a sliding window of size T with a shift of size S.

2. METHOD

This section details slow motion detection from video captured by an airborne camera. In order to detect motion in a given scene, we build a graph of keypoint trajectories and cluster them based on the difference between neighboring edge-connected keypoints' long term motion trajectories.

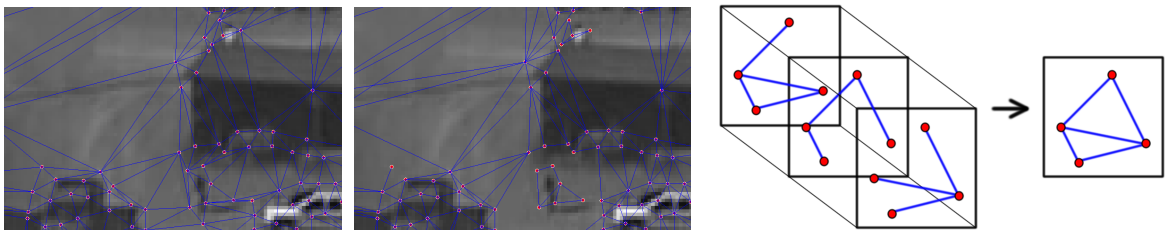
2.1. Building Graphs from Keypoint Trajectories

We partition a video sequence into time windows of size T with a shift of S frames to obtain short-term keypoint trajectories. Figure 2 depicts the time window partitioning. We begin by using the Kanade-Lucas Tomasi (KLT) Tracker [20, 21] to track the keypoint trajectories. Once keypoint trajectories are found, we prune away any trajectories that do not last the entirety of the time window. The set of stable keypoint trajectories is used to construct a graph $G(v, e)$ within each time window. Each vertex v of the graph represents a separate keypoint trajectory. The edges of the graph are then connected using a Delaunay Triangulation [22, 26]. Figure 3a depicts the resulting graph.

The graph node represents the keypoint trajectory over the time window. We compute the graph node attributes to represent how the keypoint trajectory fits its respective estimated camera motion trajectory. We leverage the camera motion to differentiate moving objects from the stationary background. The camera motion is estimated by RANSAC [23] with the KLT keypoints. The estimated homography matrix is an estimate of how the camera moved between frames. This means that the stationary keypoints in one frame, when transformed by the estimated homography matrix, should match their corresponding keypoints in the next frame. On the other hand, the non-stationary keypoints will not match because their motion was influenced both by the camera motion and their own independent motion.

Using the first frame as our reference, we compute the homography matrices over the time window. We construct a $1 \times T$ feature vector V_d that characterizes the discrepancy between the keypoint motion and the camera motion induced at the same point, given by

$$V_d(t) = \sqrt{(v(t) - H_t(v(t)))^2} \quad (1)$$



(a) (b) (c)

Fig. 3: From Feature Points to Clusters: (a) We construct a graph $G(v, e)$ where the vertices v shown in red are KLT keypoints and the edges e shown in blue are defined by the Delaunay Triangulation. (b) We prune edges e for each graph $G(v, e)$ by calculating a threshold for each vertex and pruning edges with weight above this threshold. (c) Overlapping windows will cause more than one graph for a given frame. We merge these graphs by keeping any edge that exists in at least half of the graphs.

where $v(t)$ is the keypoint location and H_t is the estimated homography function at window t ($t = 1, 2, ..., T$). The edge weight is computed based on dissimilarity between the motion trajectories of the two connected verticies i and j. The edge weights are given by

$$\omega(V_d^i, V_d^j) = \sum_{t=1}^{T} \sum_{t'=1}^{t} |V_d^i(T - t') - V_d^j(T - t')|. \quad (2)$$

where V_d^i and V_d^j represent the feature vector V_d from Equation 1 for two vertices i and j.

This metric is similar to the Match distance [24], a special case of the Earth Movers distance that perceptually measures the differences by comparing cross-bins in histogram comparison. Instead, we use the cumulative vector of the absolute differences between two trajectories to deal with negative vector elements. The accumulation is in a reverse order in order to maximize the motion differences.

2.2. Automatically Pruning Edges

Once edges have been assigned, we prune edges between dissimilarly moving objects; in this case, moving objects and stationary objects. Rather than using a hard threshold, we formulate a simple method based on [25] for automatically finding a threshold value for a local set of edges around a keypoint. First, we define a Noise Index $NI(v)$ for each keypoint v where $NI(v) =$ *LocalMean(v) / GlobalMean*. Here, *LocalMean(v)* is calculated by summing of the edge weights of all connected neighbors of keypoint v and then dividing by the number of connected neighbors of keypoint v. Similarly, *GlobalMean* is calculatd simply taking the mean of all edge weights. Next, we define a tolerance $T(v) =$ *StandardDeviation(v) / $NI(v)$* where *StandardDeviation(v)* is calculated by taking the standard deviation of the edge weights of all connected neighboring keypoints to v. Finally, we define the cut-off threshold $F(v) =$ *GlobalMean* $+ T(v)$.

Then, we prune edges of keypoint v if the edge weight is greater than the corresponding cut-off threshold $F(v)$. Repeating this process for each keypoint results in a pruned graph. An example pruned graph is shown in 3b.

2.3. Merge Graphs and Differentiate Moving Objects

Depending on the choice of window size T and the time shift S, it is possible to have several different graphs representing a given frame. Thus, it is necessary to merge these graphs into one representation of the frame. To merge several graphs, we employ a simple voting strategy. For a given edge, it must exist in at least half of the graphs representing the frame for it to exist in the final merged graph. For example, if we have 3 graphs for a frame, an edge must exist in 2 of the graphs for that edge to be merged into the final graph; a depiction of the merging process that matches the given example is shown in Figure 3c. This strategy works well because it does an additional round of pruning to fully separate the edges of moving objects from the background.

Once the graphs for a frame are merged into a final graph, we differentiate the clusters representing the moving objects from the stationary background cluster. To do this, we simply remove any clusters above a certain number of keypoints. This leaves us with only clusters that our algorithm has detected as moving objects.

3. RESULTS AND DISCUSSION

We demonstrate the effectiveness of our algorithm by presenting both quantitative and qualitative results using the VIRAT aerial data set. To show the improvements made by our system, we make a comparison with a baseline system that is created by setting shift $S = 1$ which removes overlapping frames and nullifies the merging and voting portions of our algorithm. For the full method, the parameters we used were: window size $T = 25$; shift $S = 5$; Max Cluster Size $= 20$.

We tested our algorithm on three different scenes. These scenes differ in terms of object motion speed, object size, and camera distance. Figure 4 provides the qualitative results for our three scenes and Table 1 provides the corresponding quantitative results, which were calculated using precision and recall rates. In our case, the true positives are calculated by the number of objects that are correctly detected, i.e. one cluster for one object. When an object is not detected, this is a false negative, and when an object is detected where it shouldnt be, this is a false positive. To decide if an object was correctly

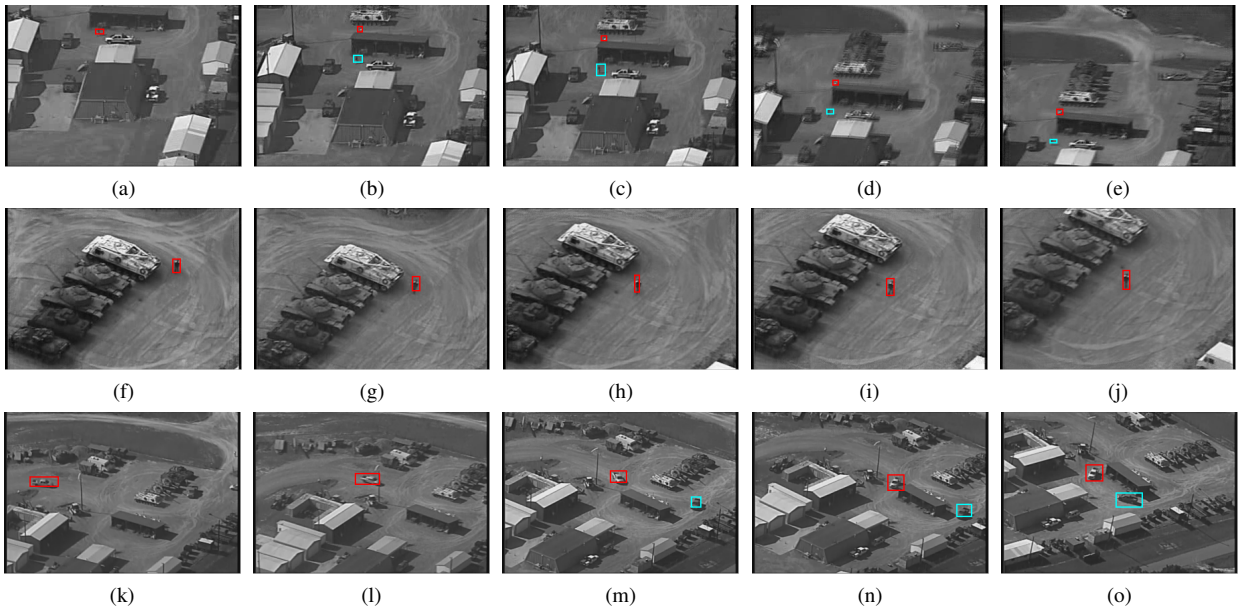


Fig. 4: Tracking results on Scene A, B, and C: (a-e) are frames of Scene A. The red and cyan boxes show the algorithm detecting the moving people; (f-j) are frames of Scene B. The red box represents the algorithm detecting the moving person; (k-o) are frames of Scene C. The red and cyan boxes show the algorithm detect the moving vehicles.

detected, we use one standard measure: the intersection of union areas between the detected bounding box and the annotated bounding box must be above 50% to be a true positive.

The incorporation of overlapping windows with merged frames improves recall for all three scenes. This can be seen in Table 1 where we achieve superior recall for all three scenes with values of 59.7%, 94.2%, and 98.7% respectively. Note that although the baseline method achieves higher precision than the full method for scene B, for all scenes the full method is still superior because the miss rate (i.e low recall) far outweights the slight advantage in precision. We reason that the baseline method is unable to find significant differences between the moving and stationary objects because of limited stable trajectories for short windows of time. By leveraging overlapping windows and merging, the full method has the added benefits of four additional graphs that span an extra 20 frames (assuming $T = 25$ and $S = 5$); increasing the number of stable trajectories and the span of frames for which discrepancies between camera and object motion can accumulate. These benefits are particularly noticable when comparing baseline recall between scene C and A because the moving objects in C move faster through the scene where the discrepancies accumulate faster. For scene A, however, the moving objects are moving slowly and the baseline doesn't have enough frames to accumulates large discrepancies for clustering moving objects. By incorporating the overlapping windows with merging, the full method is able to leverage additional frames and trajectories, making discrepancies accumulate, and improving the clustering. Thus, we see that our windowing approach improves motion detection; particularly for slow moving objects. In addition, we achieve good results despite sporadic camera motion inherent to the data and without the need for object detection. Lastly, there are two main failure cases (Figure 5) which occur when two objects are close together and moving at similar rates, or when one large object gets split into several clusters. We will address these fail cases in future work by incorporating additional measures into the edge weight. One possibility would be to incorporate pixel intensity histograms for the triangular regions between edges as demonstrated in [26].

Fig. 5: Failure Cases. Left: Two objects are close and moving at similar rates, so they are clustered together. Right: One large object gets split into several clusters.

Prec/Rec(%)	Frames	Baseline	Full Method
Scene A	100	68.1 / 29.6	88.9 / 59.7
Scene B	350	100 / 48.7	98.5 / 94.2
Scene C	300	51.6 / 54.2	64.4 / 98.7

Table 1: This table shows the computed precision and recall precentages for the number of frames for each scene.

Acknowledgement: This work is sponsored by the Department of the Air Force under Air Force Contract #FA8721−05−C−0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

References

[1] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, C.-C. Chen, Jong Taek Lee, Saurajit Mukherjee, J. K. Aggarwal, Hyungtae Lee, Larry Davis, Eran Swears, Xioyang Wang, Qiang Ji, Kishore Reddy, Mubarak Shah, Carl Vondrick, Hamed Pirsiavash, Deva Ramanan, Jenny Yuen, Antonio Torralba, Bi Song, Anesco Fong, Amit Roy-Chowdhury, and Mita Desai, "A large-scale benchmark dataset for event recognition in surveillance video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[2] Harpreet S. Sawhney, Yanlin Guo, and Rakesh Kumar, "Independent motion detection in 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1191–1199, Oct. 2000.

[3] Chang Yuan, G. Medioni, Jinman Kang, and I. Cohen, "Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1627–1641, 2007.

[4] Jong Taek Lee, Chia-Chih Chen, and J.K. Aggarwal, "Recognizing human-vehicle interactions from aerial video without training," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2011, pp. 53–60.

[5] Ismail Haritaoglu, Davis Harwood, and Larry S. David, "W4: Real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[6] Chris Stauffer and W. Eric L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[7] A. Elgammal, R. Duraiswami, D. Harwood, and L.S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151–1163, 2002.

[8] Omar Javed, Khurram Shafique, and Mubarak Shah, "A hierarchical approach to robust background subtraction using color and gradient information," in *Workshop on Motion and Video Computing*. 2002, MOTION '02, pp. 1–6, IEEE Computer Society.

[9] Y. Sheikh and M. Shah, "Bayesian object detection in dynamic scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, vol. 1, pp. 74–79 vol. 1.

[10] Rita Cucchiara, Andrea Prati, and Roberto Vezzani, "Real-time motion segmentation from moving cameras. real-time imaging," *Real-Time Imaging*, vol. 10, pp. 127–143, 2004.

[11] Dong Zhang and Ping Li, "Motion detection for rapidly moving cameras in fully 3d scenes," in *Pacific-Rim Symposium on Image and Video Technology*, 2010, pp. 444–449.

[12] Ali Elqursh and Ahmed Elgammal, "Online moving camera background subtraction," in *European Conference on Computer Vision*, Berlin, Heidelberg, 2012, pp. 228–241, Springer-Verlag.

[13] William B. Thompson and Ting-Chuen Pong, "Detecting moving objects," *International Journal of Computer Vision*, vol. 4, no. 1, pp. 39–57, 1990.

[14] R.C. Nelson, "Qualitative detection of motion by a moving observer," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1991, pp. 173–178.

[15] Jinman Kang, I. Cohen, G. Medioni, and Chang Yuan, "Detection and tracking of moving objects from a moving platform in presence of strong parallax," in *IEEE International Conference on Computer Vision*, 2005, vol. 1, pp. 10–17.

[16] Andreas Wedel, Annemarie Meißner, Clemens Rabe, Uwe Franke, and Daniel Cremers, "Detection and segmentation of independently moving objects from dense scene flow," in *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2009, pp. 14–27.

[17] Thomas Brox and Jitendra Malik, "Object segmentation by long term analysis of point trajectories," in *European Conference on Computer Vision*, 2010.

[18] P. Ochs and T. Brox, "Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions," in *IEEE International Conference on Computer Vision*, 2011.

[19] G. Georgiadis, A. Ayvaci, and S. Soatto, "Actionable saliency detection: Independent motion detection without independent motion estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 646–653.

[20] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[21] Carlo Tomasi and Takeo Kanade, "Detection and tracking of point features," Tech. Rep. CMU-CS-91-105, Carnegie Mellon University, Pittsburgh, PA, 1991.

[22] L. Paul Chew, "Constrained delaunay triangulations," *Algorithmica*, vol. 4, pp. 97–108, 1989.

[23] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[24] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, Nov. 2000.

[25] Vladimir Estivill-castro and Ickjai Lee, "Amoeba: Hierarchical clustering based on spatial proximity using delaunay diagram," in *International Symposium on Spatial Data Handling*, 2000, pp. 7–26.

[26] D. Sugimura, K.M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, "Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait," in *IEEE International Conference on Computer Vision*, 2009, pp. 1467–1474.