Resilient Mobile Cognition: Algorithms, Innovations, and Architectures

R. Viguier, C-C. Lin, K. Swaminathan, A. Vega, A. Buyuktosunoglu, S. Pankanti, P. Bose
IBM T J Watson Research Center, Yorktown Heights, NY 10598

H. Akbarpour, F. Bunyak, K. Palaniappan
Department of Computer Science, U. Missouri Columbia, MO 65211

G. Seetharaman
Advanced Computing Concepts
US Naval Research Laboratory 4555 Overlook Avenue SW, Washington DC, 20375

Abstract—The importance of the internet-of-things (IoT) is now an established reality. With that backdrop, the phenomenal emergence of cameras/sensors mounted on unmanned aerial, ground and marine vehicles (UAVs, UGVs, UMVs) and body worn cameras is a notable new development. The swarms of cameras and real-time computing thereof are at the heart of new technologies like connected cars, drone-based city-wide surveillance and precision agriculture, etc. Smart computer vision algorithms (with or without dynamic learning) that enable object recognition and tracking, supported by baseline video content summarization or 2D/3D image reconstruction of the scanned environment are at the heart of such new applications. In this article, we summarize our recent innovations in this space. We focus primarily on algorithms and architectural design considerations for video summarization systems.

I. INTRODUCTION

The next generation distributed infrastructure is expected to comprise of billions of interconnected mobile devices. It is easy to imagine that these infrastructures need to be inherently cognitive: be intelligent and be able to naturally interact with their users. If cognitive abilities need to be injected in the fabric of the next generation infrastructure, the prevailing centralized approach to cognition is unlikely to meet its requirements since typical mobile computing environments present severe challenges in terms of flexibility, agility, resilience, cost, and power. Taking the concrete context of wide area moving platform imagery summarization as a mobile cognitive task, we describe algorithmic subsystems that comprise the overall solution and present illustrative results on UAV video.

II. VIDEO SUMMARIZATION

A. Overview

The UAV multimedia content can be summarized in terms of overall spatial coverage area of the cameras (coverage summarization) or it could be summarized in terms of the events within the content (event summarization). The coverage summarization relies on spatially relating different video frames from the cameras, projecting them into common view space, and stitching them together to build a single panorama. The event summarization begins by first estimating the motion of the camera by registering its successive video frames. The overall summarization output generically consists of visualizing detected events within the videos on the panoramic summary image. Overall system diagram is shown in Figure 1.

Fig. 1. Block diagram of our video summarization approach using both image-based visual and sensor-based geospatial context information.

B. Previous Work

Many researchers have worked in summarizing videos from static cameras (e.g. [1], [2]) and for broadcast/production videos (e.g. [3]), a few researchers have focused their work on UAV video summarization (e.g. [4]). UAV video summarization needs the development of new analytics approaches that can fully exploit all of the information available including onboard geospatial location and pose sensors. However, these metadata are often noisy and research is needed to develop robust techniques for accurately integrating the additional metadata in a resilient fashion. Secondly, given the resource constraints (e.g., onboard computing, energy sources, space, weight, communication bandwidth), it is challenging to design a solution delivering real-time summarization capabilities. We would like to leverage the foundation laid by recent award winning flux tensor with split gaussian (FTSG) spatio-temporal analytic framework [5] and the previous related parallelization research, along with information fusion (e.g., [6]–[8]) towards developing an optimized summarization appliance.

III. EVENT SUMMARIZATION

The event summarization tasks consist of motion estimation, object detection, tracking, and event detection subsystems.

A. Motion Estimation

The flux tensor is the temporal variations of the optical flow field within the local 3D spatiotemporal volume. As shown
elsewhere (e.g., [6]), the moving and non-moving regions in video of a scene can be robustly determined using flux tensor information within the video.

### B. Object detection

From the flux tensor trace, detecting moving objects is straight-forward. Thresholding the flux response and running a connected component algorithm before filtering according to size or appearance will reliably segment moving structures from a stabilized background (See Fig. 2). This approach can be further refined in order to tackle some of the issues with motion detection: gaps inside slow moving homogeneous objects and inaccurate object boundaries due to the fact that the tensor response is high over the union of the moving object locations in the temporal window [6].

![Fig. 2](image)

**Fig. 2.** Side by side, one image taken from a VIVID sequence, with six moving vehicles and its corresponding foreground detection, using flux tensor.

### C. Tracking

After the objects have been correctly detected, a data association problem still remains. This can be formulated as a minimization problem. Each possible assignment between a track and a detected object has a cost calculated from positional or appearance matching distance. There is also a cost to terminate a track (i.e. no object assigned to it in the new image), and a cost to start one (i.e. new detected object not present before). Finding the global minimum is intractable, in general and a simplified (i.e., closest distance association tracker) approach is used for preliminary results. We select the option with the minimal cost for each object, between association to a previous track and the start of a new one, and in a second step we try to assign the tracks which were not updated with the objects that were not associated to a track during the first step. An example of tracking vehicles and pedestrians is shown in Fig. 3.

![Fig. 3](image)

**Fig. 3.** Example of tracking result for unstabilized moving camera. The tracks were filtered according to minimum length, and size of the target. Unlike the pedestrian, the car is not moving but is detected because of the camera motion.

### D. Event Detection

There are a variety of events that are of interest to the end users. Typically, these events can be broadly categorized into three types: (i) change-based: any structural changes in the scenes over time (e.g., new trenches in an area); (ii) objects and (inter)actions-based: specific type of object (e.g., helicopter), action (e.g., vehicles exceeding specific speed), or interaction (e.g., transfer of goods from one vehicle to another), and (iii) anomaly detection: characterizing typical populations of objects and their (inter)actions and detecting outliers thereof (e.g., cars traveling in opposite direction in a lane). The primary basis of event detection are spatio-temporal features, object detections and their tracks. In particular, Oh et al [9] focused on human action recognition in VIRA T videos. Additional pattern recognition techniques are typically needed to define specific event of interest. For illustrative purposes, we will assume that tracks exceeding certain duration are of interest to the end-user.

### IV. Coverage Summarization

In this section, we present details of the algorithms used to generate panoramas to obtain an area summarization covered by videos from UAV cameras.

#### A. Meta-Data Exploitation

In many UAV videos, metadata information is embedded in the video and can be extracted. However, it has some issues in accuracy, e.g. non-synchronized frame number, GPS offset, etc. Depending on different types of UAV, the inaccuracy issues might be different. For the DARPA VIRA T video dataset, the metadata stream may not always be synchronized with the video stream which results in large outliers for geolocalization. This can be compensated during post-processing of the metadata stream and the adjusted metadata can be used to improve the summarization results. In order to compensate for GPS inaccuracy and metadata alignment we use the the four corners of the FOV given in latitude and longitude coordinates to model the quadrilateral shape from each frame mapped to the ground onto which the image is warped.

#### B. Transformation Estimation

As mentioned in [4], between two consecutive frames, we use SIFT (scale-invariant feature transform) to detect feature points and Nearest-Neighbor Matching to find the putative corresponding points. It is followed by applying RANSAC (random sample consensus) to estimate the homography transformation between the frames while rejecting outliers among their putative matches. Absolute average difference in the overlapping area and the number of inliers are used to evaluate
accuracy of the estimated transformation. The estimated transformation is then further verified. If it is abnormal or cannot be found, the transformation between two consecutive frames is set to empty.

C. Mini-Panorama Generation

After estimating all the transformation parameters, we then determine the first and the last frames of each segment by checking the consecutive image transformations. To create a mini-panorama for each segment, we align every frame to the first baseframe within the segment. The global transformation for each frame is calculated using the previous estimated homographies $H$ between each pair of frames. The global transformation from the $k$th frame to the first frame of the segment is the cascade of the inverse adjacent transformations between the $k$th and first frames. To avoid a blurred panorama caused by parallax, when each frame is transformed to the panorama coordinate system, we only update the uncovered area. Because each video may contain several segments with dissimilar viewing angles and settings, each segment is represented by a mini-panorama.

Fig. 4(a) shows a top-down mini-panorama view created by warping quadrilateral shapes using frame metadata. In the warped images, the features on the ground have less distortion. When the warped images are used to detect feature points, their positions are more accurate. Therefore, the stitching results have less errors caused by the parallax effect. Additional accurate mini-panorama is shown in the Fig. 4 (b). The stitching result using metadata has fewer discontinuities around the road regions and the building with red roof.

D. Mini-Panorama Registration

We also use metadata to help register the mini-panorama. Assuming that we want to register two mini-panoramas generated from two different segments, the metadata are used to find the closest individual frames between those two segments in terms of geological locations. However, like Fig. 5, the scale difference is too big to find the corresponding points between those two frames. We create small panoramas using the neighboring frames around the individual frame and also update the relative size of small panoramas using metadata. After this, we can find the corresponding points, shown in Fig. 6 (a). Fig. 6 (b) shows that the two panoramas are registered correctly and automatically based on the corresponding points.

V. ARCHITECTURAL CONSIDERATIONS

Conventional design assumes that the analytic workload is pre-determined and proceeds to answer the following question: what system support architecture will help achieve the requirements implicit in the specification of the analytic workload? Under resilient operating conditions, it is instead desirable to assume that it is possible to deliver a continuum of useful workloads with associated spectrum of resource constellation. Taking UAV video summarization as an illustrative real-time analytic workload for resource constrained environment, we ask following questions: (i) how in practice we will be able to match the most useful analytic for given (dynamic) video

---

Fig. 4. (a) Stitched result using original VIRAT [9] frames; (b) stitched result using warped VIRAT frames.

Fig. 5. Two VIRAT mini-panoramas for registration.

Fig. 6. (a) Corresponding points between the small VIRAT panoramas with updated relative sizes; (b) registration result of mini-panoramas in VIRAT dataset.

Fig. 7. Video summarization result on VIVID [10] dataset.
data scenario. (ii) what architecture will help flexibly adapt (e.g., setting the spatio-temporal scale of the processing to match the content the video) analytics (e.g., visual saliency) to the given resources? Finally, there are significant opportunities of exploiting pixel, block, frame, and video level parallelism by building temporal dependency graphs under Star b-S model [8] that has already been demonstrated cell-processor [7] architecture and in multiple scenarios (e.g., analytics [6]). Below, we provide a qualitative (e.g., upper-bounds) summary of computational gains that can be achieved by the system by adopting appropriate architectural design.

Let us assume that a single camera \( r \times r \times 3 \) pixel-resolution video at \( f \) \( fps \) is available in raw format (i.e., no decoding overheads) in the main memory of a single CPU that is assumed to compute at \( c \) operations per second (\( ops/s \)). The CPU, multi-core or GPU processors are (for simplicity) assumed to have no memory capacity limitations and there are no parallelization/synchronization overheads. If we assume smooth trajectory of UAV field of view and the number of features and objects in the video significantly smaller than the number of pixels, the motion estimation subsystem is expected to dictate the overall computational complexity of the system. In practice, Palaniappan et al. [7] have improved the throughput of motion estimation subsystem in a multi-core implementation by a factor of 12–40 and accomplished power efficiency improvement by a factor of 50–160; the qualitative analysis below relates the potential improvements under different architectural considerations.

Assume that four parameters for the flux operator are \((n_{D_3}, n_{D_1}, n_{A_3}, n_{A_1})\) 1D filter sizes of spatial derivative, temporal derivative, spatial smoothing filter, and temporal smoothing filter, respectively [7] are identical and equal to \( k \). Baseline unoptimized computational load of the flux operator per image will be proportional to \( r^2 k^3 \) (cf. [7], Fig. 1). If we ignore for the moment boundary issues related to smoothing and derivatives computation, an FFT-based optimization of the baseline algorithm can be implemented by decomposing the 3D convolutions into three 1-D convolutions and is expected to improve the computational load by \( \frac{k^3}{m \log k} \). A similar pipelined optimization involving boxed-smoothing operator is expected to improve the baseline algorithm performance by a factor of \( k^3 \). A multi-core memory shared implementation will improve the baseline performance by a factor of \( ck^3 \) where \( c \) is the number of cores (normalizing for any differential in CPU \( ops/s\) rating). In a GPU implementation, the factor is expected to be \( \frac{ck^3}{g} \), where \( g \) is the number of GPU cores (normalizing for any differential in CPU \( ops/s\) rating) and \( m \) is per pixel appropriately normalized overhead of memory transfer delay for each image data to be transferred to GPU memory. Improvements in power efficiencies with respect to baseline implementation are expected to be proportional to the improvements in computational load. Further power gains can be achieved by appropriately spatio-temporally subsampling the video stream. Further this subsampling can be adaptive to the salient content in the video or based on the available resources. Companion work [11] focuses on details of the trade-offs of on-board and off-board computing for summarization depending upon environmental uncertainties.

VI. CONCLUSIONS AND FUTURE WORK

In this work, we describe an end-to-end UAV summarization application that provides both coverage and event summaries of the video content and associated noisy meta-data. We show that our method of exploiting noisy meta-data with coverage summarization is effective. Further, we observe that the more oblique the UAV views, the more meta-data contributes to the improvement in coverage summarization accuracy. Finally, accuracy of the panoramic coverage map also helps to improve event summarization visualization. In future, it would be desirable to empirically realize computational gains in end-to-end embedded environments summarizing video content. Designing adaptive metrics for determining subsampling rate would be a desirable investigation. Studying the tradeoffs involved in detecting and summarizing videos from swarms of UAV cameras would be an interesting follow on research topic.

ACKNOWLEDGMENT

This work is sponsored by Defense Advanced Research Projects Agency, Microsystems Technology Office (MTO), under contract no. HR0011-13-C-0022 and AFRL grant FA8750-14-2-0072. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. This document is: Approved for Public Release, Distribution Unlimited.

REFERENCES