Applying Video Summarization to Aerial Surveillance

K. Pitstick, J. Hansen, M. Klein, E. Morris, J. Vazquez-Trejo

Presenter: Jeffery Hansen, MTS – Engineer

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM18-0497

Problem

DoD Operational Deficiency

- Volume of streaming and archived surveillance video is outpacing the ability of analysts to manually monitor and view it
- Our collaborators from AFRL's Human-Centered ISR Division, confirmed there is a lack of automated tools to assist Processing, Exploitation, and Dissemination (PED) analysts in monitoring real-time video or analyzing archived video
- First task of Project Maven, an initiative to provide computer algorithms and artificial intelligence to warfighter, is to provide computer vision algorithms to assist PED analysts

Solution

Goal: To develop technologies that automatically extract information from militarily relevant video data that improve situational awareness.

One Approach: Video Summarization

- Computer vision/machine learning task to condense a long video into a shorter “trailer” which contains the key or unique segments
- A current interest area in academic communities, including the Machine Learning Department at Carnegie Mellon
- Techniques Include: (1) key frames, (2) key frame sub-shots, (3) key objects

Key Question: Can academic algorithms for video summarization be applied to military relevant video data?

Summarization Approach

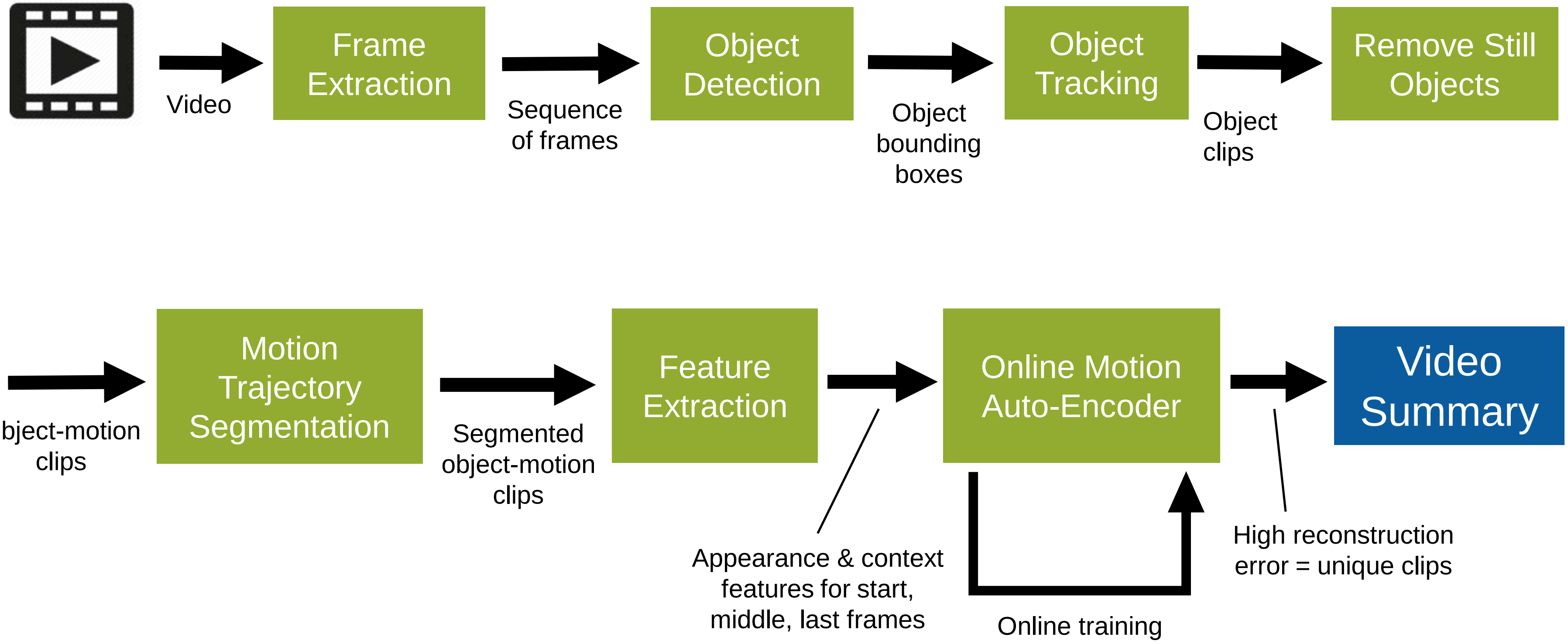
Object-Level Video Summarization via Online Motion Auto-Encoder

A novel unsupervised video summarization pipeline which functions on extracted clips of objects in motion

1. Extract clips of objects in motion from video
   - Object detection, object tracking, and object clip segmentation
2. Feed each object clips' features through auto-encoder
   - Auto-encoder attempts to reconstruct the input
3. Clips with highest reconstruction error (adjustable threshold) become the summary
   - All clips are used as online training to the auto-encoder to learn "on the fly"

CMU Machine Learning Dept. Collaborators: Xiaodan Liang and Eric Xing

Video Summarization Pipeline



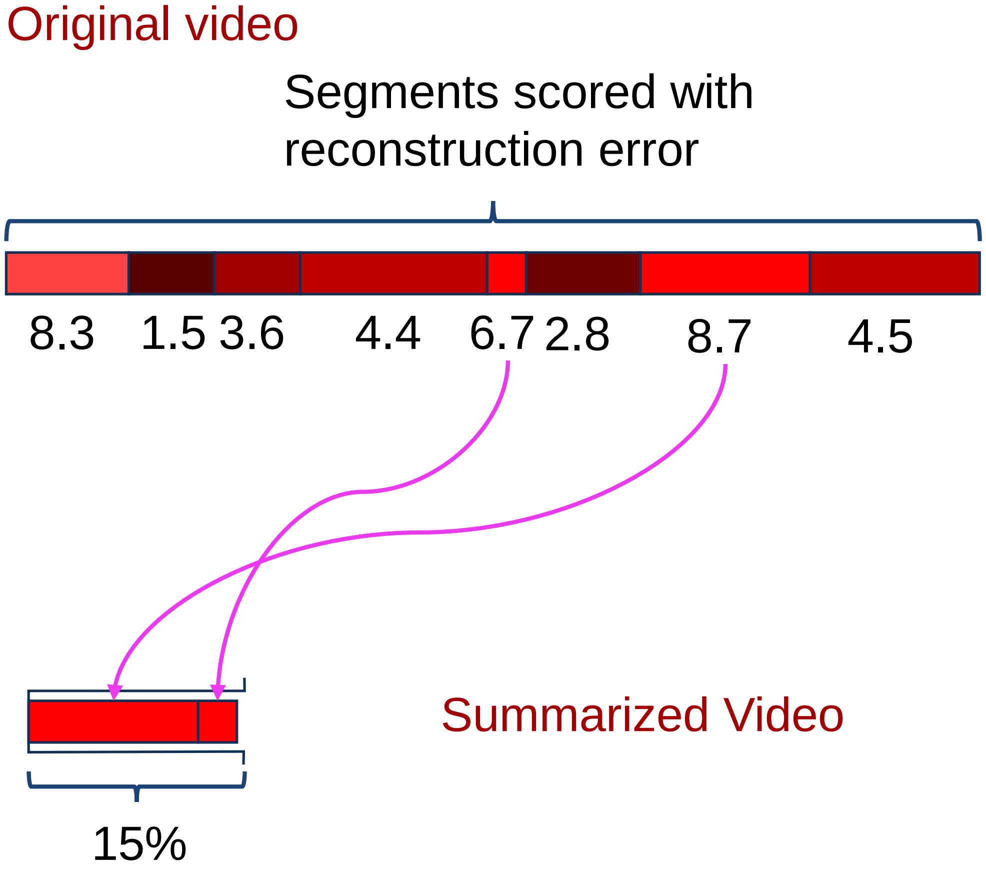
Summary Construction

Video segmented into “superframes” using techniques in [Gygli 2014].

Video summarization pipeline applied to calculate reconstruction error for superframes.

Summary length specified as percent of original video length.

Knapsack algorithm as in [Gygli 2014] applied to select superframes to combine into the summary.



Experiments

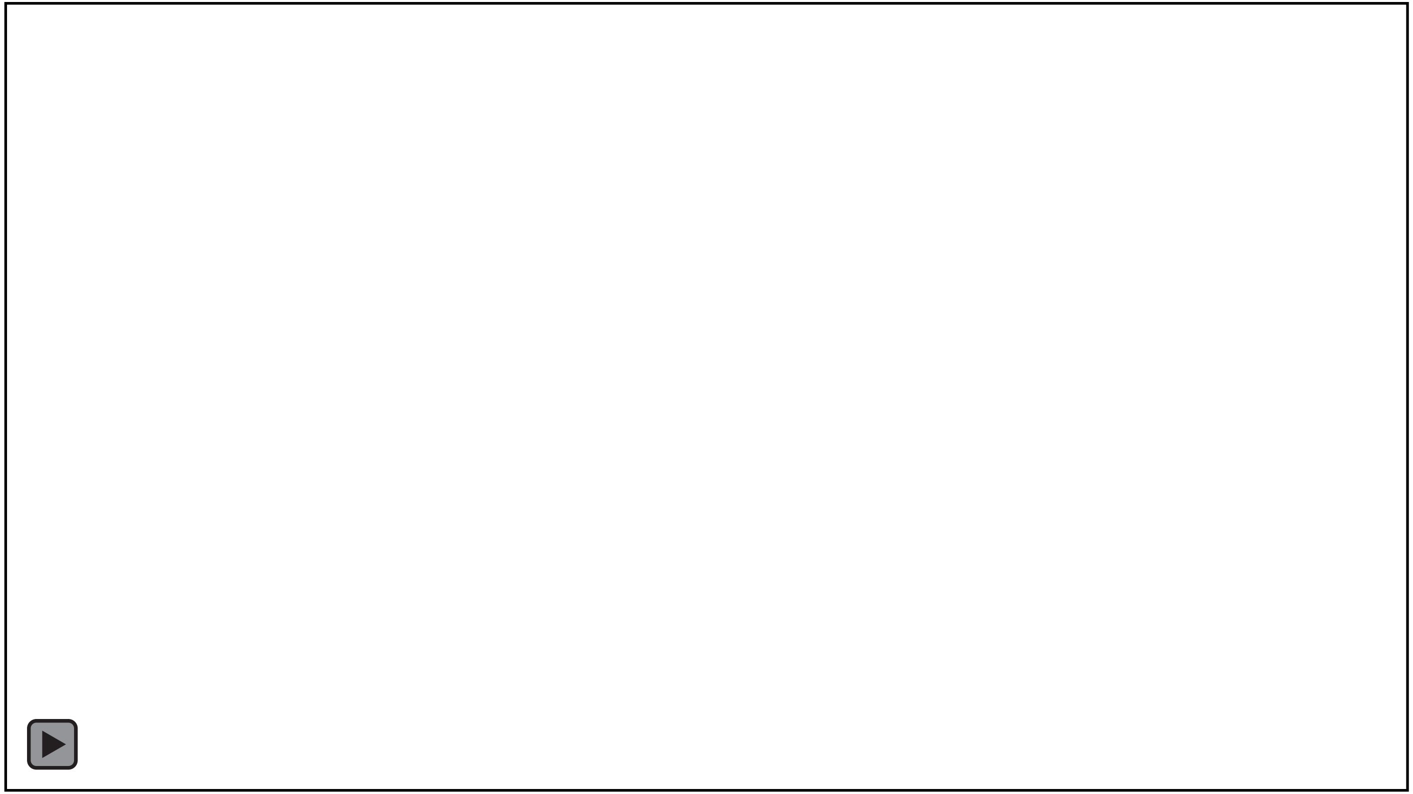
- **Datasets:** Orangeville (new), Base Jumping, SumMe, TVSum
- **Key Metrics:** Area under ROC curve (AUC), Average Precision (AP), F-measure (at threshold = 0.5)
- **Object-level:** Orangeville, **Subshot-level:** Base Jumping, SumMe, TVSum

Original: 100 seconds

From “Orangeville” dataset (described in paper submission)

Summary: ~17 seconds

Demonstration Summary Video



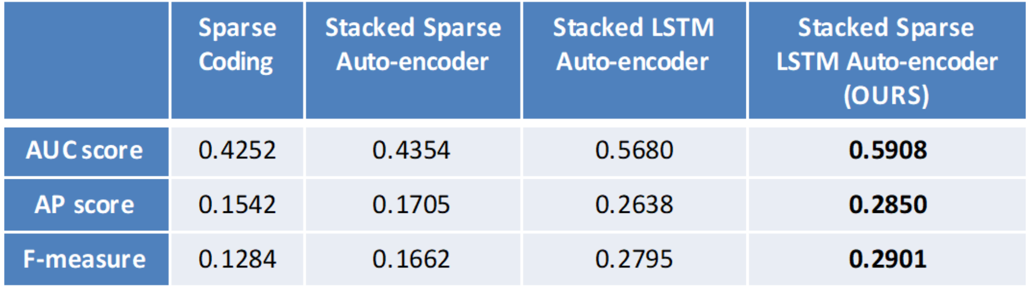
Orangeville Results

Quantitative - Table 1

- Ground-truth annotated manually for key clips (fast moving cars, people crossing road, cars turning)
- Comparison with competing unsupervised, online approaches: sparse coding, alternate auto-encoders

Qualitative – Figure 1

- 15 subjects watching original at 3x speed followed by summary
- Assign rating from 1 to 10



	Sparse Coding	Stacked Sparse Auto-encoder	Stacked LSTM Auto-encoder	Stacked Sparse LSTM Auto-encoder (OURS)
AUC score	0.4252	0.4354	0.5680	**0.5908**
AP score	0.1542	0.1705	0.2638	**0.2850**
F-measure	0.1284	0.1662	0.2795	**0.2901**

Table 1: Object-level summarization results between competing approaches on **Orangeville** dataset

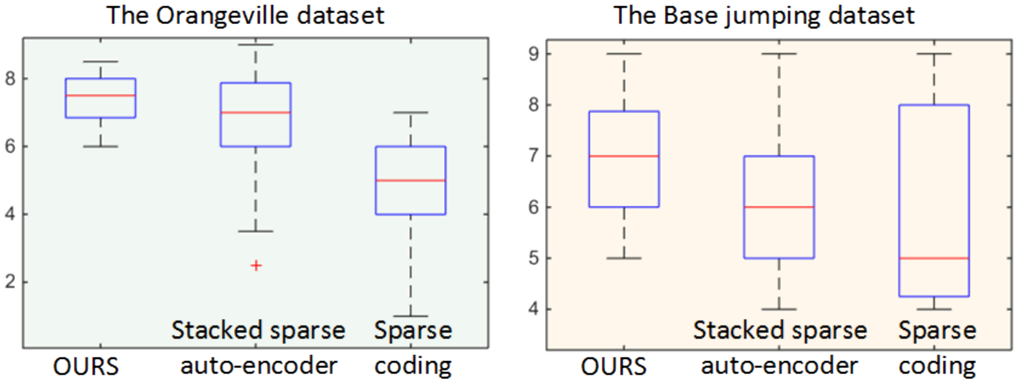


Figure 1: User study evaluation scores between competing approaches on **Orangeville** and **Base Jumping** datasets

SumMe and TVSum Results

- Adapt pipeline for subshot-level summarization to compare our auto-encoder against subshot-level approaches (e.g., TVSum, LiveLight, etc)

Method	F-measure
Video MMR	0.266
TVSum	0.266
VSUMM_1	0.328
VSUMM_2	0.337
Stacked GRU Auto-Encoder	0.354
Online Motion AE (OURS)	**0.377**

Table 1: Subshot-level summarization results on **SumMe** dataset

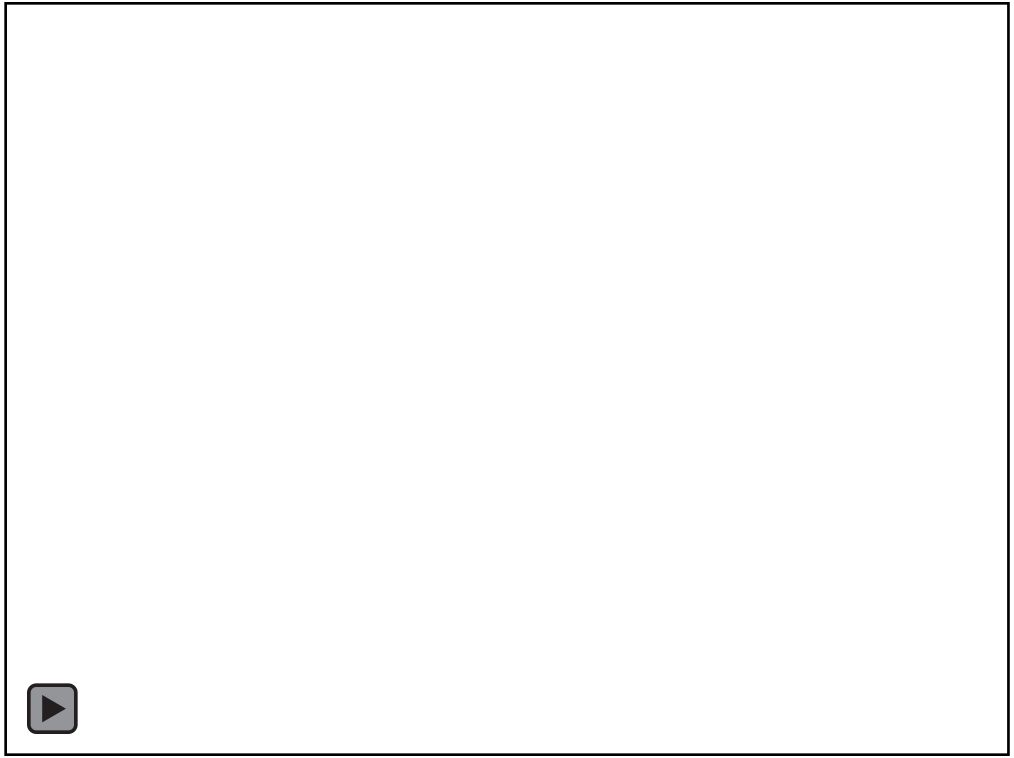
Method	F-measure
Web Image Prior	0.360
LiveLight	0.460
TVSum	0.500
Stacked GRU Auto-Encoder	0.510
Online Motion AE (OURS)	**0.515**

Table 2: Subshot-level summarization results on **TVSum** dataset

Analyzing DoD Full Motion Video (FMV)

While results are promising, DoD full motion video (FMV) differs from ground surveillance

- Mix of Infra-red (IR) and electro-optical (EO) with switches between them.
- Moving camera vs. stationary camera
- Aerial viewpoint vs. ground viewpoint
- Changing zoom levels and rapid panning
- Most images in grayscale
- Inconsistent and shifting lighting and perspective.



Publicly released by U.S. Central Command Public Affairs on CENTCOM’s website - http://www.centcom.mil/MEDIA/VIDEO-AND-IMAGERY/VIDEOS/videoid/520438/

[DISTRIBUTION STATEMENT A] Approved for public release and unlimited distribution.

FBI Surveillance Video

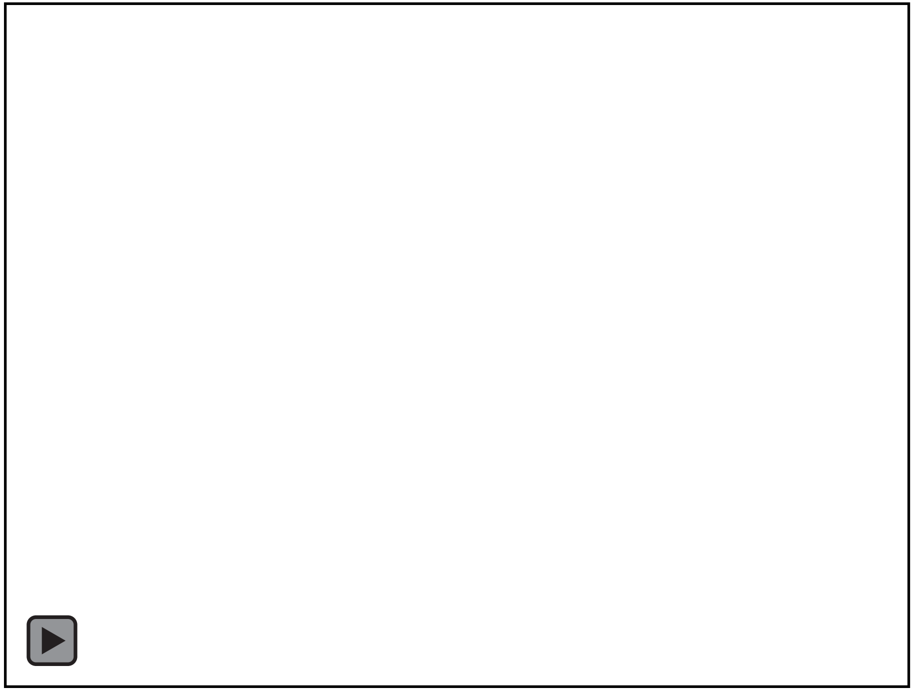
Used FBI video of protests in Baltimore as proxy for aerial surveillance dataset

- Divided into approximately 30 min. segments.
- Mix of EO and IR video
- Vehicles could be detected with precision of 0.89
- However, state-of-the-art ML techniques not currently able to detect and track people.
  - Could not apply full-pipeline due to the “small people” problem.
  - Applied autoencoder to full video instead of object-level clips.

5% Summary Result

Original 30 min. video was reduced to 1 min. 30 sec. summary.

- At least some of most important scenes included in summary (e.g., bright flashes, potentially pyrotechnics)
- Summary gives a sense of the kind of scenes in the video, though it may be difficult to determine the “plot”.
- Potentially useful for triage of a large collection of videos.
- Deeper analysis techniques may be needed for real-time video.



Processing Time

Total time to process a 30 min. video was approximately 70 min.

- Majority of processing time was the frame extraction and segmentation.
  - Due to n^2superframe comparison needed to determine segmentation points.
- Core summarization time required approximately 8 min. of processing.

Process	Time in Minutes
Frames	14:19
Segments	48:17
Features	2:55
Layer 1	1:51
Layer 2	0:28
Layer 3	0:30
Summarization	2:16
Total	70:36

Conclusion & Challenges

- Recognizing military-relevant objects at multiple scales
  - particularly detection of small scale objects
- Adapting to the characteristics of aerial surveillance platforms that move
  - Changing viewpoints, panning/zooming, IR/EO changes, etc.
- Recognizing activities
- Processing in real-time or near real-time
- Rapid training of learning algorithms with relatively few labeled training examples
- Provision of a chain of reasoning for algorithms, including: confidence in the algorithm (based on internal algorithm scoring and on historic performance); explanations of key factors in algorithm decisions; and traceback to raw video data
- Higher-level analysis of video and metatdata that recognizes patterns over time and space to provide (for example) information about movement, trends, patterns of life, anomalies, and similar clusters of objects, activities, people and events.

Contact Information

Presenter

Jeffery Hansen

MTS – Engineer

Email: jhansen@sei.cmu.edu

SEI Team

- Ed Morris
- Kevin Pitstick
- Jeffery Hansen
- Mark Klein
- Mike Konrad
- Keegan Williams
- Javier Vazquez-Trejo