Cross-Camera Video Analytics in Large Enterprise Camera Deployments

Samvit Jain[1,2]; Ganesh Ananthanarayanan[2]; Junchen Jiang[2,3]; Yuanchao Shu[2]
[1]UC Berkeley, [2]Microsoft Research, [3]University of Chicago

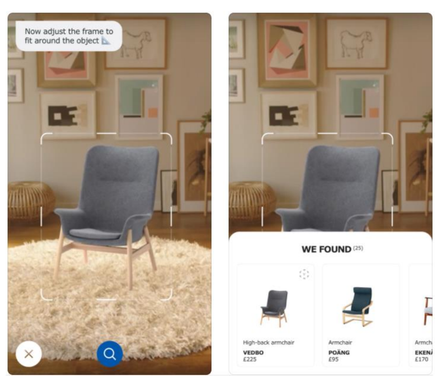
Introduction

Two trends in video analytics are triggering an increase in size and prevalence of enterprise camera deployments (both commercial and government):

1) **Falling camera costs** – an HDTV-quality camera with on-board SD card storage costs $20 today, down from $1,500 about three years ago.

2) **Advances in computer vision** – using neural nets (NN), can now: scan passenger identities via facial recognition (KLM's biometric passports), take a photo of a furniture item and search for it in an online catalog (IKEA's Place AR app), and convert a 2D video into a live 3D visualization in a AR device (UW and Facebook's 'Soccer on Your Tabletop').

KLM's biometric passport



IKEA's visual search app

UW's Soccer On Your Tabletop

Impact of larger camera deployments:

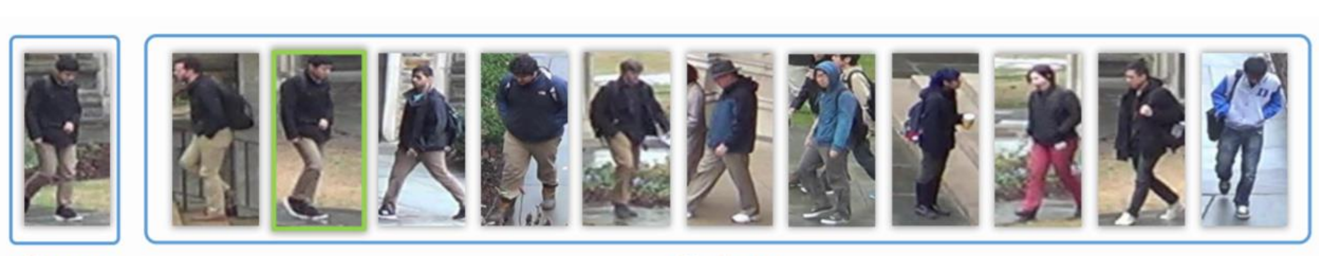
1) **Proportional increase in compute requirements** – as deployments scale, increased pressure placed on edge clusters, requiring greater resource provisioning (e.g. expensive GPUs), compute time, and human attention.

2) **Contention for human attention** – as deployments scale, human attention becomes the scarce resource. Central question: can we add more cameras, to cover larger areas, without requiring more human operators?

Multi-camera tracking

Problem template: track a person P through a camera network, in forward direction ("real-time tracking") or backward direction ("investigative search"), returning frames F in which person P is found. Track until P exits the network.

Multi-camera person tracking is built on a computer vision primitive known as person *re-id*entification. Re-id involves computing features on a query image **q** and each image $\{\mathbf{g_i}\}$ in a gallery, and ranking each $\mathbf{g_i}$ by its *feature distance* to **q**.



Given a query image q, rank images in a 'gallery' based on their similarity to q

Dataset

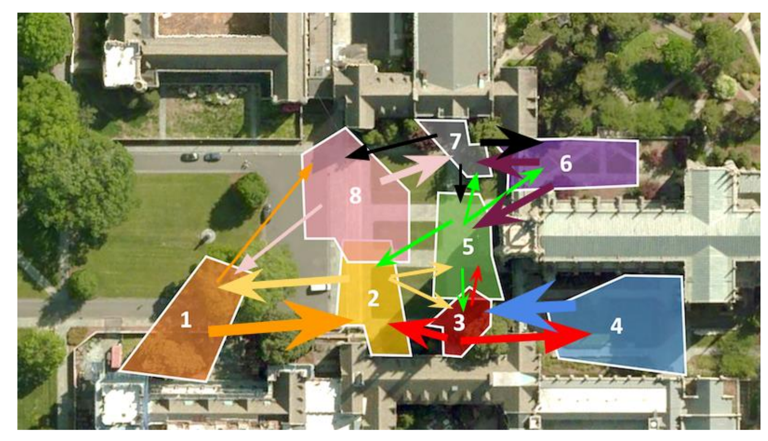
DukeMTMC dataset – footage from eight cameras installed on Duke U. campus

Opportunities

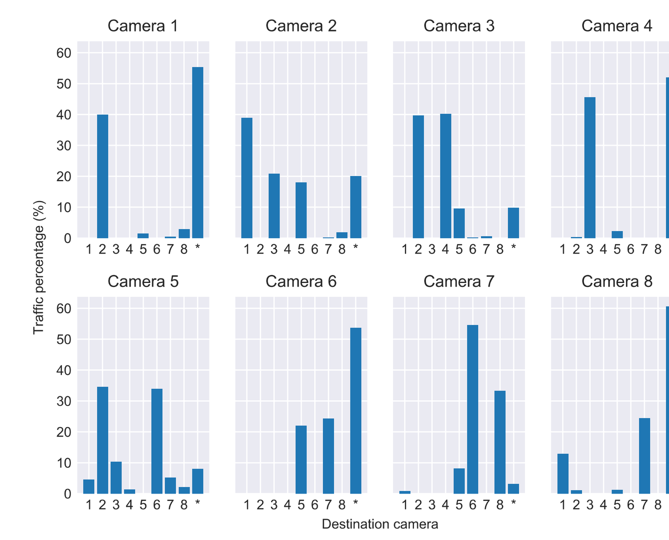
We leverage **spatio-temporal correlations** in large camera deployments to:

1) **Reduce resource cost** – by localizing a target object or person to a small group of cameras, can eliminate possibility of appearance in most other cameras in near-future. This in turn means that expensive, deep NN models can be turned off (or e.g. run at lower frame rates) on those video feeds.

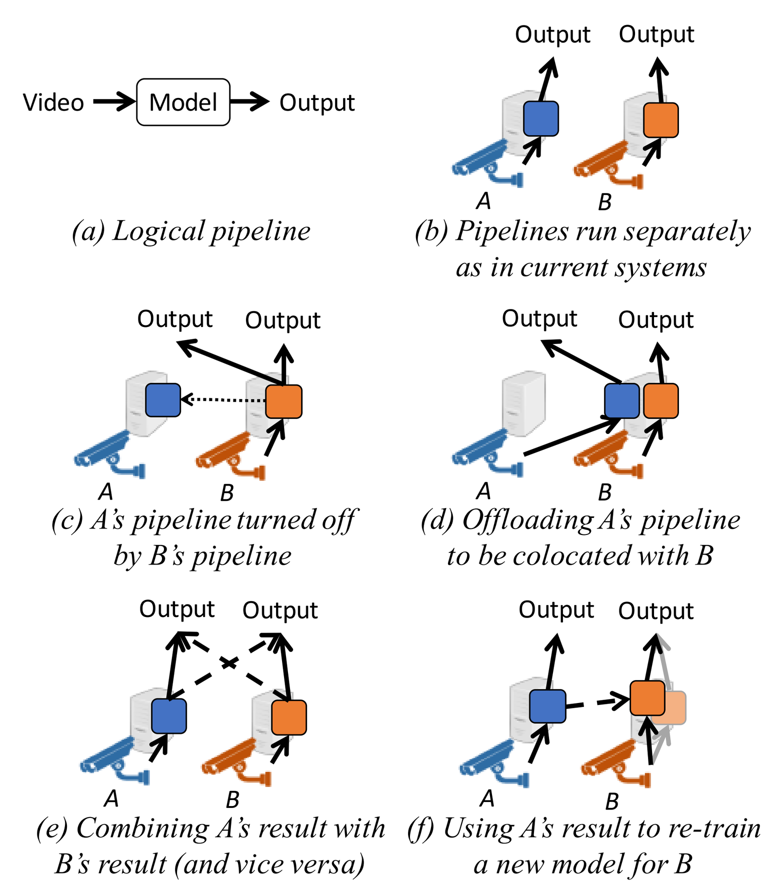
2) **Improve inference accuracy** – by pruning the search space, can reduce the probability of matching against incorrect detections (false positives), which dislodge subsequent tracking. Experiments confirm that this is a real and significant effect in tracking scenarios (see results).



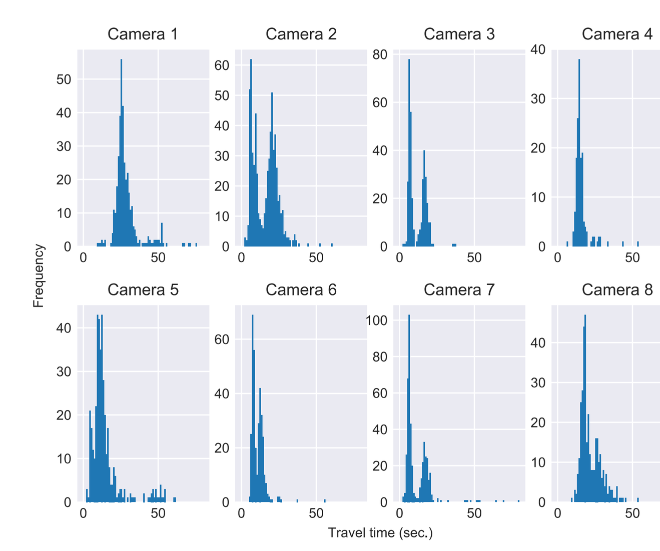
Duke dataset traffic visualization



Spatial traffic patterns (Duke)



Cross-camera pipeline opportunities



Temporal traffic patterns (Duke)

As the plots above indicate, foot traffic in the Duke dataset demonstrates significant spatio-temporal localization. Most traffic *leaving* a camera appears in only 2-3 other cameras (top). Travel times tend to fall in a narrow window that varies across cameras (bottom).

Results

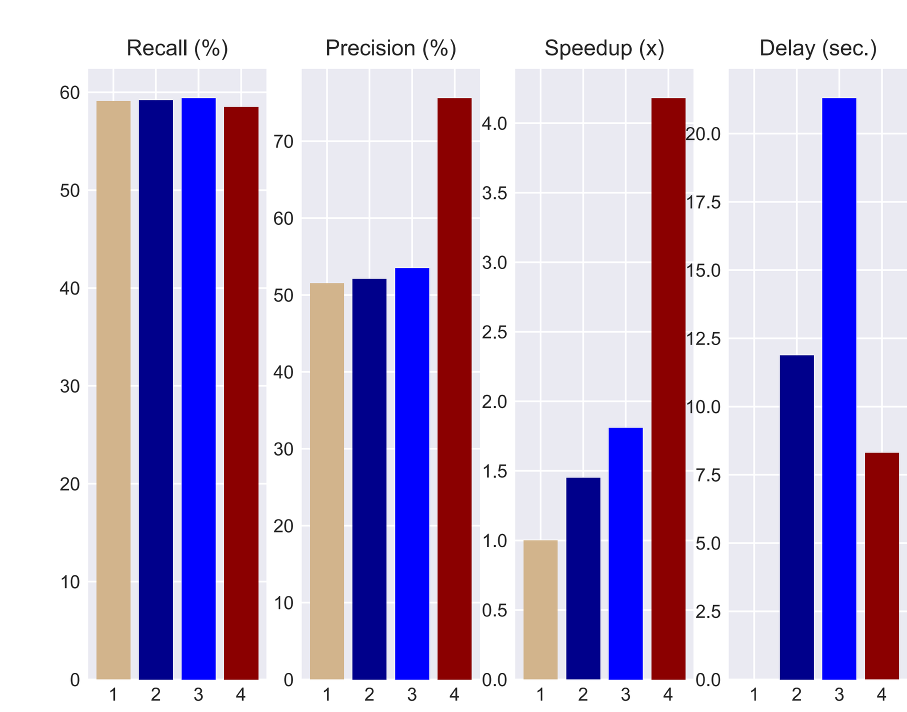
Four schemes:

1) **Baseline** – searches all 8 cameras until t=exit_time*

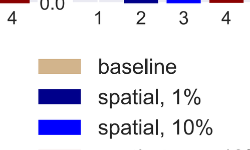
2) **Spatial filter, 1%** - only searches cameras expecting ≥1% of traffic

3) **Spatial filter, 10%** - only searches cameras expecting ≥10% of traffic

4) **Spatio-temp filter, 10%** - in addition to (3), only searches during time window containing [0, 99%] of traffic



*Defined as the average exit time across all 8 cameras in scheme (4).



Observations:

1) **Speedup** factor increases from 1.0 (baseline) → 1.8x (spatial filter) → 4.2x (ST-filter) on 8 cams.

2) Surprisingly, **recall** improves slightly with spatial filtering over the baseline, from 59.1% → 59.4%.

3) Spatio-temporal filtering improves **precision** over the baseline, from 51.5% → 75.6%. Pruning the search space reduces false positive matches.

4) The price of reduced resource usage with ST-filtering is **delay** (lag between tracking and video), from matches found later in pruned search space.

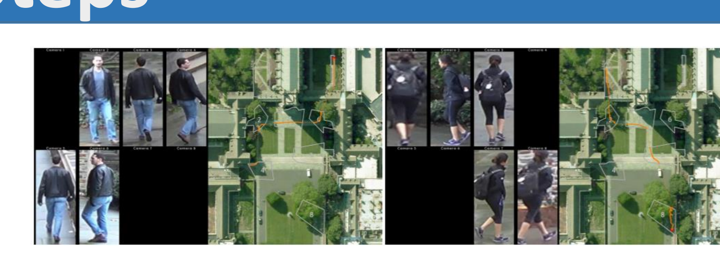
Next Steps

Follow-up work:

1) Use multi-target tracking to build ST-model

2) Mitigate delay with fast-forward search

3) Run scaling experiment on 15+ cam dataset



Contact

Samvit Jain
RISELab
University of California, Berkeley
465 Soda Hall, MC-1776
Berkeley, CA, 94720-1776

Email: samvit@eecs.berkeley.edu

References

1. Ristani, E., Solera, F., Zou, R. S., Cucchiara, R., Tomasi, C. *Performance Measures and a Data Set for Multi-Target, Multi-Camera Tracking*. In: Workshop on Benchmarking Multi-Target Tracking at ECCV. (2016)
2. Ristani, E., Tomasi, C.: *Features for Multi-Target Multi-Camera Tracking and Re-Identification*. In: CVPR. (2018)
3. Xiao, T., Li, S., Wang, B., Lin, L., Wang, X. *Joint Detection and Identification Feature Learning for Person Search*. In: CVPR. (2017)
4. Zheng, L., Yang, Y., Hauptmann, A. G. *Person Re-identification: Past, Present and Future*. arXiv: 1610.02984. (2016).
5. Zhang, H., Ananthanarayanan, G., Bodik, P., Philipose, M., Bahl, P., Freedman, M. J. *Live Video Analytics at Scale with Approximation and Delay-Tolerance*. In: NSDI. (2017)
6. Kang, D., Emmons, J., Abuzaid, F., Bailis, P., Zaharia, M. *NoScope: Optimizing Neural Network Queries over Video at Scale*. In: VLDB. (2017)
7. Hsieh, K., Ananthanarayanan, G., Bodik, P., Venkataraman, S., Bahl, P., Philipose, M., Gibbons, P. B., Mutlu, O. *Focus: Querying Large Video Datasets with Low Latency and Low Cost*. In: OSDI. (2018)
8. Jiang, J., Ananthanarayanan, G., Bodik, P., Sen, S., Stoica, I. *Chameleon: Video Analytics at Scale via Adaptive Configurations and Cross-Camera Correlations*. In: SIGCOMM. (2018)