Spatula: Efficient cross-camera video analytics on large camera networks

Xun Zhang

Samvit Jain (UC Berkeley)
Xun Zhang (Univ of Chicago)
Yuhao Zhou (Univ of Chicago)
Ganesh Ananthanarayanan (Microsoft Research)
Junchen Jiang (Univ of Chicago)
Yuanchao Shu (Microsoft Research)
Victor Bahl (Microsoft Research)
Joseph Gonzalez (UC Berkeley)

Background

Computer Vision is improving

Advances in computer vision

- **Image** – classification, object detection
- **Video** – action recognition, object tracking

Rise of large video analytics operations

- **London** – 12,000 cameras on rapid transit system
- **Chicago** – 30,000 cameras across city
- **Paris** – 1,500 cameras in public hospitals

Background

CV is a powerful tool

BUT

It is challenging to scale it to proliferating large camera deployments.

Huge Cost of current Computer Vision task on large camera deployments

For Chicago Public Schools, 7000 security cameras installed as a counter to crimes.

- $28 million in GPU hardware (at $4,000 / GPU)
- $1 million/month in GPU cloud time (at $0.9 / GPU hour)

Cross-camera analytics

Problem statement

- Given: instance of query identity Q
- Return: all later frames in which Q appears

Application space

Cross-camera video analytics is important!

Many applications rely crucially on cross-camera video analytics

- Real-time search: Track threat (e.g. AMBER alert)
- Post-facto search: Investigate crime (e.g. terrorist attack)
- Trajectory analysis: Learn customer behavior

Cross-camera analytics

When it comes to large camera deployments.

Challenges: High compute cost and low inference accuracy

How to go?

Cross-camera analytics

Prior work falls short of addressing this challenge.

Methods in recent systems to reduce cost:

- Frame sampling
- Cascade filter for discarding frames.

However

Just cost/accuracy tradeoffs

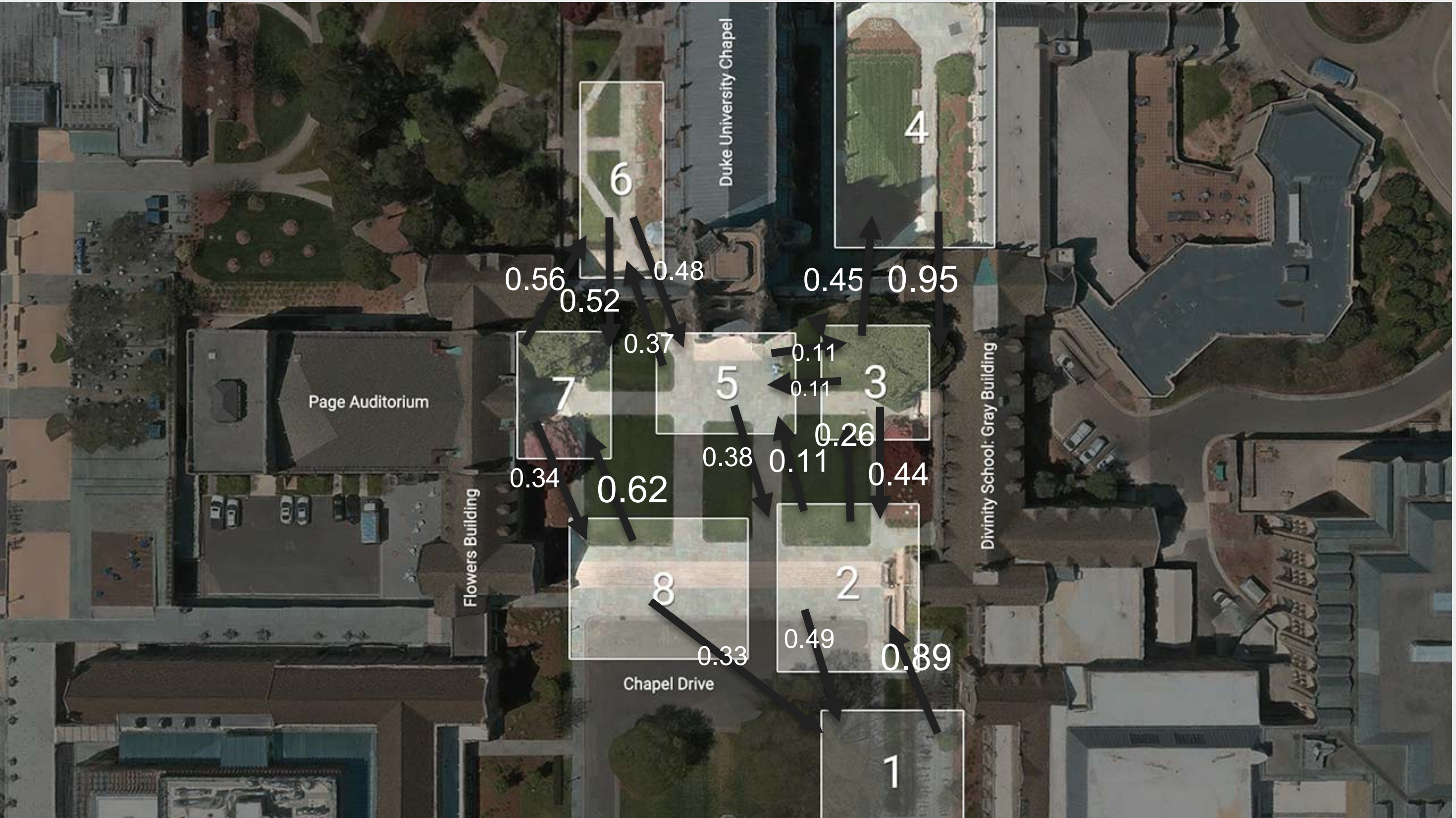
Optimization of one video stream is independent of other streams.

Compute/network cost grows with the number of cameras,

and with the duration of the identity’s presence in the camera network.

Spatial correlations among cameras

Challenges: High compute cost and low inference accuracy



Cam1 → Cam2 0.89 means 89% of all traffic leaving Camera 1 first appears at Camera 2

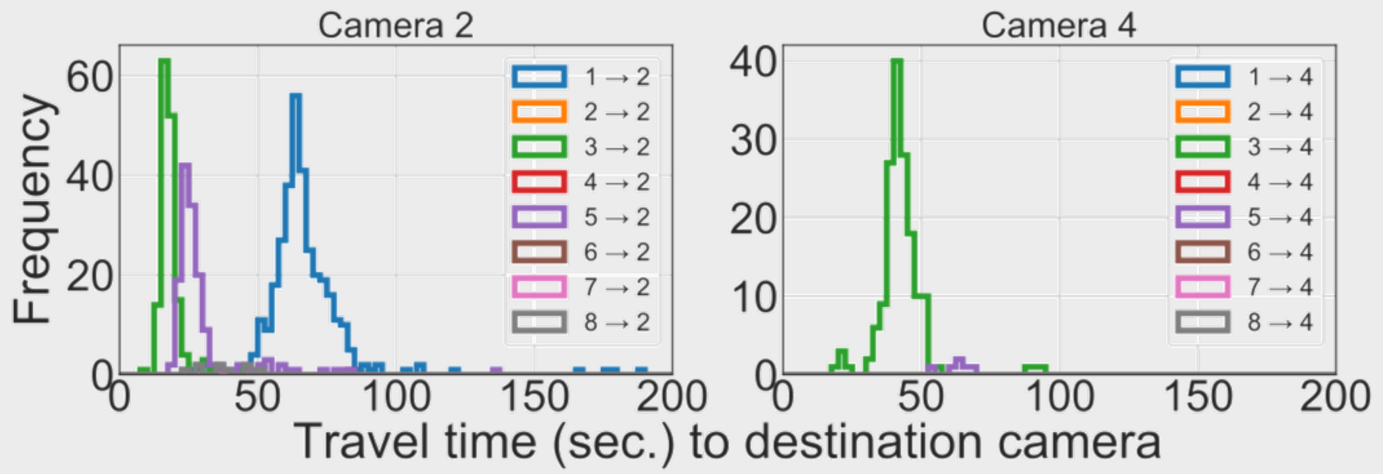
Geographical proximity is not a good filter, eg. Cam 5

Learning these patterns in a **data-driven** fashion is a more robust approach!

Temporal correlations among cameras

The velocity of the object is within a certain range.

The travel times between cameras can be clustered around a mean value.



For objects which leave from camera 1 and next appear at camera2, the travel times are likely clustered around a mean value 66.

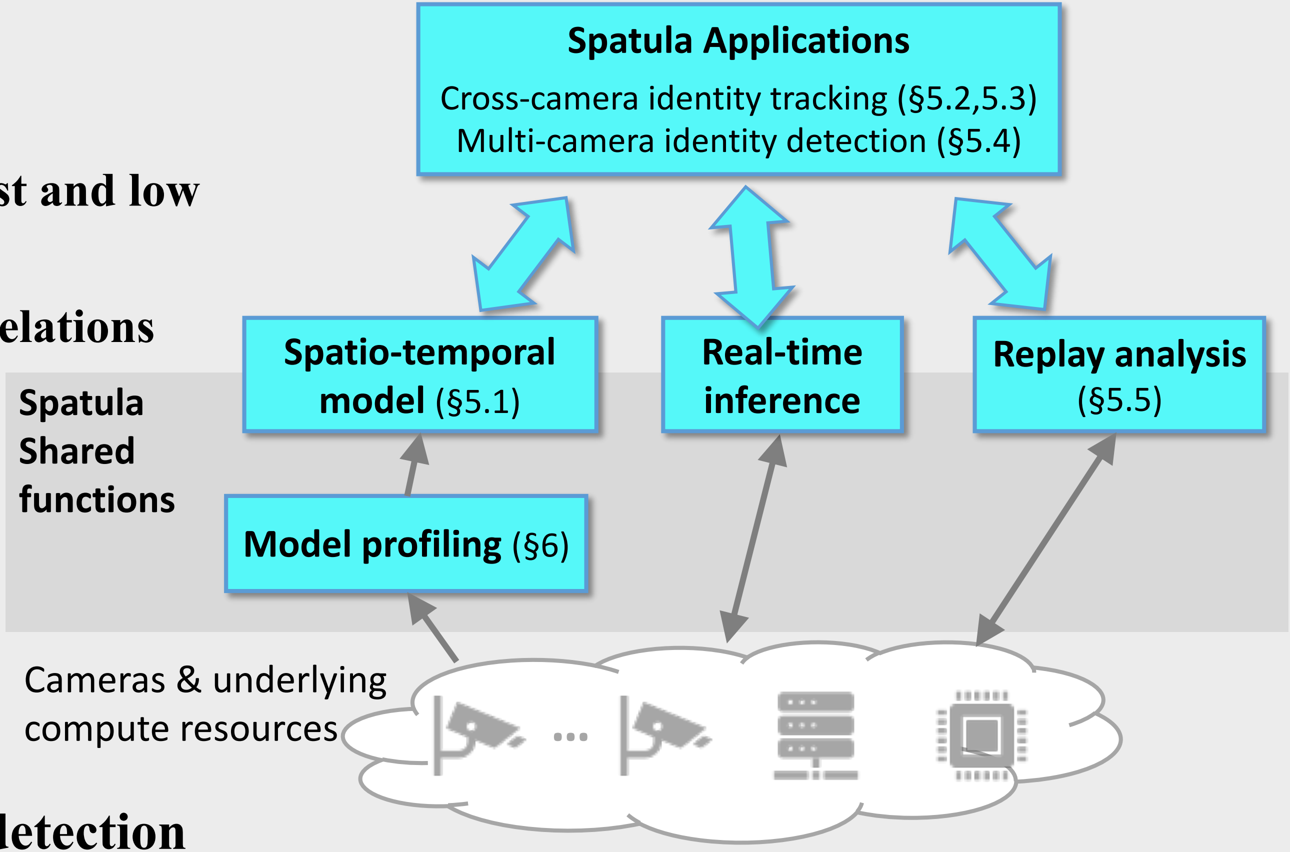
In the DukeMTMC dataset, the average travel time between all camera pairs is 44.2s , and the standard deviation is only 10.3s (or only 23% of the mean)

Spatula

Challenges: High compute cost and low inference accuracy

Methods: Using physical correlations to prune the search space

- **Spatio-temporal model**
- **Replay analysis**
- **Multi-camera identity detection**



Spatio-temporal model

Definition of spatial correlation

$$S(c_s, c_d) = \frac{n(c_s, c_d)}{\Sigma_i n(c_s, c_i)}$$

$n(c_s, c_d)$: the number of individuals leaving the source camera c_s's stream
for the destination camera c_d

Definition of temporal correlation

$$T(c_s, c_d, [t_1, t_2]) = \frac{n(c_s, c_d, t_1, t_2)}{n(c_s, c_d)}$$

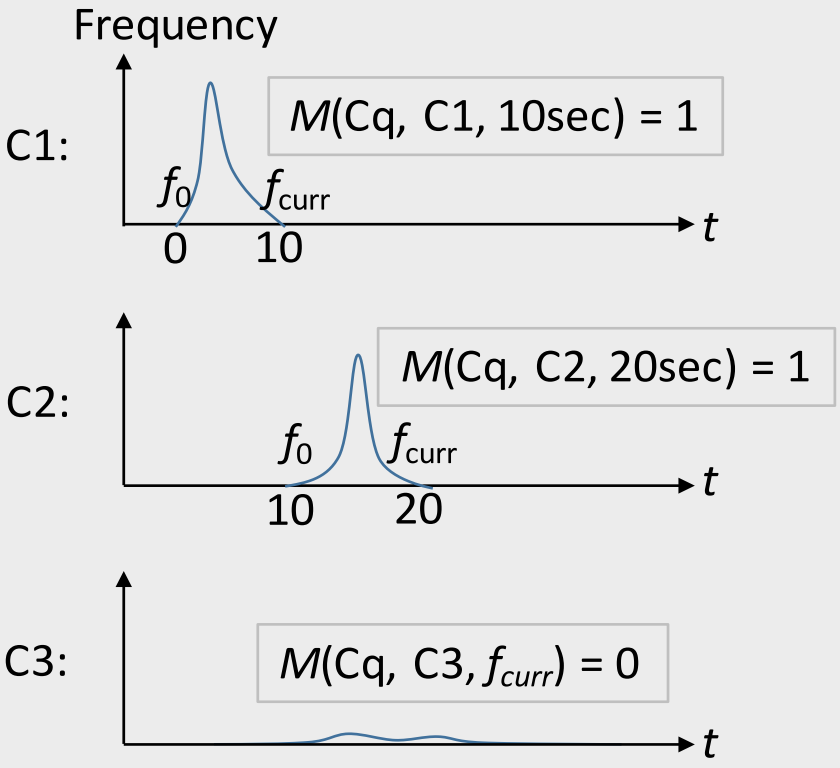
$n(c_s, c_d, t_1, t_2)$: individuals reaching c_d from c_s within a duration window $[t_1, t_2]$

Spatio-temporal model

$$M(c_s, c_d, f_{curr}) = \begin{cases} 1, & S(c_s, c_d) \geq s_{thresh} \quad and \quad T(c_s, c_d, [f_0, f_{curr}]) \leq 1 - t_{thresh} \\ 0, & otherwise \end{cases}$$

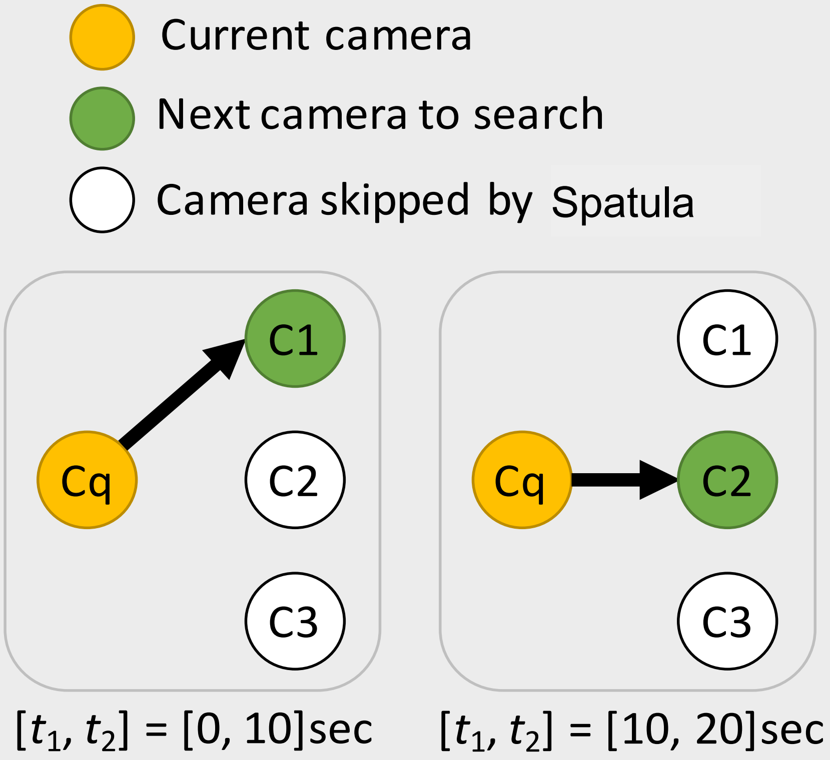
f_0 is the frame index at which the first historical arrival at c_d from c_s was recorded.

Spatio-temporal model



(a) Spatio-temporal correlations

Spatio-temporal model



(b) Pruned search based on spatio-temporal model

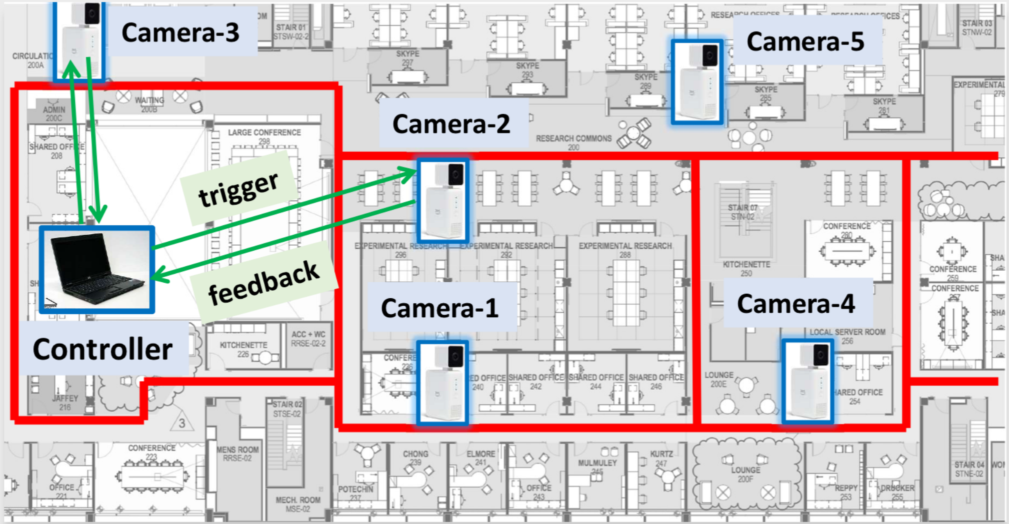
Experimental setup

Dataset: AnonCampus, DukeMTMC, Porto, Beijing

Metrics: Compute cost, Network cost, Recall, Precision, Delay

Baseline:

- Baseline-all: Searches for query identity q in all the cameras at every frame step.
- Baseline (GP): Searches for query identity q only in the cameras that are in geographical proximity to the query camera at every frame step.

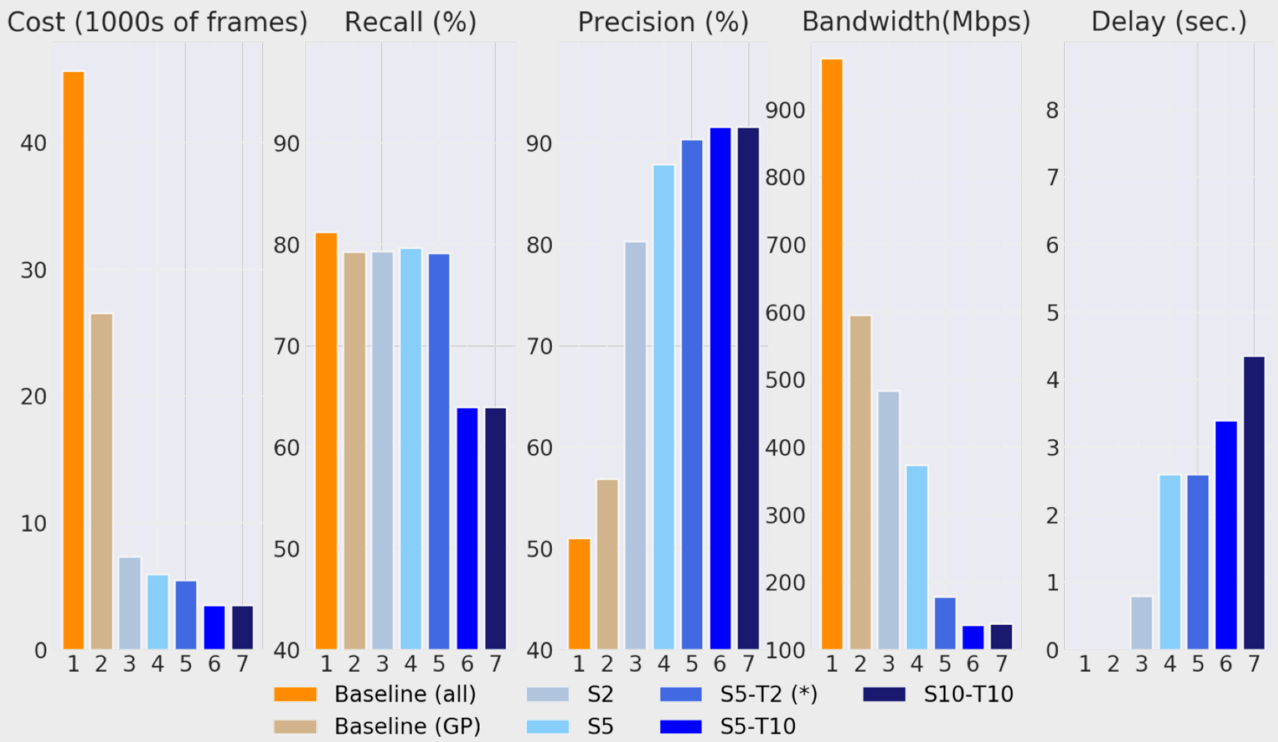


AnonCampus Dataset, we developed 5 cameras at Uchicago, JCL.

Experimental result

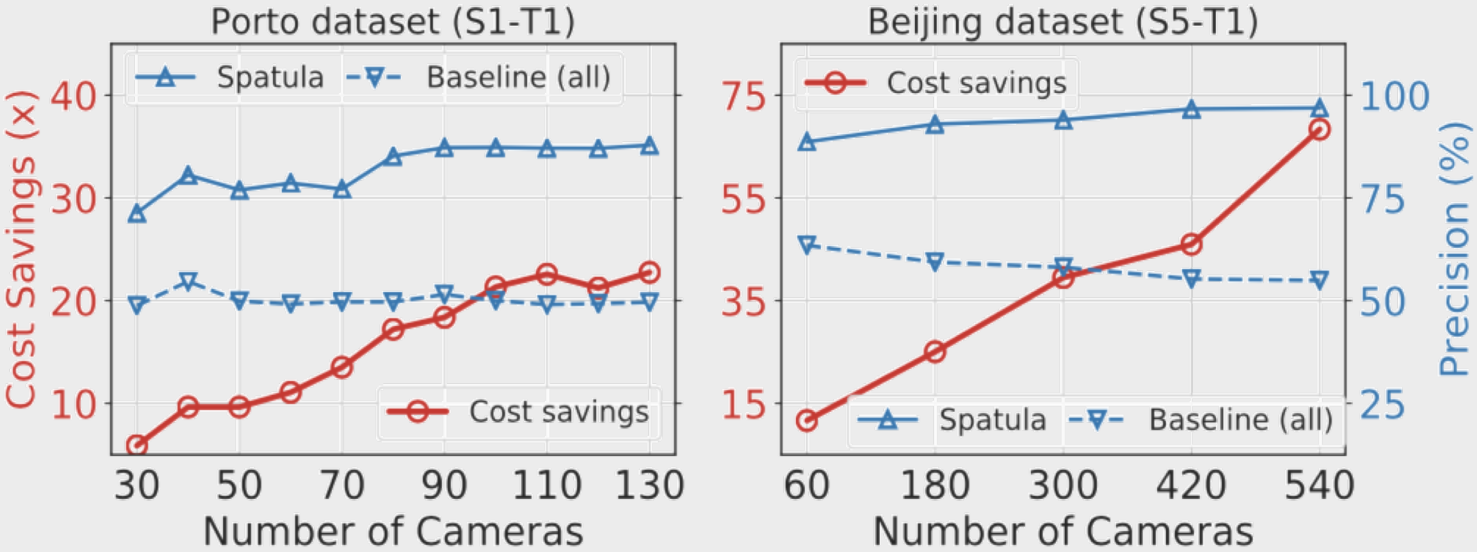
Results for different versions of spatula and baseline.

For spatula, each version is coded as Ss-Tt, where s indicates the spatial filtering threshold and t indicates the temporal filtering threshold.



Experimental result

Cost savings and precision of Spatula with increasing number of cameras



Experimental result

Highlight results about spatula on 4 datasets.

Dataset	Comp.sav.	Netw.sav.	Prec.	Recall
AnonCampus	3.4x	3.0x	21.3% ↑	2.2% ↓
DukeMTMC	8.3x	5.5x	39.3% ↑	1.6% ↓
Porto	22.7x	n/a	36.2% ↑	6.5% ↓
Beijing	85.5x	n/a	45.5% ↑	7.3% ↓

Key Takeaways

Problem:

cross-camera analytics is data and compute intensive

Our Approach:

computation can be drastically reduced by exploiting the spatio-temporal correlations

Key results:

spatula reduces compute load by 8.3x on an 8-camera dataset, and by 23x - 86x on two datasets with hundreds of cameras

Spatula: Efficient cross-camera video analytics on large camera networks

Xun Zhang

Samvit Jain (UC Berkeley)
Xun Zhang (Univ of Chicago)
Yuhao Zhou (Univ of Chicago)
Ganesh Ananthanarayanan (Microsoft Research)
Junchen Jiang (Univ of Chicago)
Yuanchao Shu (Microsoft Research)
Victor Bahl (Microsoft Research)
Joseph Gonzalez (UC Berkeley)

Spatula: Efficient cross-camera video analytics on large camera networks

Thanks!