Opportunities and Needs for the Visual Cloud

Kayvon Fatahalian
CMU

http://www.istc-cc.cmu.edu/

Intel Science & Technology
Center for Cloud Computing

My background

- High-performance systems for computer graphics
  - Real-time rendering algorithms
  - Parallel programming systems for supercomputing and heterogeneous machines: CPUs+GPUs
  - Optimizing compilers for high-level graphics languages
  - GPU hardware architecture

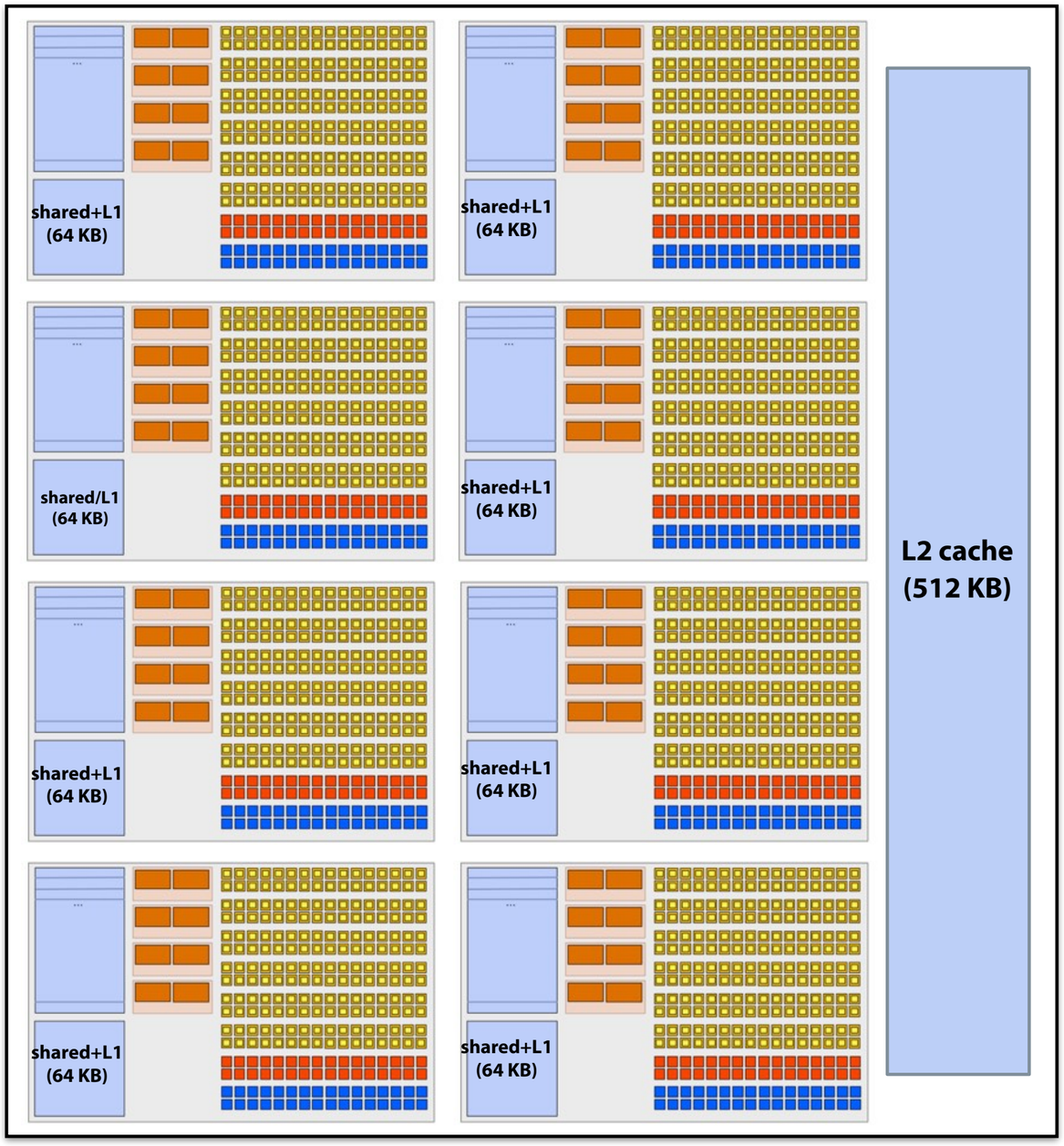
- Growing area of personal focus: “internet-scale” image processing and analytics

Our community

- "Game developer" mentality is pervasive in this community: **pack flops into system via heterogeneity/specialization + use every flop you can get**

- Definition of high performance = application realizes significant fraction of peak arithmetic capability of instruction pipelines
  - e.g., latency stalls of concern are due to branch mispredicts, jumps due to virtual function calls, or waiting on DRAM or the LLC (as opposed to disk/SSD I/O latency)

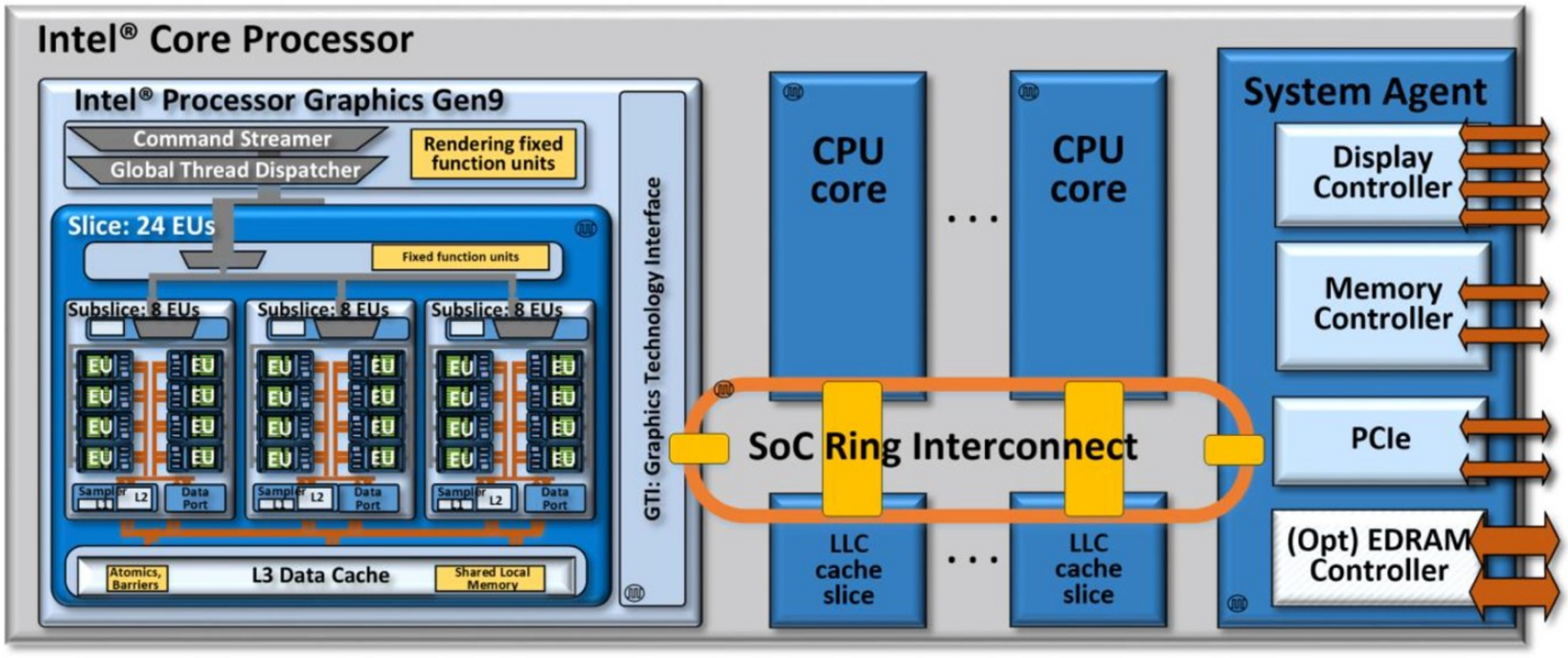
Modern GPU?



- A many-core, interleaved multi-threaded processor, featuring wide SIMD instruction support
- With some fixed-function logic for domain-specific data compression and a few common graphics primitives
- NVIDIA GTX 680: (2012)
  - 8 “core” chip
  - 64-way multi-threading per core
  - Threads issue 32-wide SIMD instr.

Integrated Gen 9 GPU (2015)

- Sits on ring bus on Core i7 architecture
- Shares physical memory and LLC with Intel CPU
- **GPU caches are coherent with CPU caches**



- Intel HD 530 graphics
  - 24 “cores” @ 1.15GHz
  - 7 threads per core, 8 or 16-wide SIMD instructions
  - Note: near-future SKUs will approach 1 TFLOP

Web-scale visual computing today

Ingesting/serving the world's photos

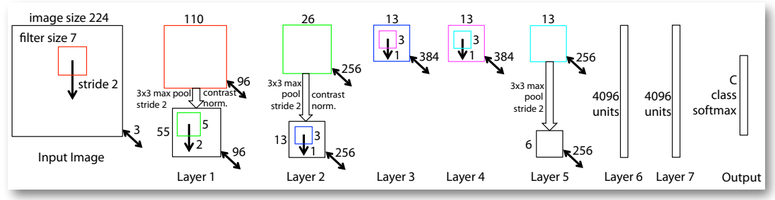
2B photo uploads and shares per day [FB2015] across Facebook sites (includes Instagram +WhatsApp)

Streaming video

Youtube 2015: 300 hours uploaded per minute [Youtube]

Cisco VNI projection: 80-90% of 2019 internet traffic will be video. (64% in 2014)

Deep learning on large-scale image collections



Distributed optimization:
DistBelief
HogWild
Parameter Server
Project Adam

Scale to more users / more photos by improving efficiency

Ingesting/serving
the world's photos

2B photo uploads and shares per day [FB2015] across Facebook sites (includes Instagram +WhatsApp)

Facebook transcodes images (resizes, recompresses) on the fly as they are served to precisely meet screen size / bandwidth / latency requirements of the user. [Cabral15]

It would be attractive to more aggressively use throughput-maximized processor architectures for these tasks

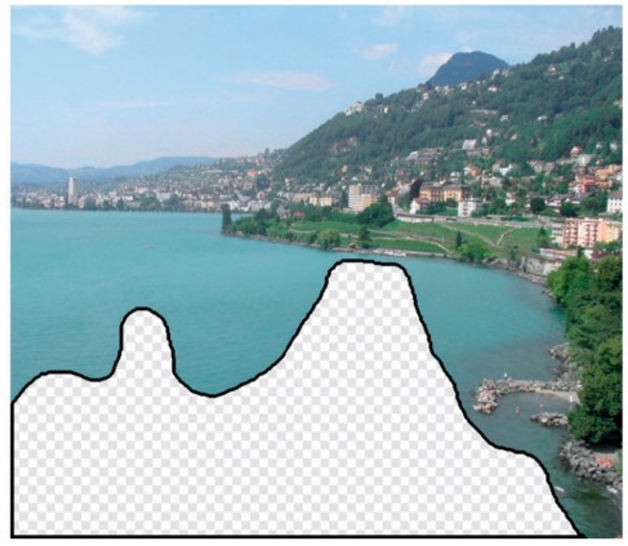
Scale functionality: perform more sophisticated processing to make photos better

Ingesting/serving the world's photos

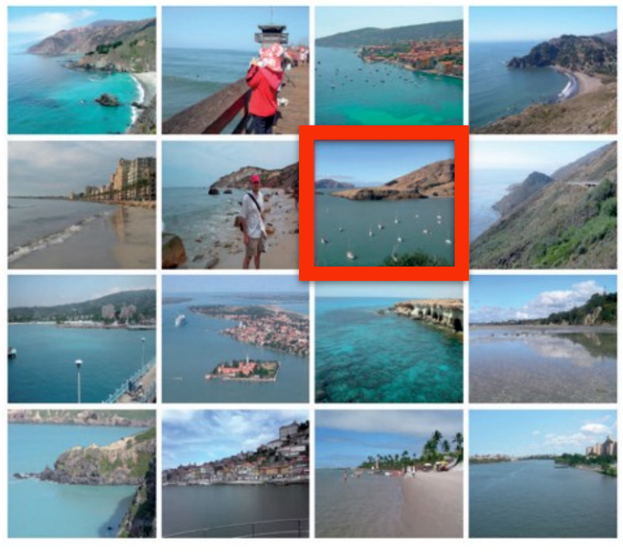
2B photo uploads and shares per day [FB2015] across Facebook sites (includes Instagram +WhatsApp)

Photo "fix up" [Hayes, Efros]

My bad photo



Part to fix



Similar photos others have taken

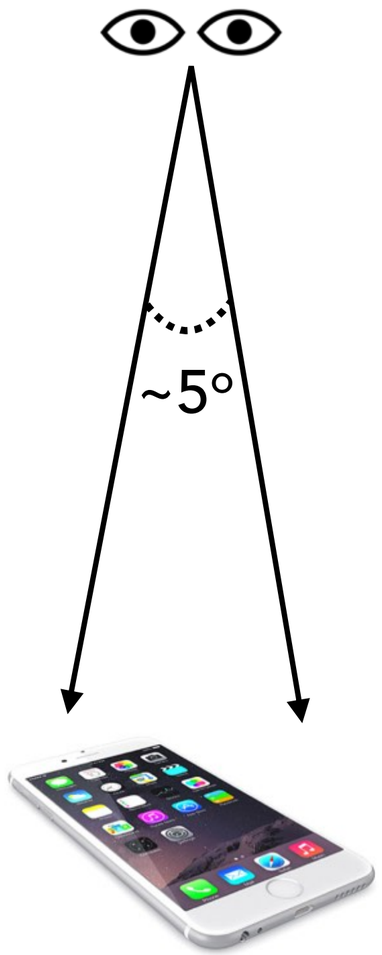
Fixed!

New opportunities

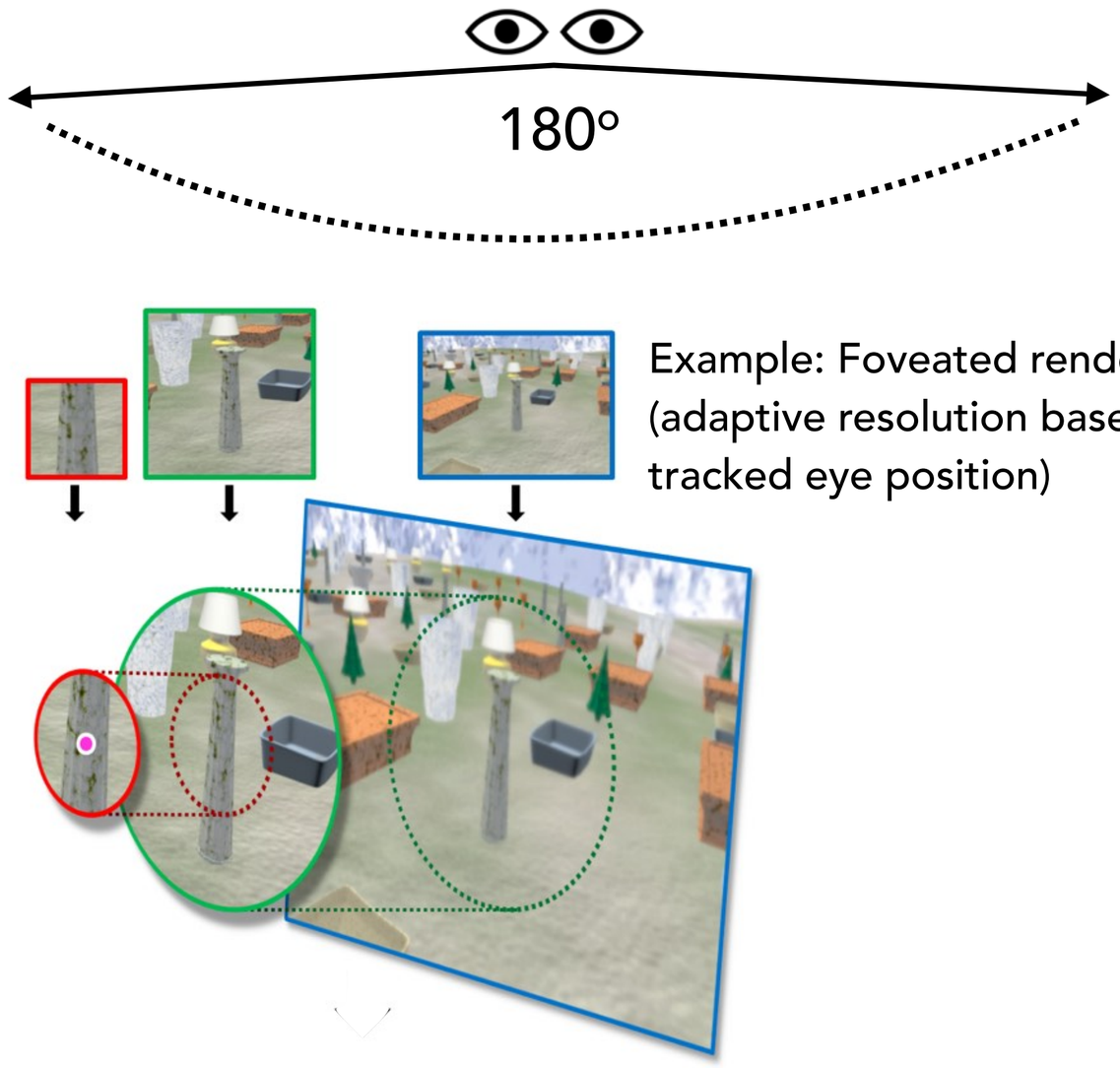
Virtual reality

Latency is king: Allowed at most 20-25ms between motion of head to emission of photons from display that reflect this head movement

VR: exceptionally high pixel counts



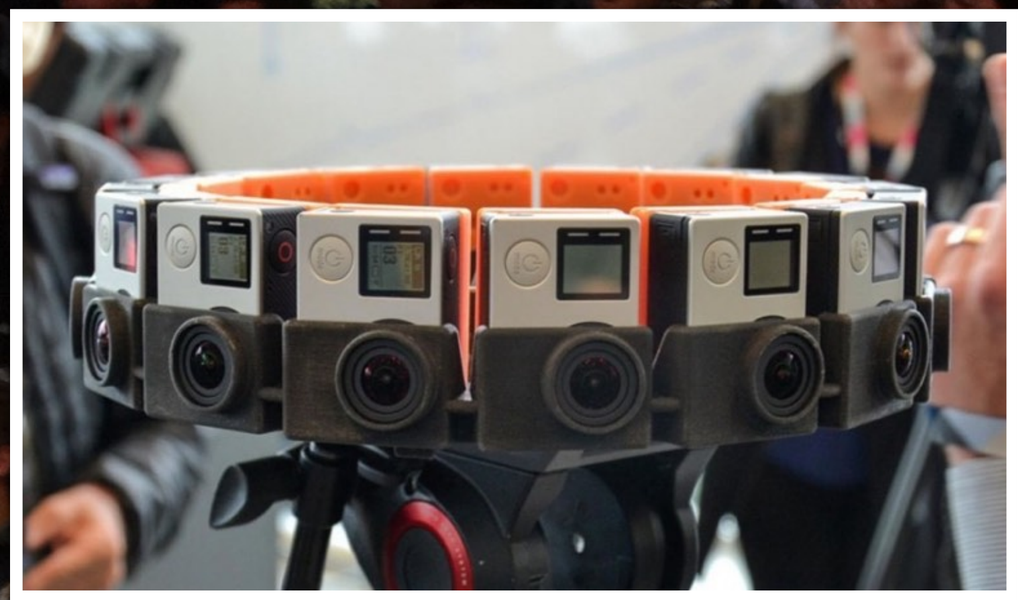
iPhone 6: 4.7 in “retina” display:
1.3 MPixel
326 ppi → **57 ppd**



Example: Foveated rendering (adaptive resolution based on tracked eye position)

Future “retina” VR display:
57 ppd covering 180°
= 10K x 10K display per eye
= 200 MPixel

Eyes designed by SuperAtic LABS from the thenounproject.com



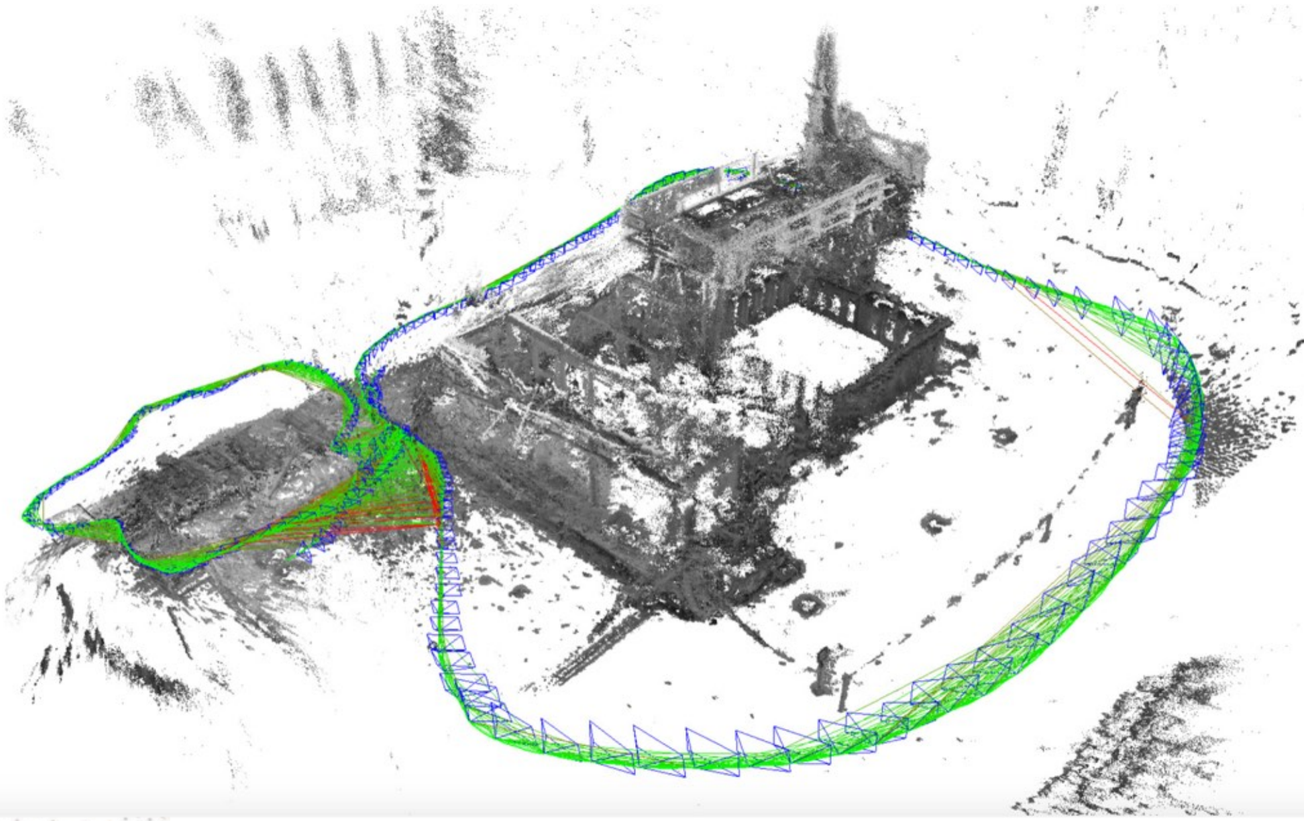
Example: Google's JumpVR video
Input stream: 16 4K GoPro cameras

Register/3D align video stream (on edge device)
Broadcast encoded video stream across the country to 50M viewers

3D reconstruction

3D reconstruction from RGB or RGBD images and video feeds

Keypoint-based approaches: [Snavely 06]



Visual odometry based approaches [LSD-SLAM 2014]

1. For each image, find interest points (map)
2. Correspondence: for each interest point in image A, find instances of similar points in other images (sparse all-to-all)
3. Iterative global optimization for 3D positions of cameras and interest points based on correspondences (iterative graph)

3D reconstruction (for localization)

Augmented reality (localizing the head)

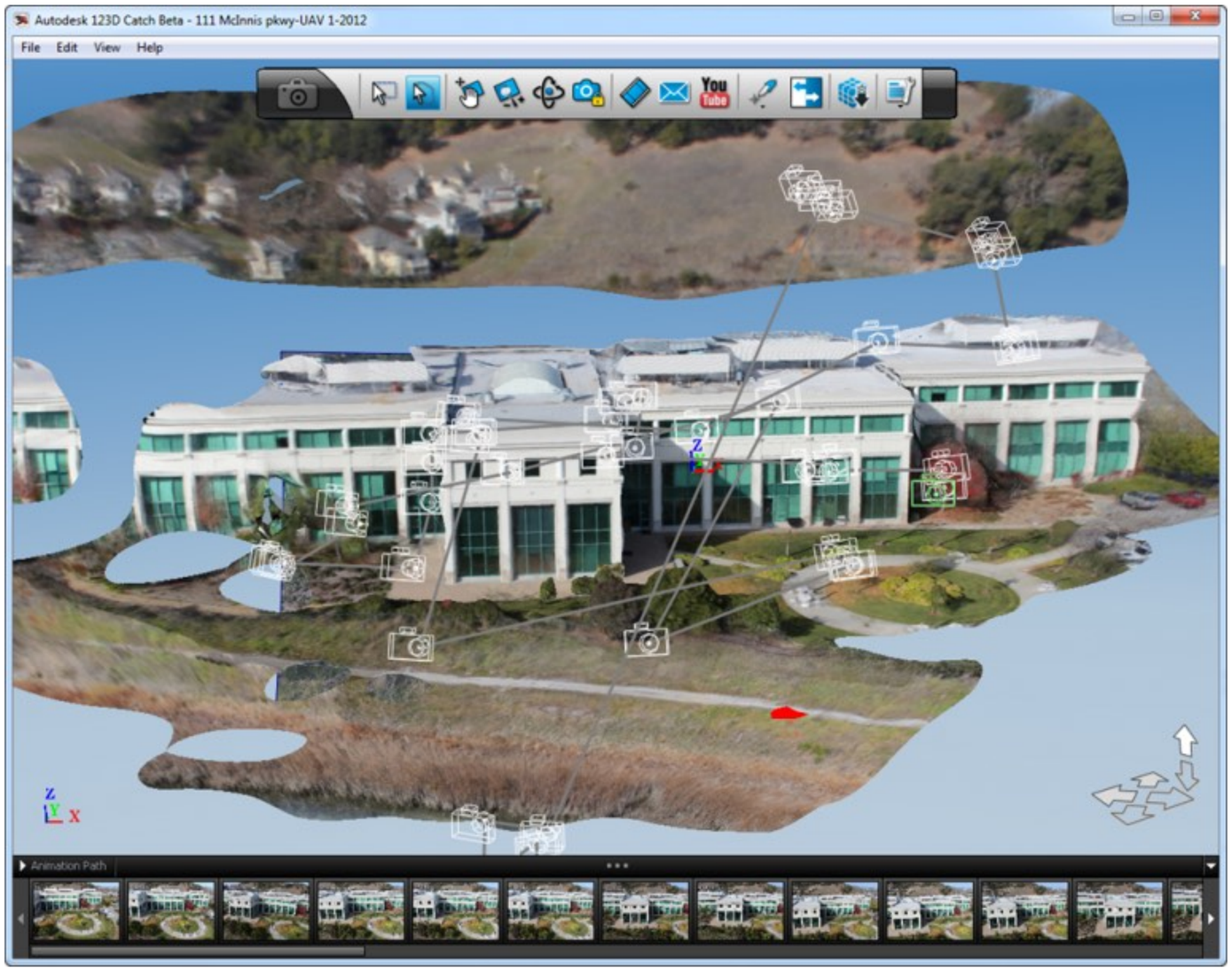
[Microsoft Hololens]

- AR requires detailed 3D map of surroundings:
  - To localize head motion of human (AR has no luxury of staying in one place like VR — must localize “inside out” using what headset sees)
  - Must know geometry to know how to render “on top” of real world
  - **Efforts to build cm-scale “3D map” of real world, used by AR apps**

3D reconstruction (for localization)

Autonomous vehicles

Flying drones on precise, repeatable paths



[Autodesk Octocopter]

Huge commercial interest in drone photography (e.g. construction site inspection)

Cloud-based gaming

Cloud-based gaming

- **Industry settling on streaming compressed pixels to thin client.**

  (not factoring computation between client and server)
- Low-latency requirement suggests need for GPU capability in future CDN architectures (to execute game + generate video streams)

Thin client (Android device)

user input

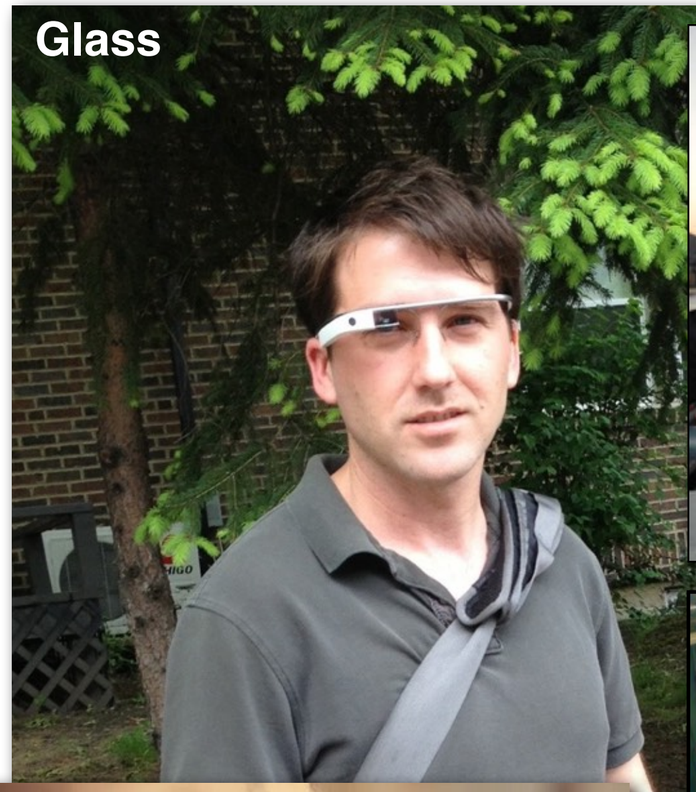
H.264-compressed HD video stream

Virtualized GPUs in datacenter (or home)

- Example: NVIDIA Tesla GRID GPU:
  - Maximum 8-to-1 user to GPU ratio
  - **Reduce latency + offload compression from CPU:** GPU HW H.264 encoder directly encodes frame buffer into H.264 bitstream, then GPU DMAs bit stream to host DRAM for NIC

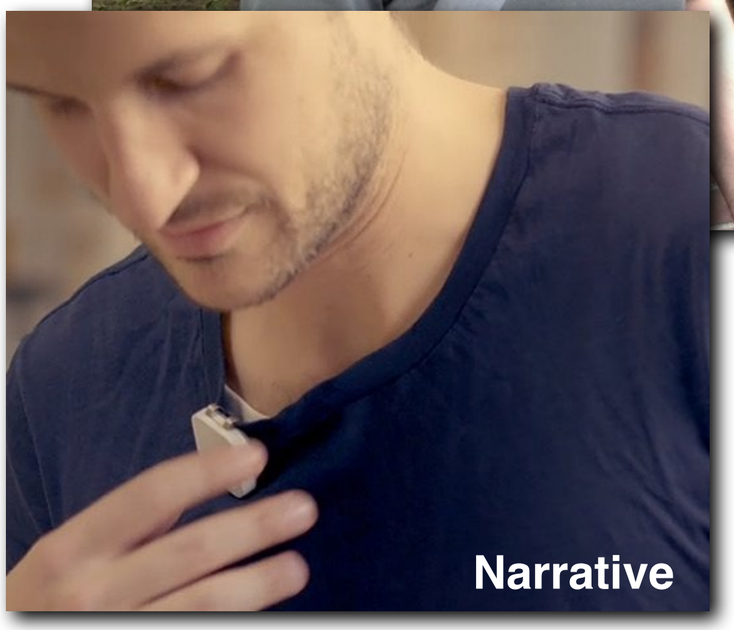
“Always-on” video stream analysis

Glass



MyLifeBits

Webcams



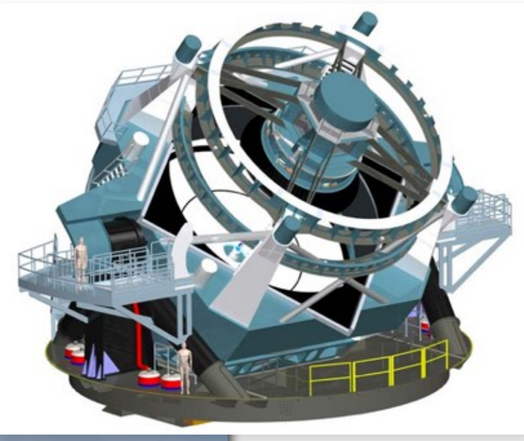
Narrative

Facebook “Live”

Traffic

Vehicles

Astronomy/Science



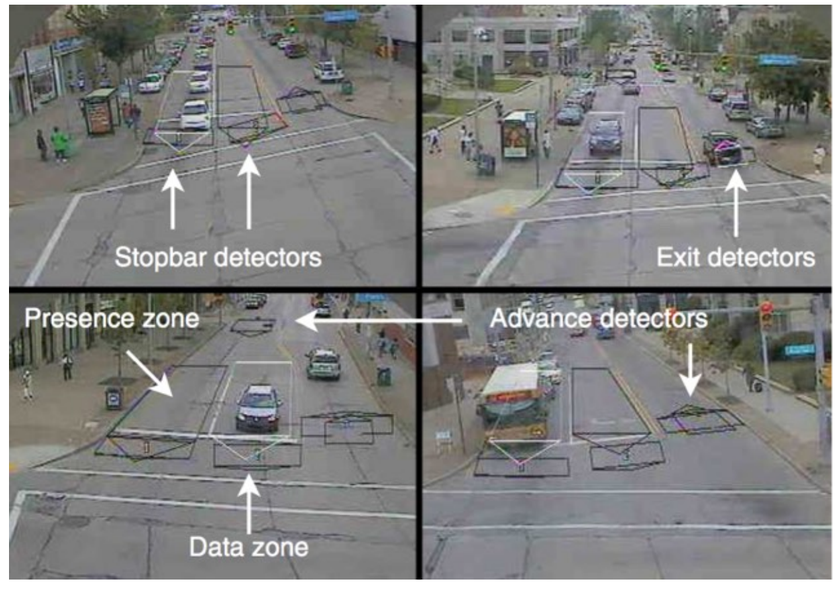
Surveillance

“Always-on” video stream analysis

- Example: GE Streetlight deployment
  - HD video cameras street light fixtures
  - Fiber runs from pole back to GE Predix datacenter in San Ramone.

- Platform Pittsburgh
  - Trying to build visual data ingest pipeline from 50 traffic cameras in Pittsburgh to a common, open data analysis platform at CMU.
  - Definitely an edge + cloud problem

Empty parking spot detection algorithm [Cisco]



SurTrac distributed traffic light control (9-intersection test in Pittsburgh)

Large-scale visual data analytics

- There is no doubt big-data analytics is central to many organizations today.
- There is comparatively little analysis done on the world's growing repository of visual information ("dark matter" of the internet) [Perona]

Enter Krishna...

Glass

GoPro

KrishnaCam dataset:

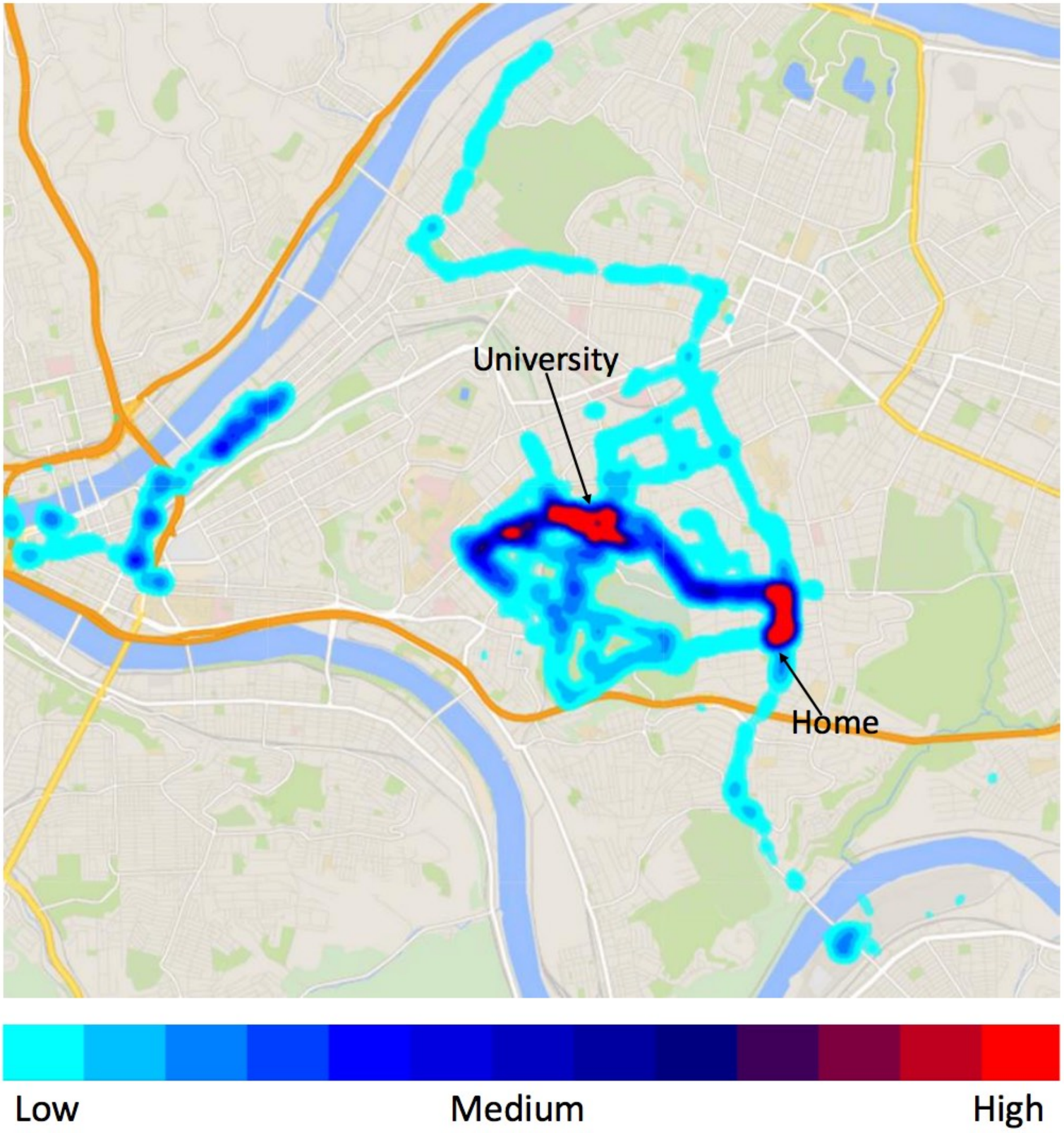
Recording by single individual:
Sep 2014 – May 2015

Duration: ~ 70 hrs
(5-30 minute clips of outdoor activities)

High dataset diversity:
urban, residential, campus, parks, day/night, seasonal change, interaction with friends

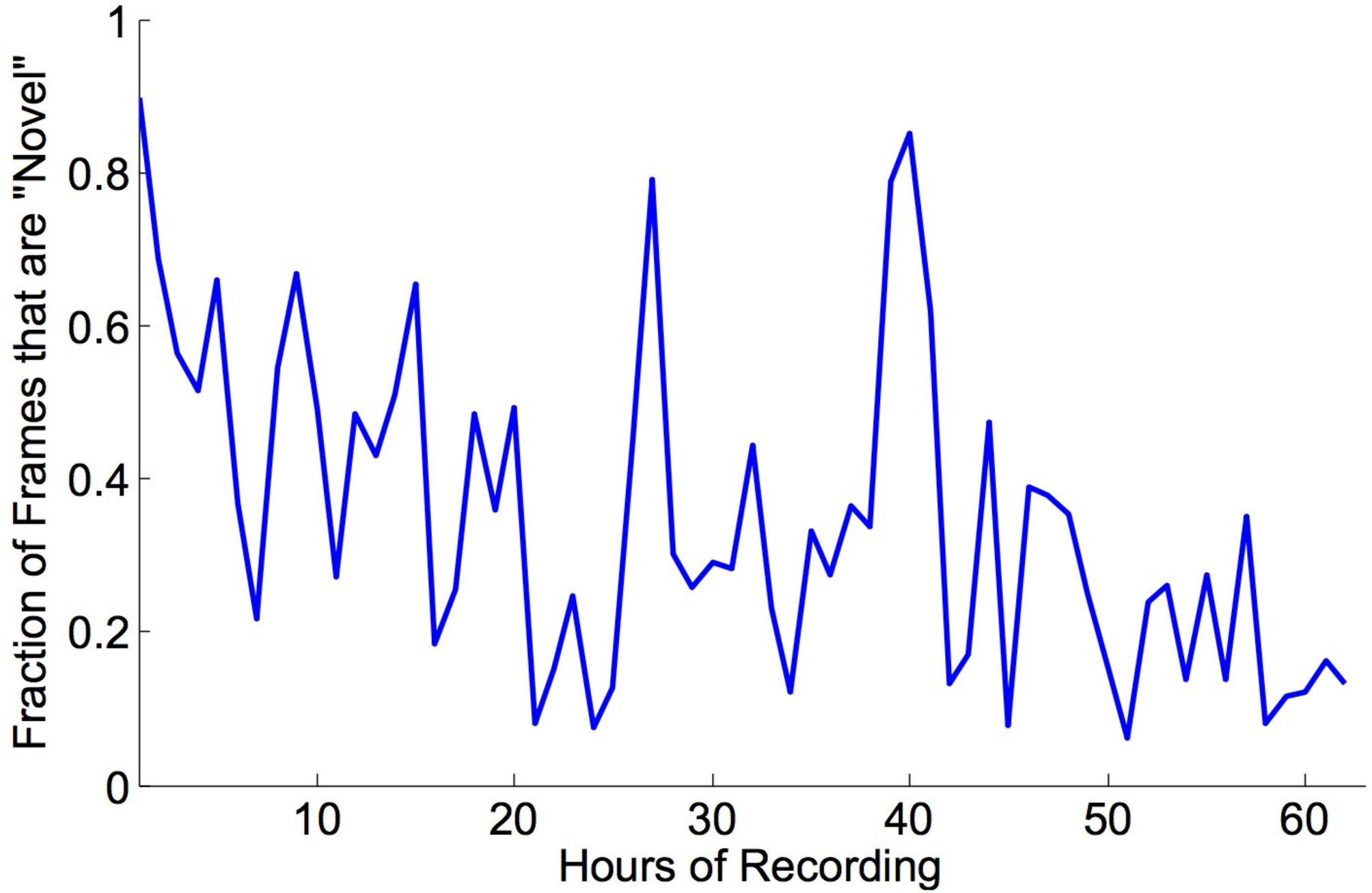
Data:
720p, 30 fps video
+ accelerometer, gyroscope, orientation, GPS on body (not camera)

Geographic location



How boring is the life of a grad student?

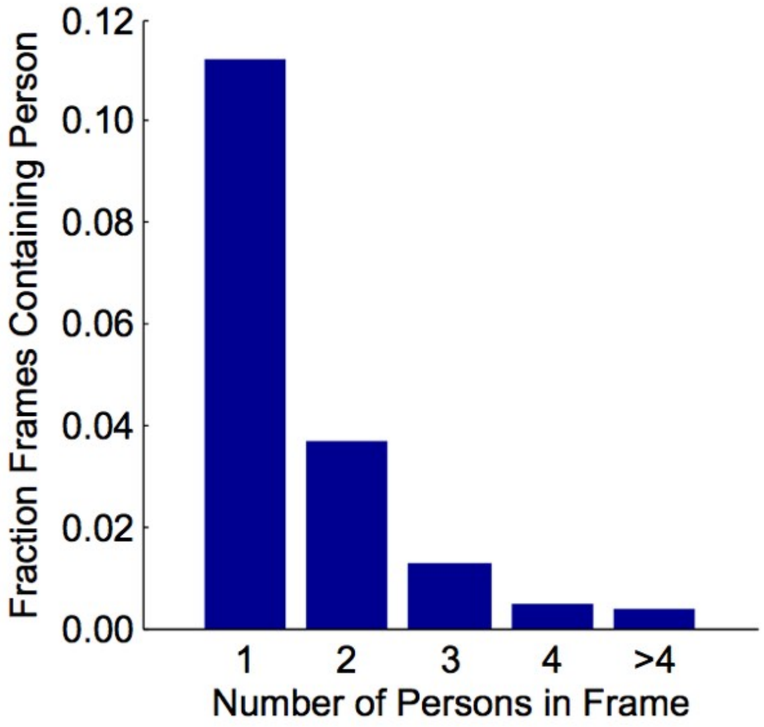
How much new visual data is seen as recording continues for months?



Similarity = cos distance of MIT Places layer 5 responses (full scene)

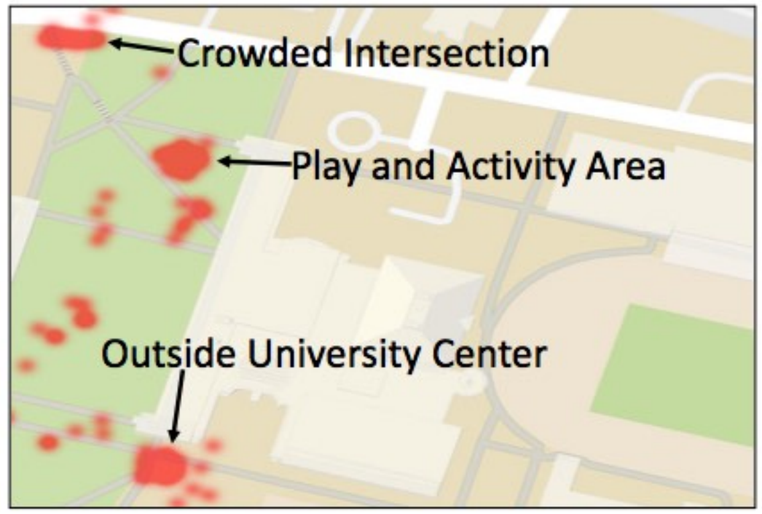
“Novel frames” = average distance to top-5 nearest neighbors greater than threshold

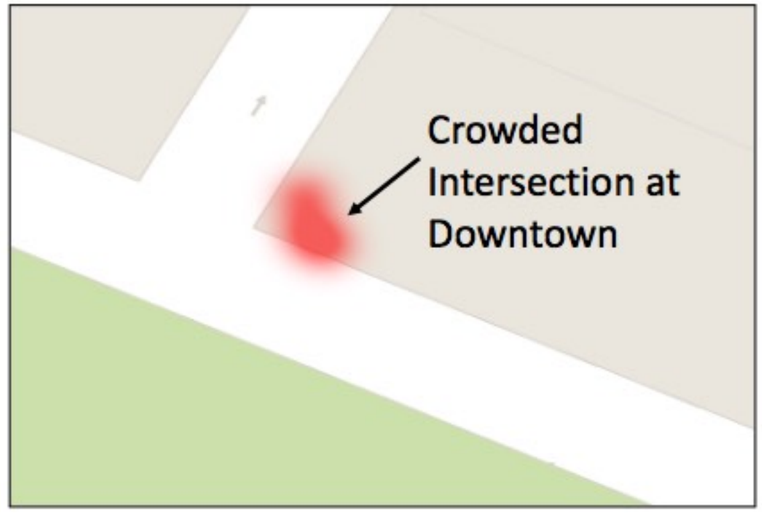
Where does Krishna see people?

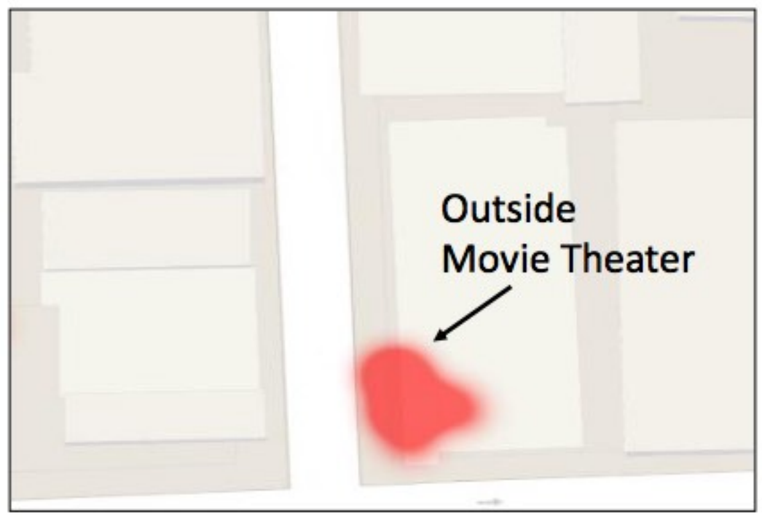


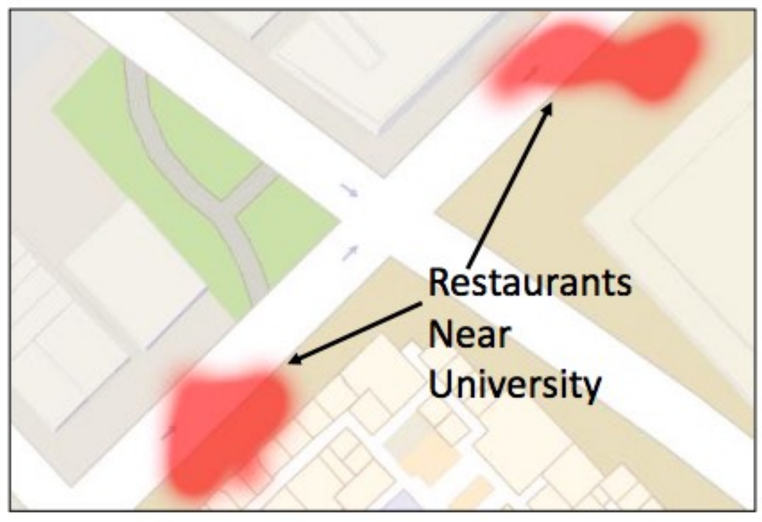
17% of frames contain at least one person (11% have exactly 1)

Highlighted areas: at least 4 people in frame on average (crowded areas around campus)

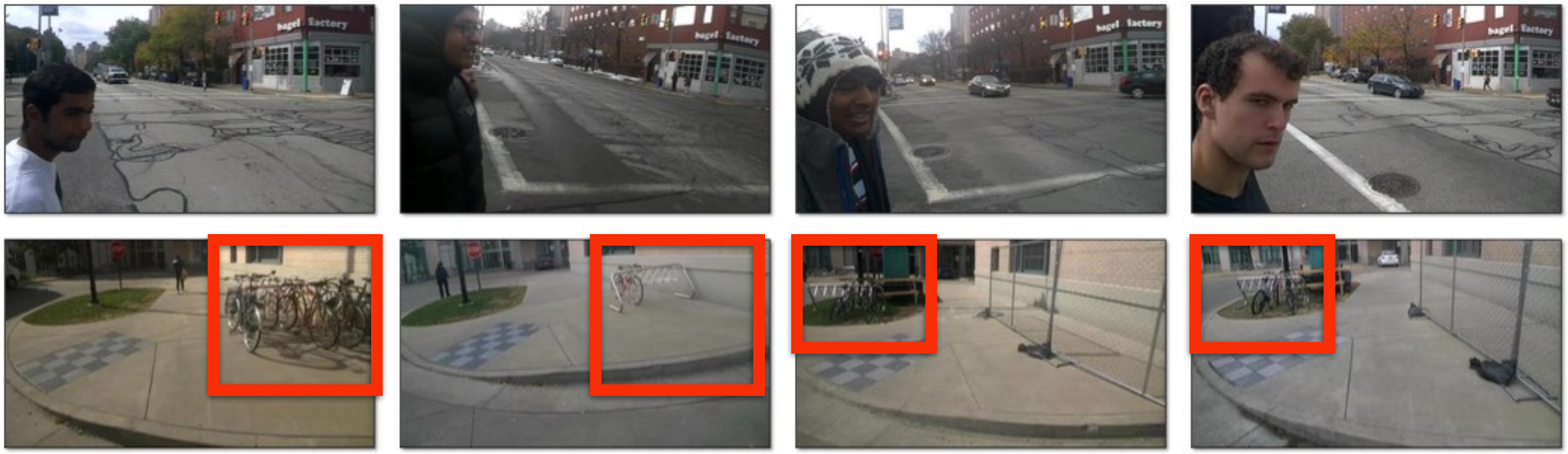








How does the world evolve?



1. Change in companion

2. Change in object location (bike rack moved for construction)

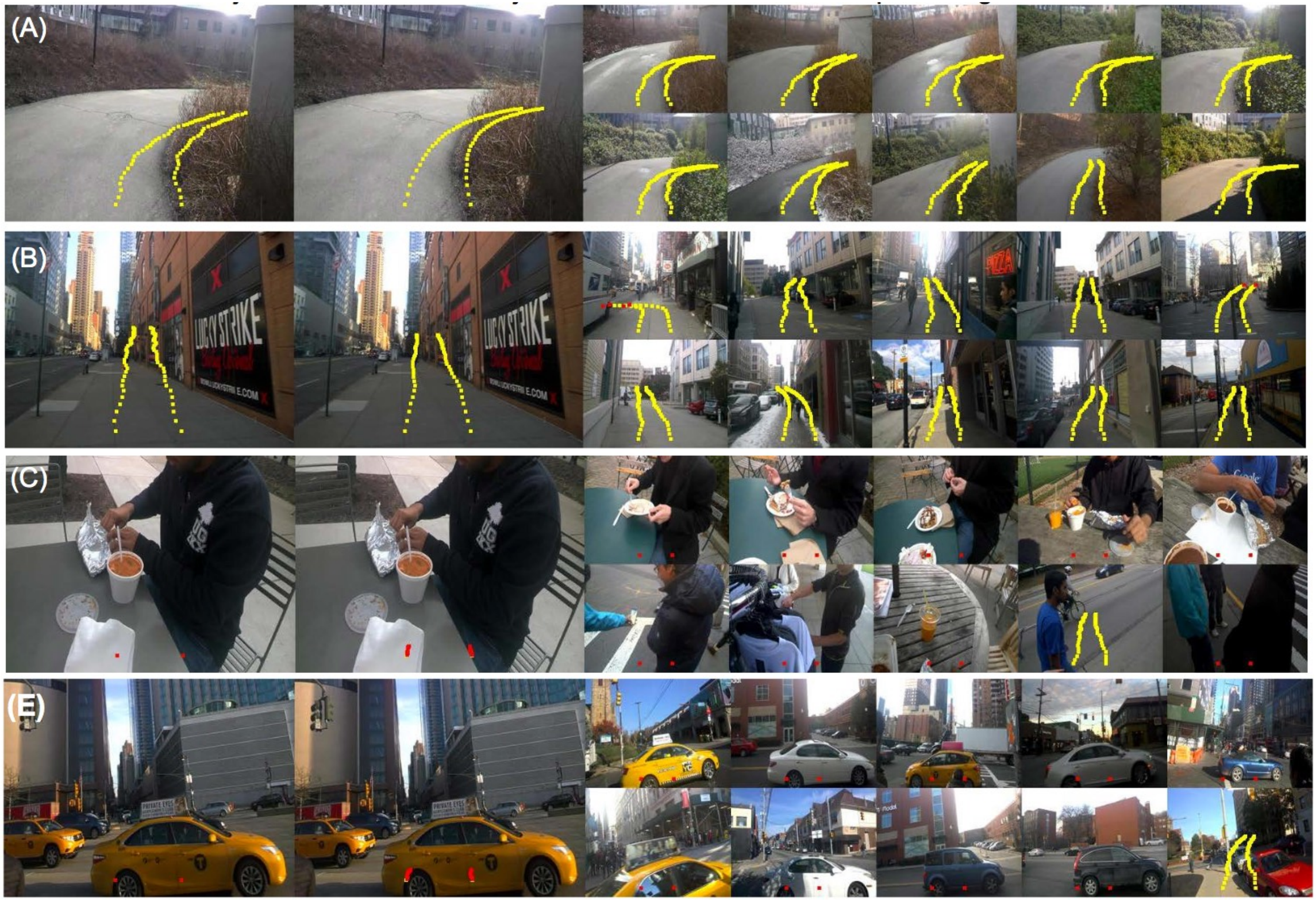
3. Change in transient object (different parked cars)

4. Change in season

5. Change in time of day (lighting conditions)

Predicting where Krishna will move next

Ground Truth | Predicted | Top 10 Nearest Neighbors from prior recording



Cross-Cutting Technologies

Portable, productive programming frameworks for high-performance image processing

- Halide language has been a big success: [Ragan-Kelley 2012]
  - Used to implement Google Photos Autoenhance, HDR+ app
- Halide = two domain-specific co-languages
  1. A purely functional DSL for defining image processing algorithms
  2. A DSL for defining "schedules" for how to map these algorithms to machines

```
Func halide_blur(Func in) {
 Func tmp, blurred;
 Var x, y, xi, yi;

 // The algorithm
 tmp(x, y) = (in(x-1, y) + in(x, y) + in(x+1, y))/3;
 blurred(x, y) = (tmp(x, y-1) + tmp(x, y) + tmp(x, y+1))/3;

 // The schedule
 blurred.tile(x, y, xi, yi, 256, 32)
        .vectorize(xi, 8).parallel(y);
 tmp.chunk(x).vectorize(x, 8);

 return blurred;
}
```

Algorithm is a series of functions (think: pipeline stages)
Side-effect-free functions map coordinates to image values
(in, tmp and blurred are functions)

Schedule describes how to map functional specification to a parallel machine

Visual computing database: What are the right representations for efficiently querying and analyzing large image/video collections?

1. We are exploring scheduling of relational, spatial, temporal queries that involve significant, heavyweight pixel manipulation
2. Result sets are subsequently used in super-computing scale processing: e.g., 3D reconstruction, alignment, optimization/training

- select (all frames on Youtube
  ten seconds BEFORE
  a frame containing a child crying)
  { then run myClassifierTrainingFunction() }

- bikes = select (all frames on Uber dashboard cams in Dec 2014 AND biker falling in the picture)
  cars = select (all frames on Uber dashboard cams in Dec 2014 AND car in the picture)
  select frames where bbox(bikes) is WITHIN 100 pixels of bbox(cars)) { … }

- select (all frames in my_image_database containing a person in a specified pixel region)
  { then scale and align all images according to bbox(person) and compute average }

Writing applications that share common sensing infrastructure at the edge

- Edge + cloud application may share edge sensing resources with other applications (in addition to sharing traditional cloud computing/storage resources)

- Think: how would a city virtualize its smart sensing infrastructure so third-parties could write applications that extract value from it?
  - Little work in “smart camera” platforms in this direction

Managing accelerated-computing in the datacenter

- To meet efficiency goals, datacenter-scale visual computing applications will seek to leverage throughput architectures: GPUs or other accelerators
  - Evaluate sufficiency of current GPU architectural support for virtualization? (reducing cost of context switch)

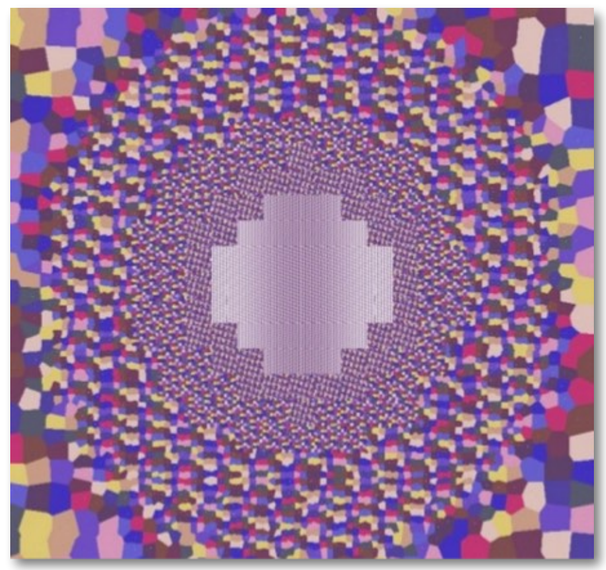
- What fixed-function logic should be added to server CPUs
  - e.g., tight integration between HW video compressor and NIC

- Current debate in graphics community: is there sufficient motivation for including a programmable image processor (ISP) in a modern system? (node = CPU+GPU+image processor)

Visual data compression

- For distributing high-resolution streams (e.g., VR video) to an array of edge devices

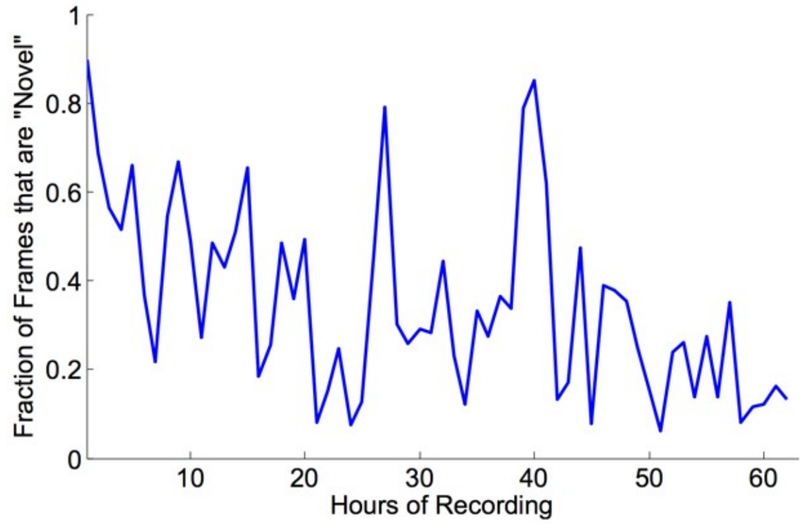
- To enable flexibility to retarget dynamically specific output devices



Example:
considering foveation

- For intelligently ingesting data from always-on sources
  - Edge computation serves as an intelligent filter/compressor (need to establish richer notions of visual data importance)

- Difference from traditional image/video compression: most data will be consumed by computers, not humans



Example: exploiting redundancy over long time scales

Algorithms for indexing visual data

- Retrieval and correspondence are search problems that are at the heart of many applications
  - Find this pattern in the same image
  - Find this pattern somewhere in any image the database

- Indexing visual data remains an open problem

- Sometimes the best way to search/analyze efficiently is to throw out most data (subsampling, aggregation, compression)

Computing on big visual data, while preserving privacy

- Visual information may lend itself to new forms of anonymization techniques
- Technologies for maintaining data provence / limiting data lifetime may become critical
  - Cameras everywhere may be more palatable if the data is not stored for long

Summary

This is going to be fun

- Next generation of visual computing workloads
  - 3D graphics community has long history of embracing domain-specific languages and specialized throughput architectures to achieve exceptionally efficient solutions (**"Use every cycle we can get"**)
  - This community is excitedly tackling emerging problems in image and video analysis and 3D reasoning (**"gosh, there's really high-impact new problems and low hanging fruit in systems-design here"**)
- Problems are moving into internet-scale regimes where we could learn a lot from the distributed system community.

Thank you

kayvonf@cs.cmu.edu