High Speed Object Tracking Using Edge Computing

Caleb Phillips, Seyed Hossein Mortazavi, Pan Zhang, Joel Dick, Eyal de Lara
Department of Computer Science, University of Toronto

UNIVERSITY OF TORONTO

Problem

- Golfers lose golf balls to slices and rough terrain.
- Affordable drones with onboard cameras can be used to take images of the ball but lack enough computational power to perform high speed tracking.

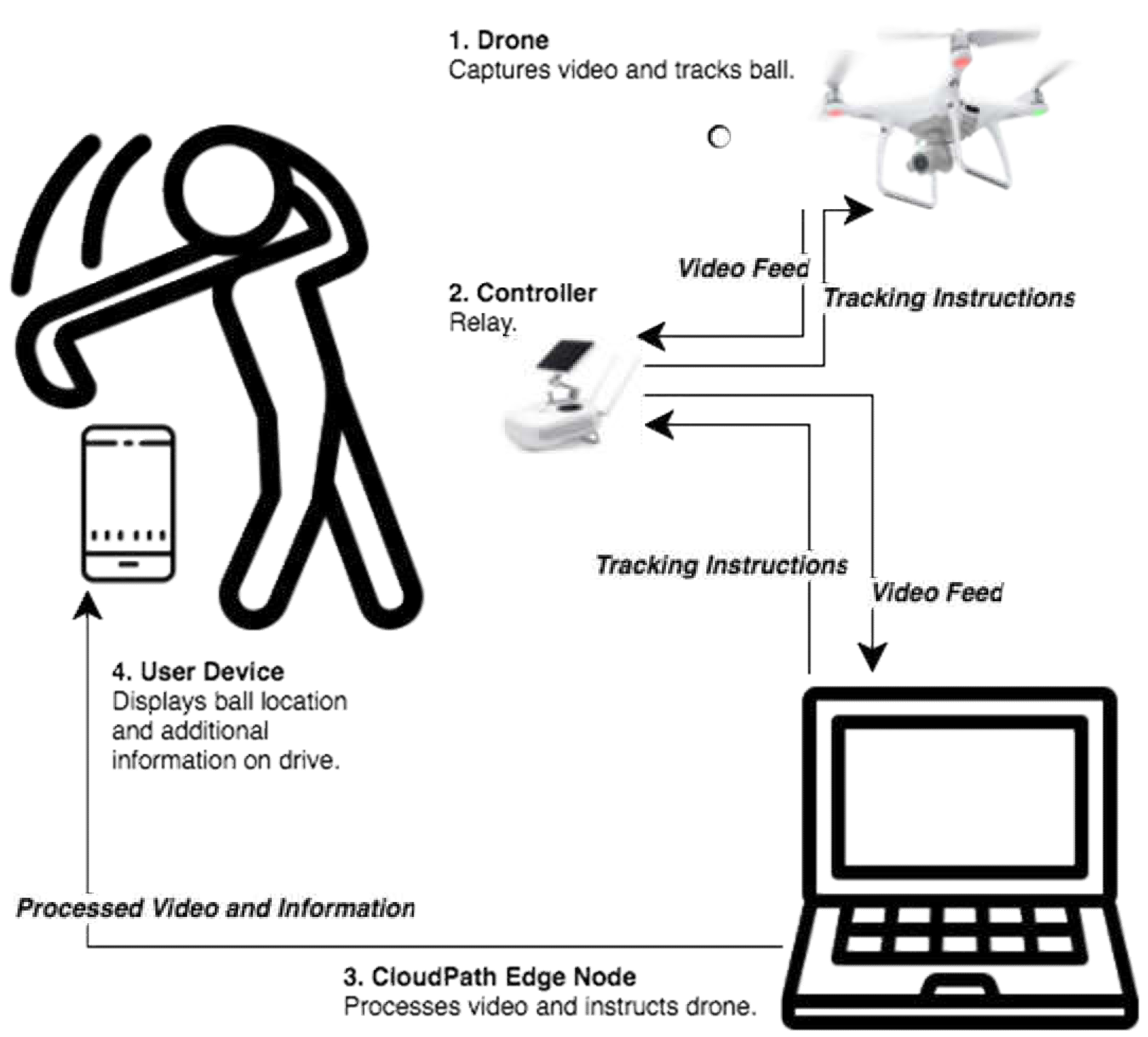
Offloading Drone Computation to the Edge

- Insufficient computation make onboard calculations impractical.
- Low Latency requirements make cloud computation impractical.

Our Solution

- Use nearby edge node to offload computation.
- Has sufficient computational ability and low latency.

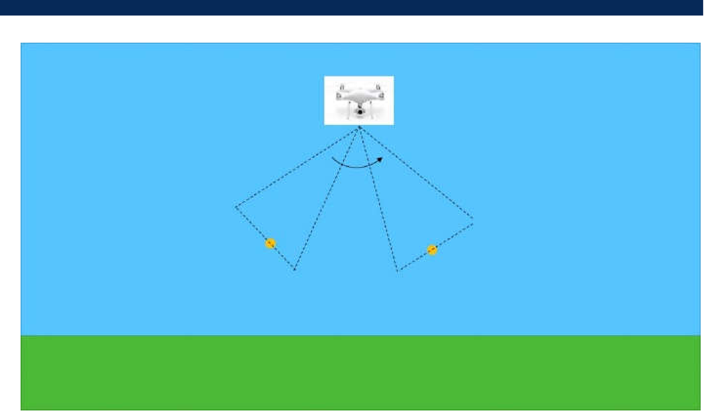
Architecture



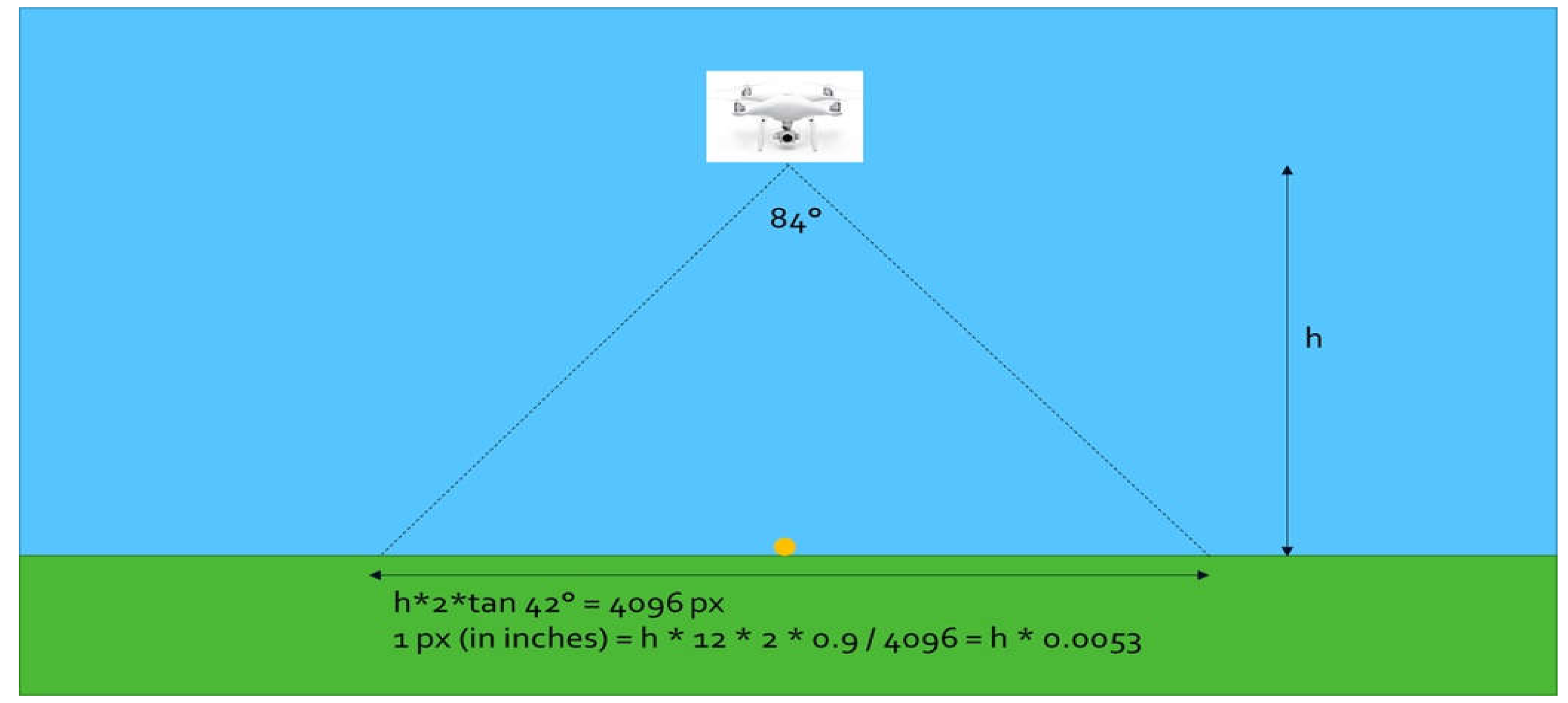
Approaches

1) **Image Tracking** – Pivot and track the ball directly for full length of shot. Decided infeasible due to large distances and limitations in tracking.

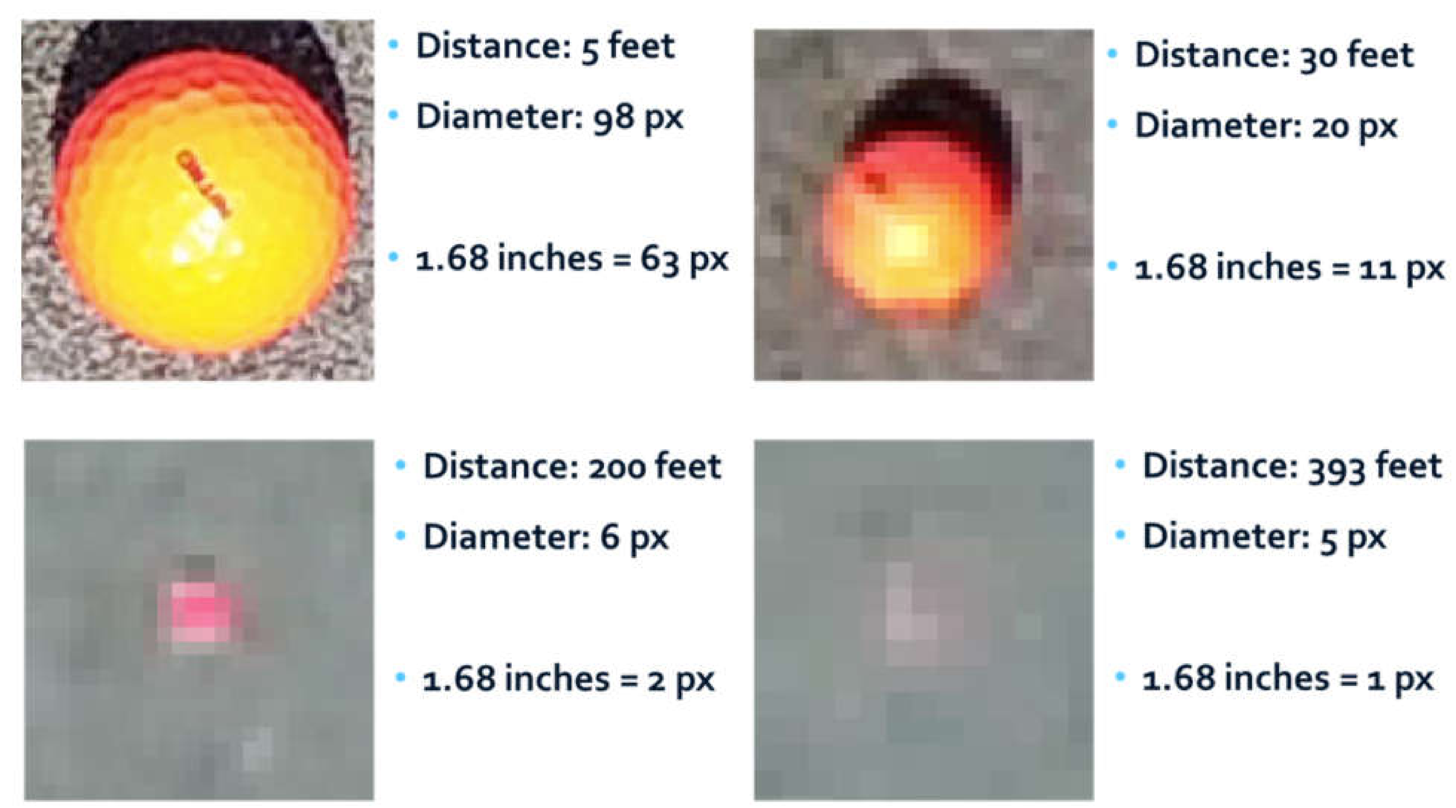
2) **Model Development** – Track ball at start of shot and infer physics model. Track ball at end for precise location. The current approach.



Prediction Using Tracking

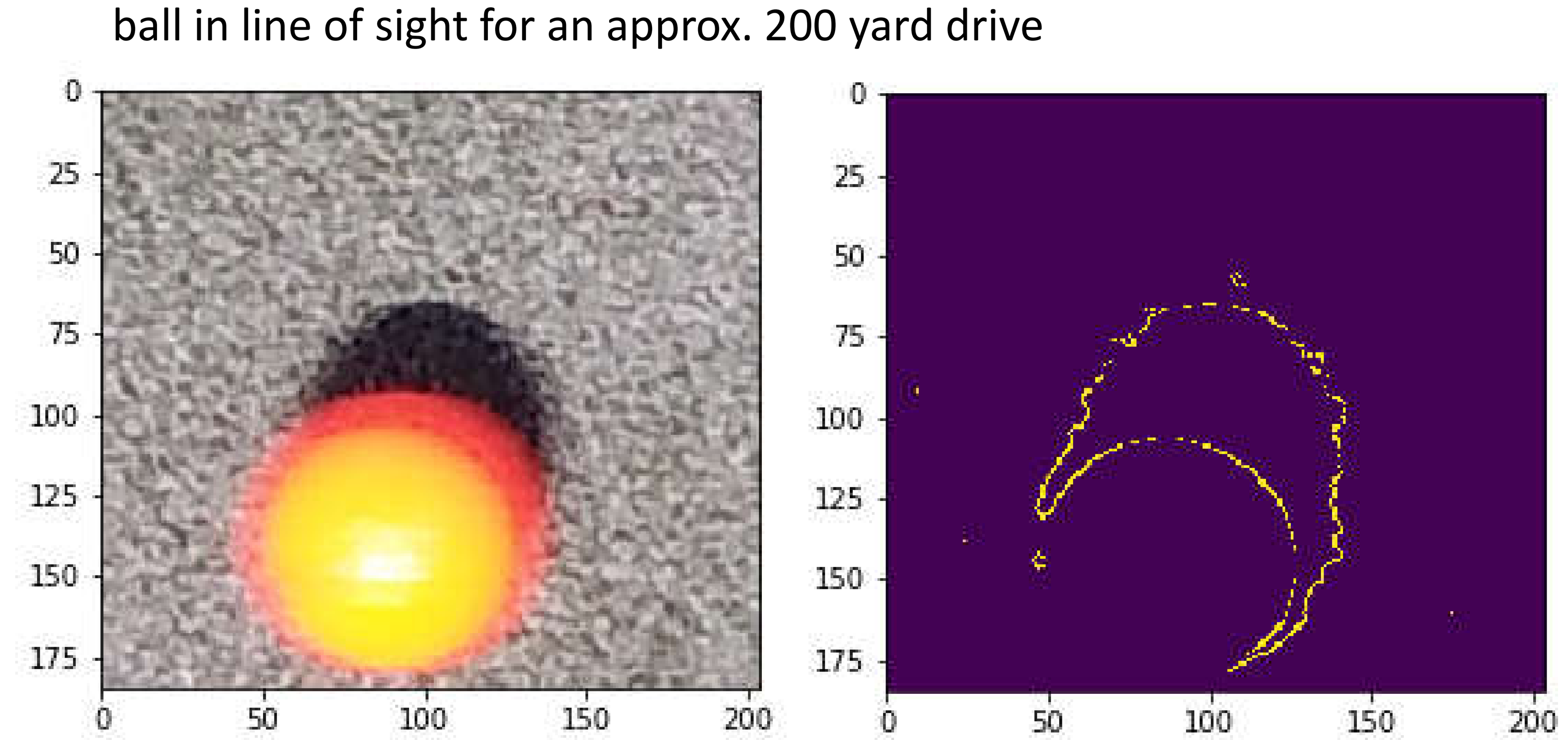


- Pixel density scales with linearly distance
- Tracking algorithms incredibly difficult at higher ranges



Prediction Using Model Development

- From top down perspective, use 2 images to predict ball trajectory.
- Use known width of the ball and pixel width to predict height.
- Develop physics based model to predict landing point and turn to view.
- Takes 1.25 seconds to turn the drone 75 degrees, far enough to keep ball in line of sight for an approx. 200 yard drive

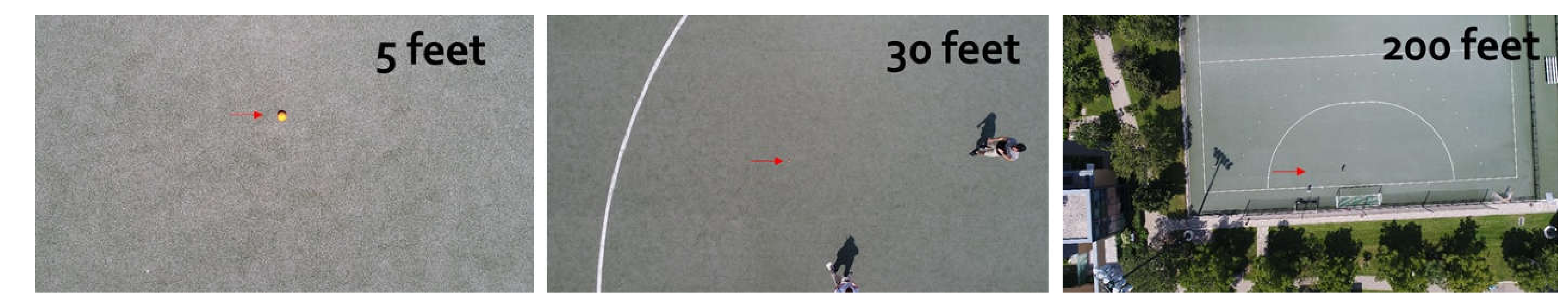


- Edge detection used to estimate ball distance from drone.
- Ball height can be inferred because height of drone is known.

Why is it difficult?

- Ball can be reasonably tracked by the drone within ~**30 feet.**
- At this distance, a slower drive will stay in frame for **500 milliseconds**.
- Latency to send video from the drone to controller is **220 milliseconds**.
- Additional latency if the controller is not directly connected to the edge.
- Additional latency to return instructions from the edge to the drone.
- This leaves us with ~**100 milliseconds** of computation time to decide where to turn the drone gimbal.

Experiments



- Tracking ball from various heights.
- Tracking with OpenCV becomes error prone at heights above 30 feet.

- Indoor dome with measured land marks
- Drone captures the first few frames and also records the places the ball lands (required for verification)

Future work and other applications

- Ideally the difficulty of the problem will allow us to use our methods in other domains.
- E.g. Offloading Computation for IoT devices performing; Tracking of Vehicles, Facial recognition, search and rescue, sports monitoring etc.