WiseCam: Wisely Tuning Wireless Pan-Tilt Cameras for Cost-Effective Moving Object Tracking

Jinlong E, Lin He, Zhenhua Li, and Yunhao Liu

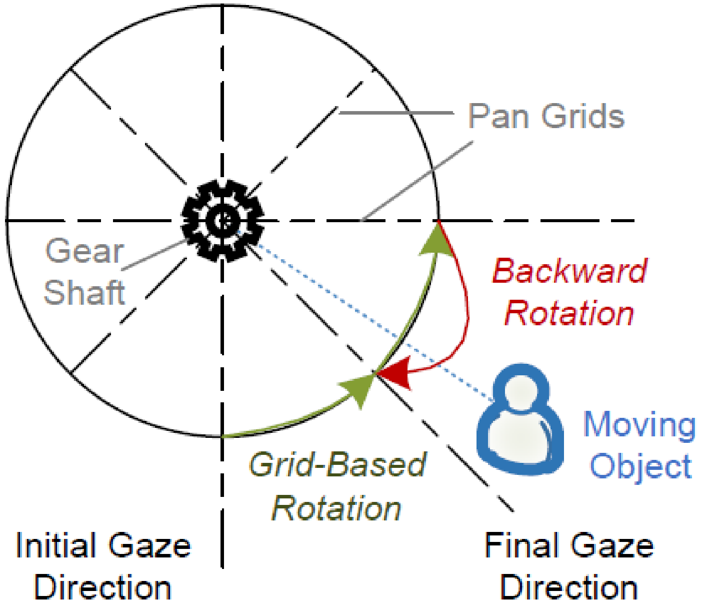
Renmin University of China & Tsinghua University

Background

- *Wireless Pan-Tilt Cameras*
  - Professional surveillance instruments (global market value > $3 billion)
  - Automatic directional control -> in replace of multiple fixed cameras
- *Moving Object Tracking*
  - Indoor *DC-supplied* scenarios: elderly/child caring, anti-theft alarming
  - Rural *off-the-grid* environments: farmlands, orchards, and fisheries *(potentially conducive to but yet to be deployed)*
  - Requirement: *long-term* & *energy-efficient* tracking

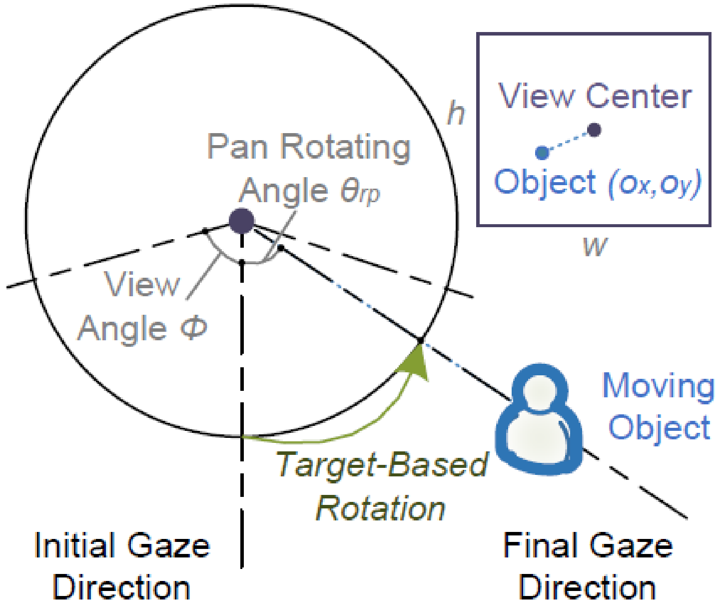
State-of-the-art Approaches

- *Grid-Based Tracking*
  - Applied by pan-tilt cameras with *stepper motors*
  - Moving object detection in each video frame *(by detection algorithms like TFD or BS)*
  - Searching for *pan-tilt grids* -> the detected object closest to *gaze direction*
  - Time-consuming comparisons and frequent acceleration/deceleration among grids
  - May trigger *backward rotation*



State-of-the-art Approaches

- *Target-Based Tracking*
  - Applied by pan-tilt cameras with *servo motors* *(assembled with fixed camera and tripod head)*
  - Similar object detection as *grid-based tracking*
  - Calculating pan-tilt *rotating angles* based on *object center*'s coordinates to *view center*
  - Higher rotation speed and position precision
  - Sensitive to small object motions -> excessive *stacked rotations*



State-of-the-art Approaches

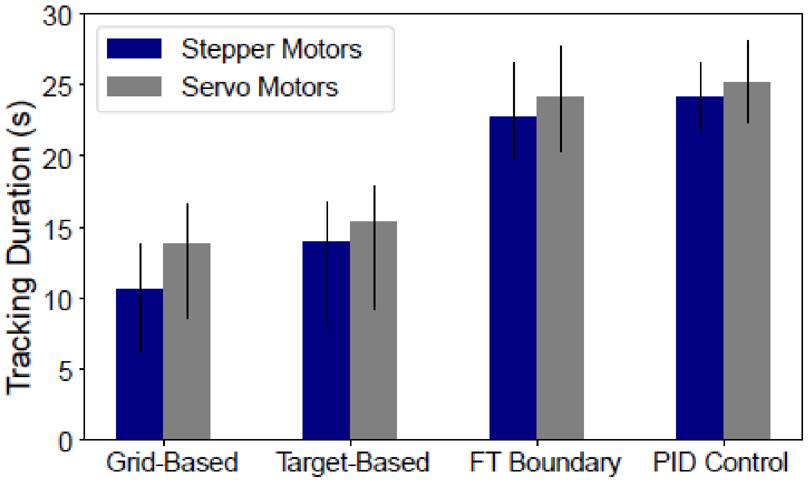
- *Fault-Tolerant Boundary*
  - Improved approach against continuous violent rotations
  - Object center within a boundary around *view center* -> no rotations
  - *Boundary size* is subject to object size, speed, and direction
- *PID Control*
  - Alternative approach to reducing frequency of rotation generation
  - Iteratively obtaining an *error* between object center and *view center*
  - Calculating *rotating angles* based on a weighted sum of the *error*'s proportional, integral, and derivative (PID)

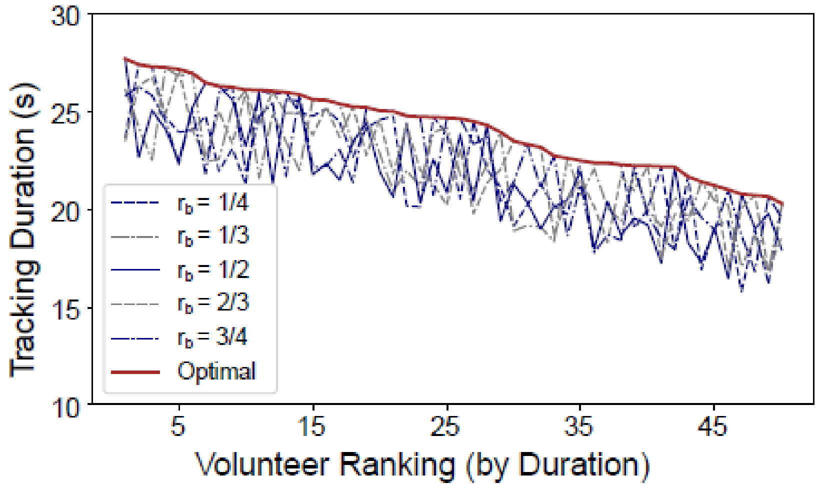
Measurements

- *Metrics*
  - *Tracking duration*: time that the camera can keep sight of an object
  - *Rotational power consumption*: power increment incurred by rotations
- *Methodology*
  - Running 4 *state-of-the-art* approaches on *stepper/servo-driven* camera
  - Tracking 50 volunteers with different body sizes and moving speeds
  - Each volunteer walks *stochastically* around a ~15m^2 room with two cameras hung in the middle of the ceiling in turn
  - Connecting *power meter* to measure the cameras' power consumption

Measurements

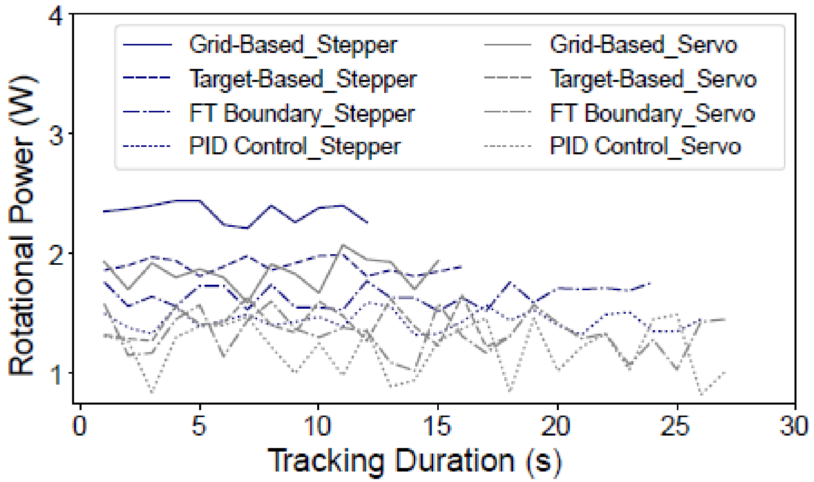
- *Observations on Tracking Duration*
  - Not ideally long with all approaches (shorter for *stepper-driven* camera)
  - Sensitivity of *target-based tracking* leads to *asynchronous* rotations
  - Optimal *boundary size* is inconsistent for various volunteers
  - Tuning of *PID coefficients* is complex and costly in practice





Measurements

- *Observations on Rotational Power Consumption*
  - *Continuously considerable* power consumption to both cameras
  - *Stepper-driven* -> always high vs. *Servo-driven* -> highly variable
  - *Energy cost* (*integral* of power consumption) is overall high
  - Incurring *power cuts* given the typical solar battery's *energy budget*



Tracking Approaches	Stepper-Driven	Servo-Driven
Grid-Based Tracking	4.546	3.641
Target-Based Tracking	4.095	3.512
With FT Boundary	3.846	3.228
With PID Control	3.639	3.090

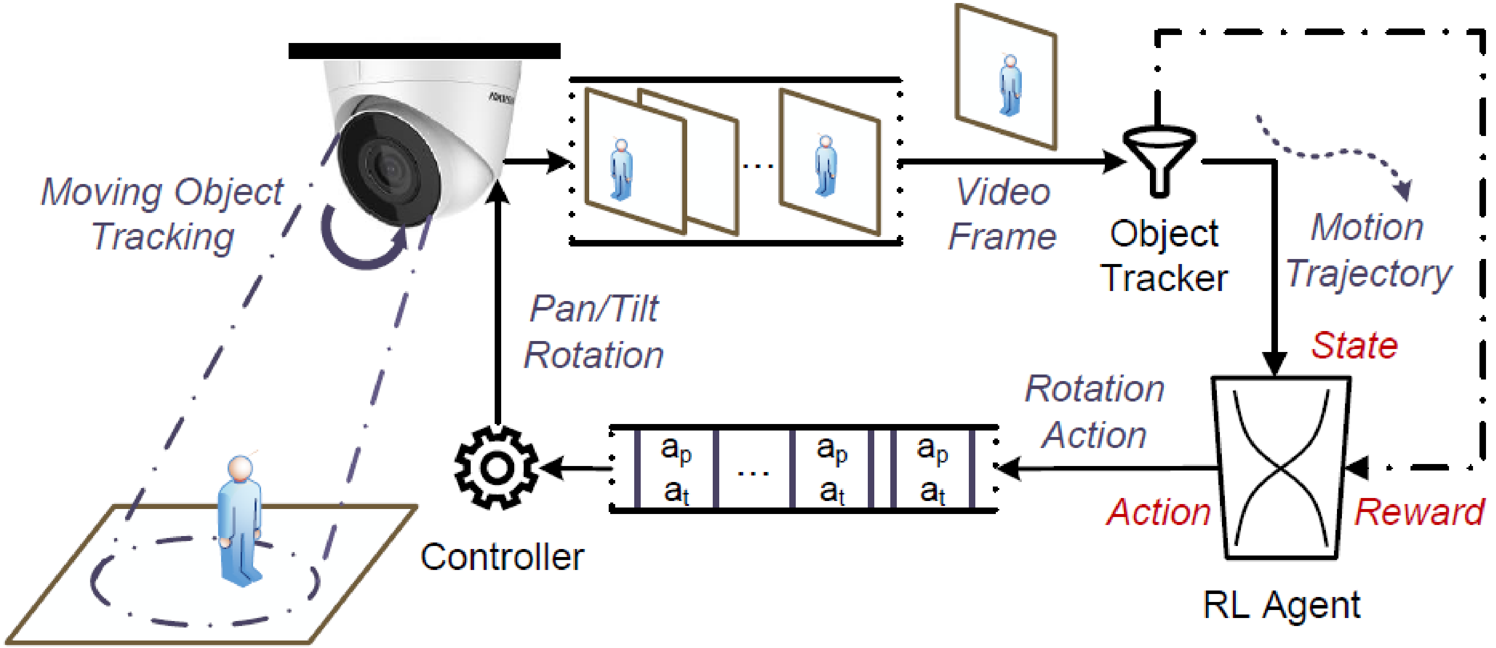
Average energy cost (Wh)

Motivation

- *Indications*
  - *State-of-the-art* approaches -> *inadequate* to provide *long-term* tracking & *inapplicable* to fulfilling *energy-constrained* scenarios
  - Performing gimbal rotations based only on *instant* object detection increases the risk of *tracking failure* and *energy costs*
  - Root cause: *stateless nature* derived from industry's *simplicity principle*
- *Opportunities*
  - Camera keeps *gazing* at the object instead of detecting it in each frame
  - Object state decides camera's *stay point* without dispensable rotations

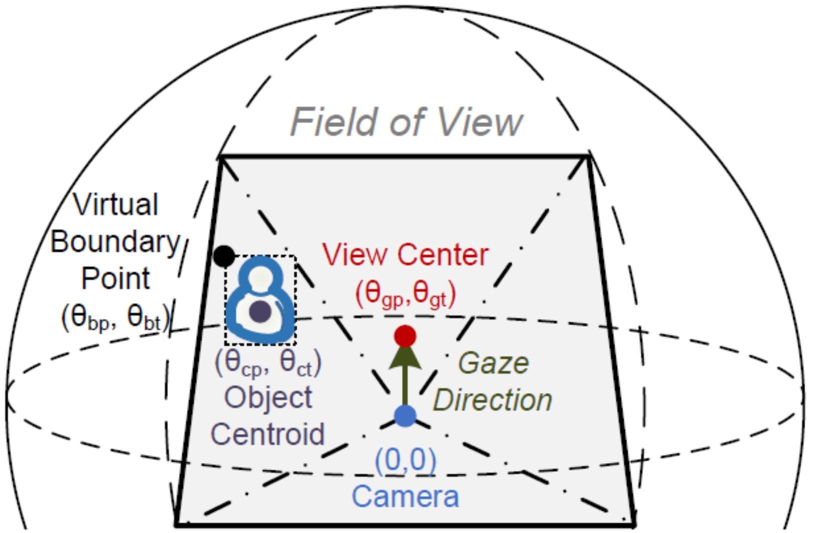
WiseCam Design

- *Overview*
  - Goal: to *minimize* rotation *costs* while maintaining *long-term* tracking
  - *Object Tracker* -> to keep a *close watch* on an object in a *unified* space
  - *RL Agent* -> to efficiently learn *object motion* for *online* determination



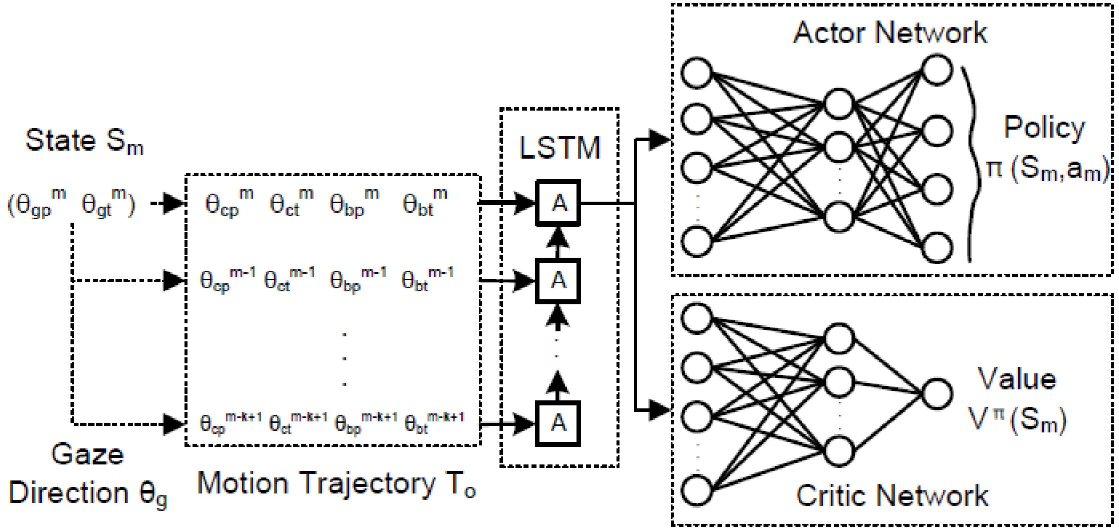
Long-Term Moving Object Tracking

- *Object Detection and Correlation*
  - Integrating *regional optical flow* with *temporal frame differencing (TFD)* for object detection -> to precisely obtain *complete object contour*
  - Leveraging *correlation filtering* for cross-frame object matching (with *position* & *scale* filters) -> to eliminate interference from other objects
- *Motion Trajectory Construction*
  - Abstracting *critical points* (*object centroid* & *virtual boundary point*) in each frame
  - Transforming their coordinates to a trajectory of *geodetic coordinates* in a *panoramic* space



Online Rotation Determination

- *RL Model Formulation*
  - *State*: camera's *gaze direction* and k quadruples of *motion trajectory*
  - *Action*: a pair of pan-tilt rotating angles within the rotation amplitudes
  - *Reward*: weighted sum of rewards on *position*, *direction*, *cost*, and *loss*
- *Model's internal structure*
  - *LSTM*: reasons *implicit features* hidden in the motion trajectory
  - *Actor NN*: outputs a rotation action
  - *Critic NN*: *judges* the action's *value*



Online Rotation Determination

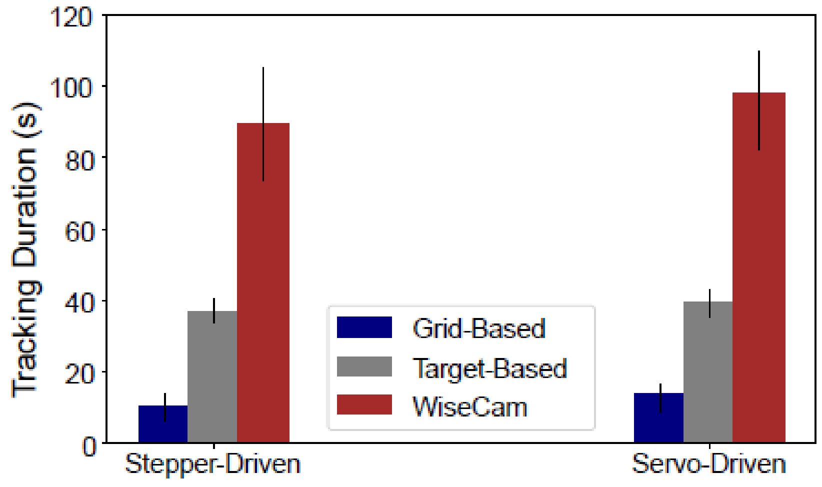
- *Fast-Convergent Training*
  - Model should *converge* soon to adapt with *online* determination
  - *Proximal policy optimization (PPO)* -> to update policy smoothly
  - To maximize *cumulative discounted reward* (with *advantage function*)
- *Multi-Model Aggregation*
  - *Quiescent period* (without camera rotations) facilitates initial training
  - Drawing on *experience* from *previous* object tracking of the same type
  - *Fusing* NN parameters of the same cell across previous objects' models

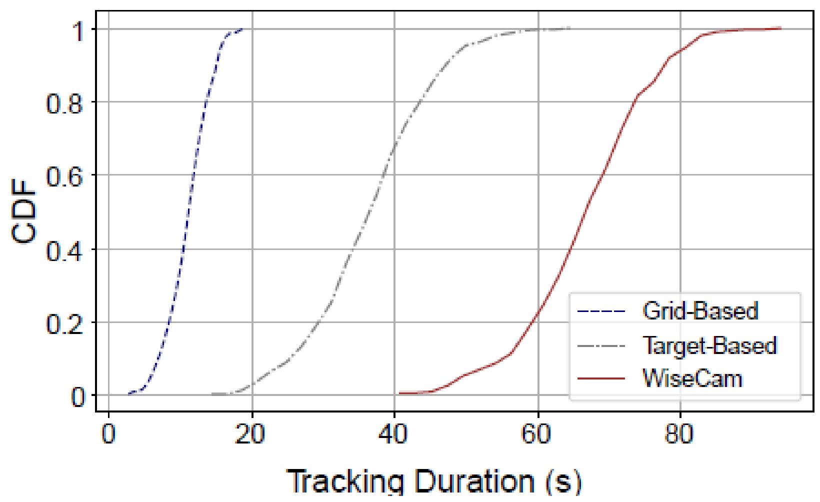
Evaluation

- *System Implementation*
  - Building a prototype on *Raspberry Pi 4B* with *OpenCV* & *TensorFlow*
  - Controlling *stepper-driven* cameras through WiFi and *ONVIF protocol*
  - Controlling *servo-driven* cameras through I2C bus and *PWM drivers*
- *Experiment Setup*
  - Metrics: *tracking duration* & *rotational power consumption* (*Ditto*)
  - Baselines: *grid-based tracking*, *improved target-based tracking* (with both *fault-tolerant boundary* and *PID control*)
  - Datasets: 50-volunteer tracking & large-scale daily walking trace

Evaluation

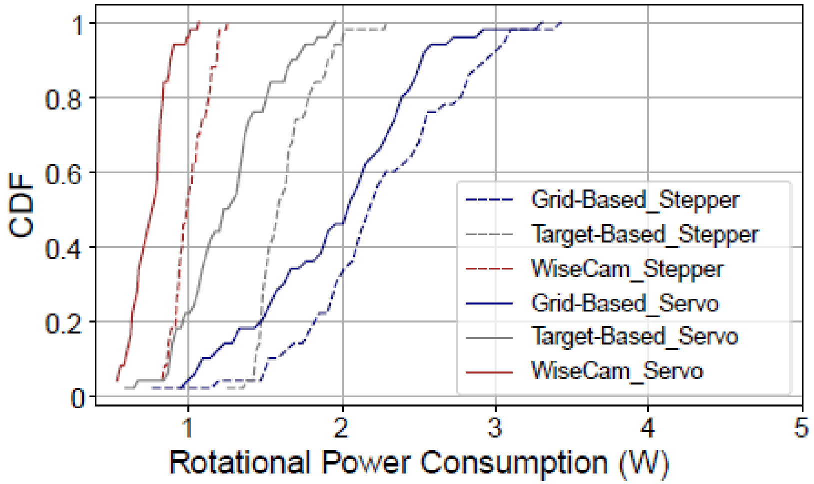
- *Improvement at Tracking Duration*
  - WiseCam keeps sight of 50 volunteers for *2~8× as long as* baselines *(due to its cross-frame object correlation and wise determination)*
  - For tracking emulation based on *large-scale* trajectory trace, WiseCam's median and tail durations are *much longer* than those of baselines





Evaluation

- *Reduction at Power Consumption*
  - Baselines exhibit an apparently longer tail, and reduction by WiseCam is *around 40%* *(due to its ability to refrain from dispensable rotations)*
  - WiseCam can reduce the *servo-driven* camera's energy cost to well support its *daily* object tracking powered by a typical *solar battery*



Tracking Approaches	Stepper-Driven	Servo-Driven
Grid-Based Tracking	4.546	3.641
Target-Based Tracking *(Improved)*	3.535	2.894
WiseCam	3.056	1.978

Average energy cost (Wh)

Thank you!

Q & A