CS7351 传感系统原理与设计

Principles and Design of Sensing Systems

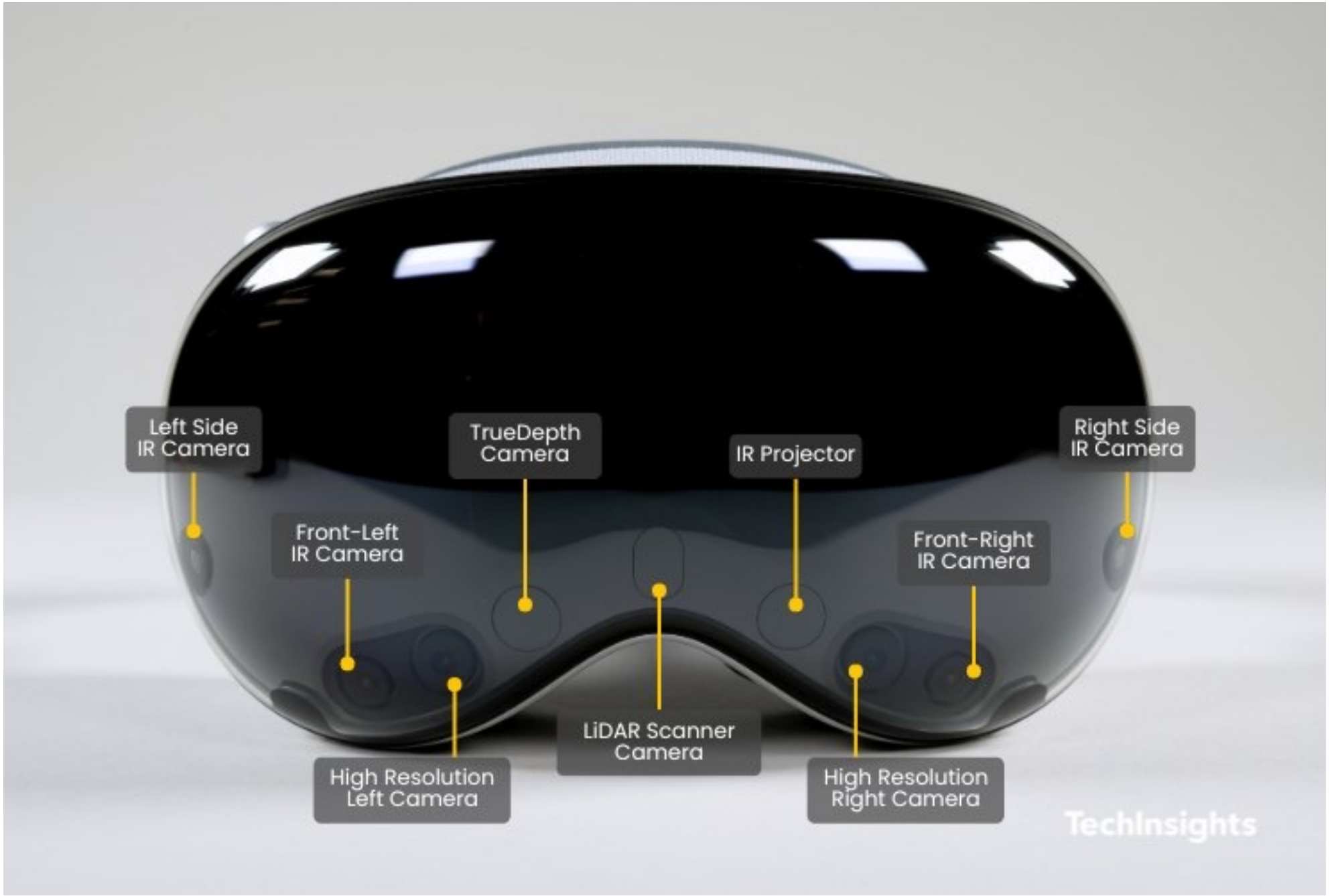
Camera-based Sensing Systems (摄像头传感器)

陈东尧

上海交通大学

chendy@sjtu.edu.cn

2024年秋季


Left Side
IR Camera
TrueDepth
Camera
IR Projector
Right Side
IR Camera
Front-Left
IR Camera
Front-Right
IR Camera
LiDAR Scanner
Camera
High Resolution
Left Camera
High Resolution
Right Camera
TechInsights

Outline

- The overview of camera-based sensing system
- Camera-sensing techniques
  - On-board object detection
  - Task offloading
  - Depth estimation

Camera-based System

- Various camera devices in life.

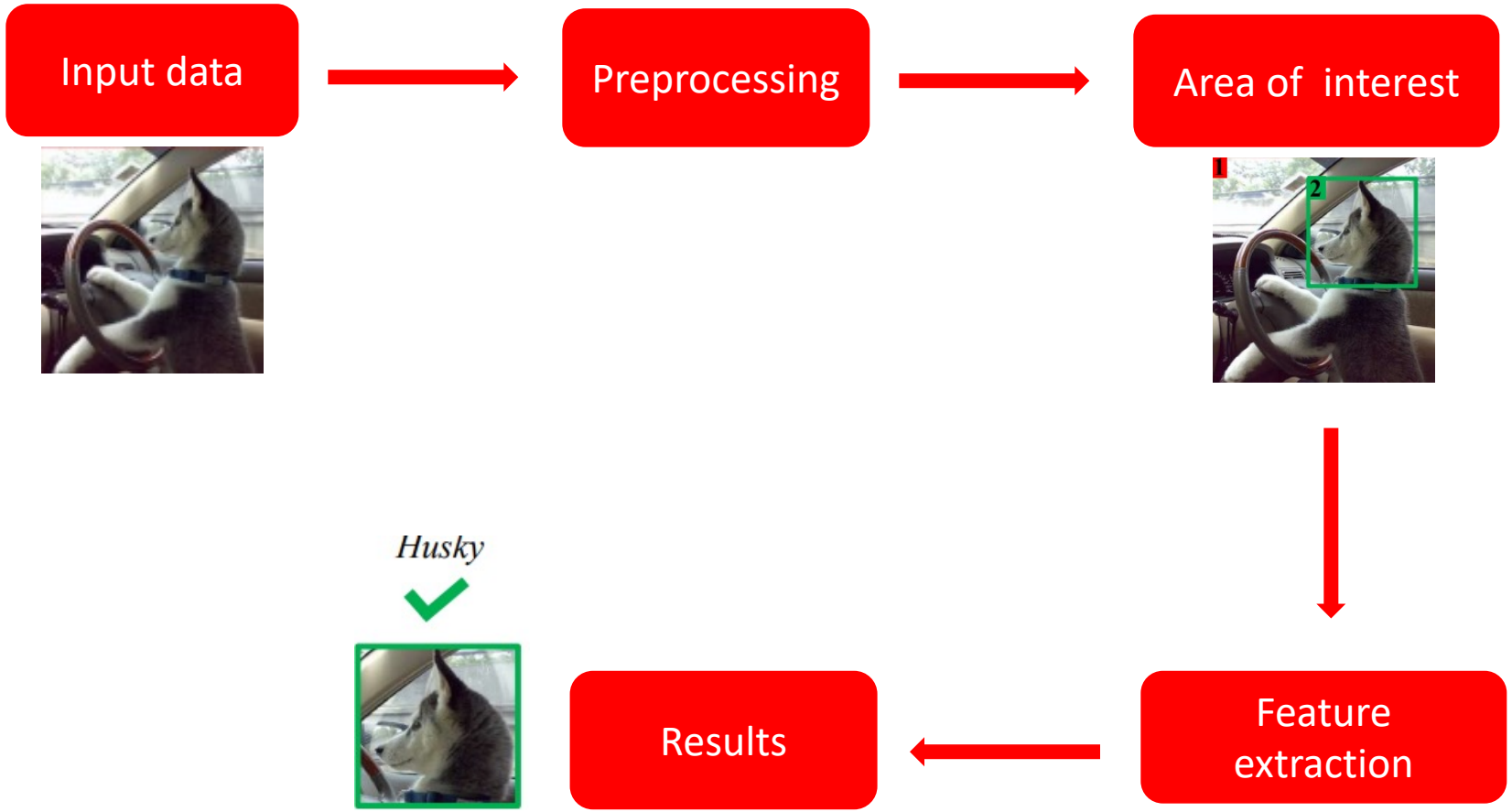
Camera-based System

- Why are camera-based methods so popular?
  1. The rapid development and wide application of computer vision(CV) algorithms.
  2. The computing power of GPU and CPU have been rapidly improving.
  3. Ubiquitous camera device.
  4. Development of the edge computing and cloud computing.

Camera-based Systems: Challenges

1. End-to-end latency.
2. Bandwidth and energy-efficiency.
3. Occlusion issue.
4. Privacy.

Camera-based System: Overview

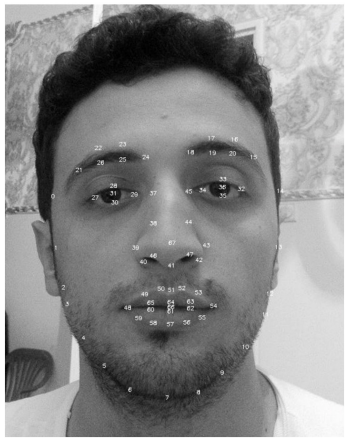


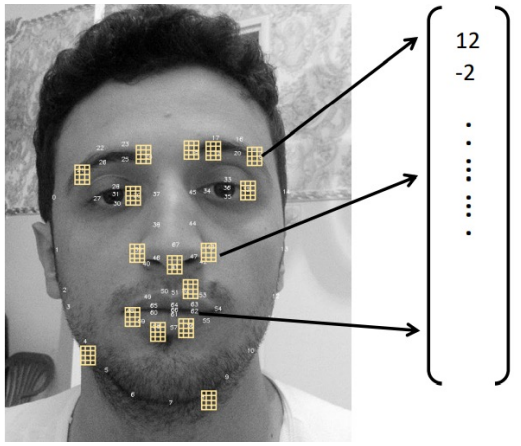
Preprocessing

- Color space.
- Geometric transformation.
- Cutting and splicing.
- Denoise.
- Image enhancement.

Feature Extraction

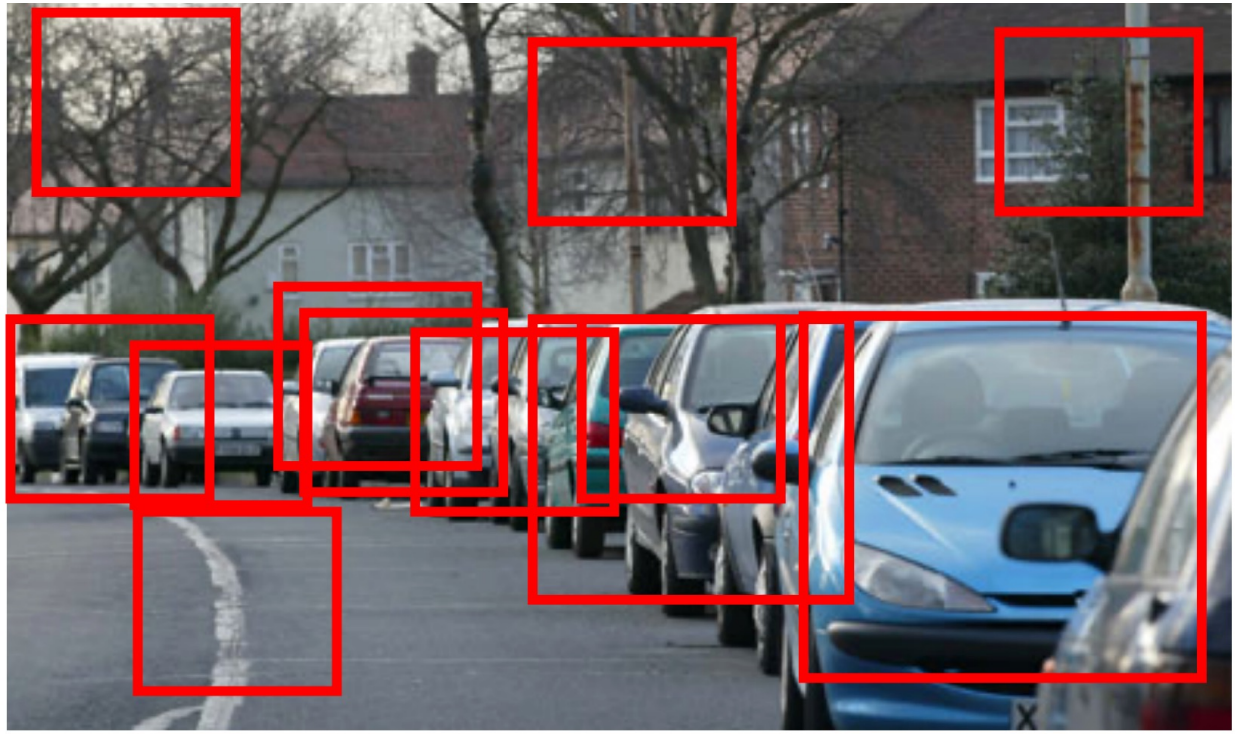
- Representative feature extraction methods
  - SIFT (尺度不变特征变换)
  - HOG (方向梯度直方图)
  - SURF (把SIFT中的高斯二阶微分的模板进行了简化)
  - ORB (ORB特征描述算法的运行时间远优于SIFT与SURF，可用于实时性特征检测)



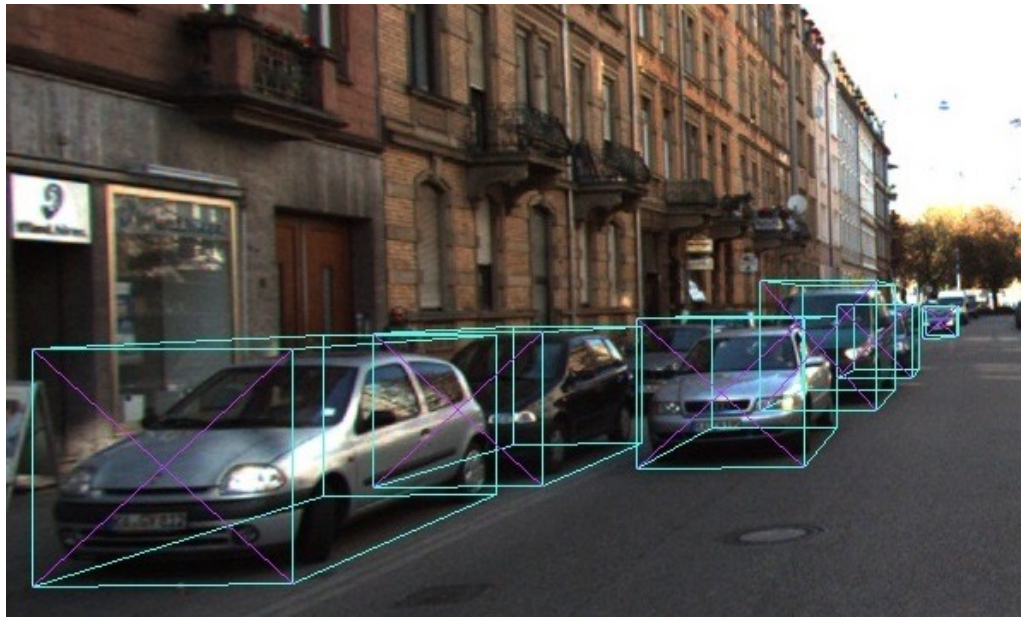


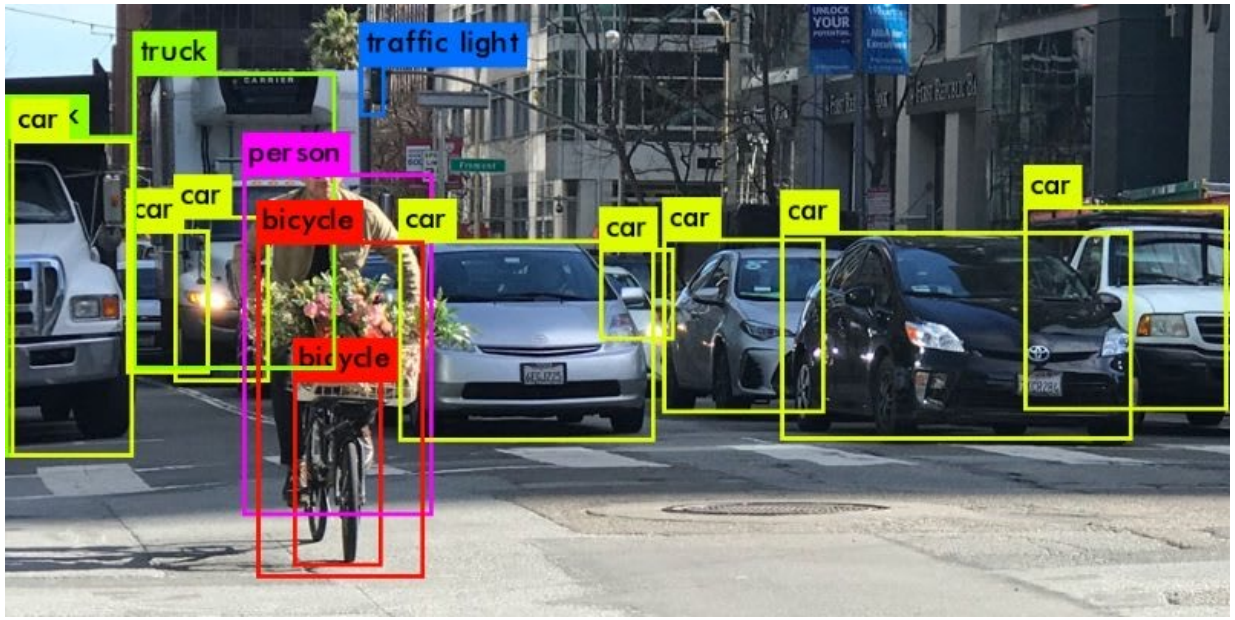
计算机视觉中的传统特征提取方法总结
https://blog.csdn.net/qq_29462849/article/details/121965502

The camera-sensing system, before the convolutional neural network



The camera-sensing system, after the convolutional neural network

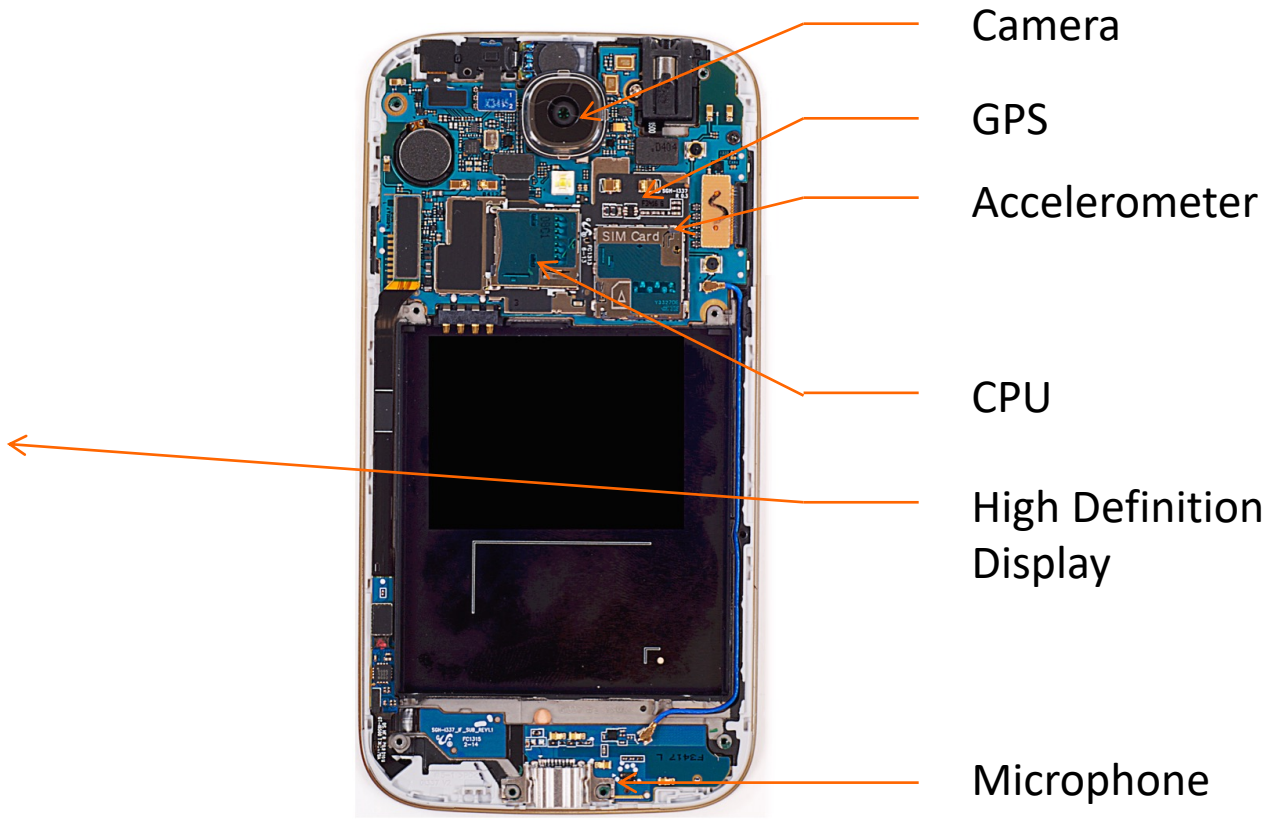




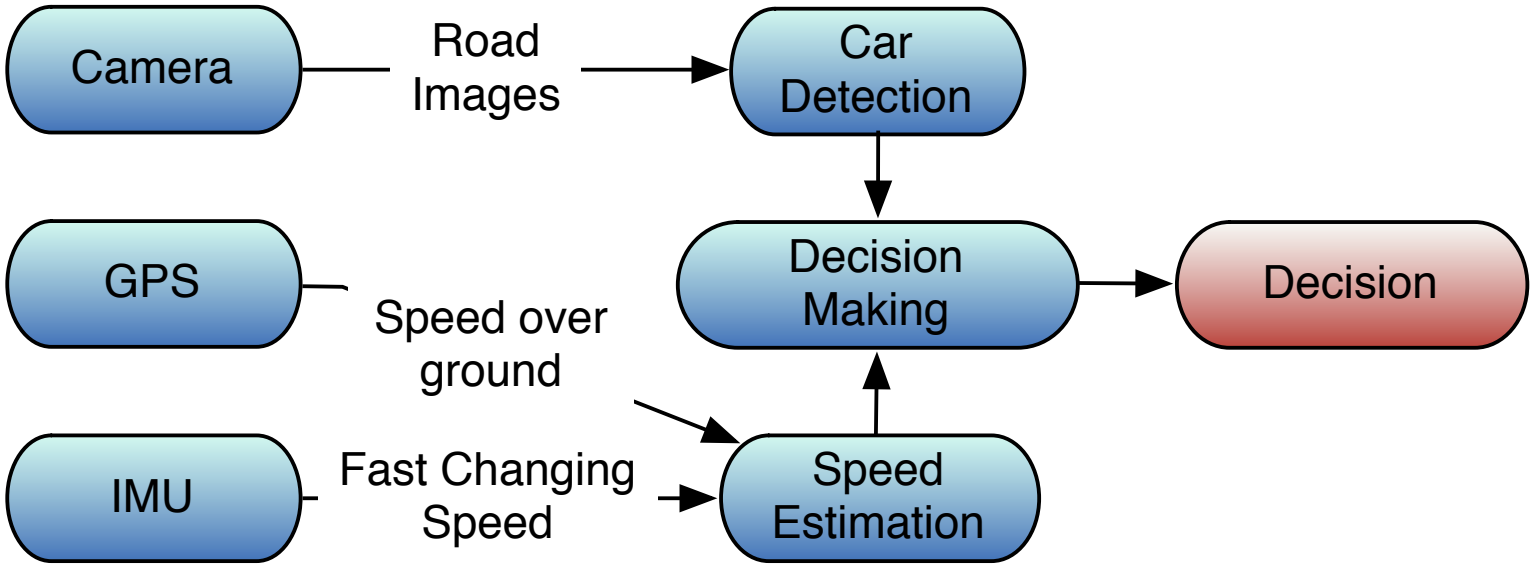
Onboard Object Detection
Driving Assistant with Multiple Sensors on Smartphone

DAWN: a Inside Look

- Driving Assistance With multiple seNsors on smartphone



Distance Detection Module



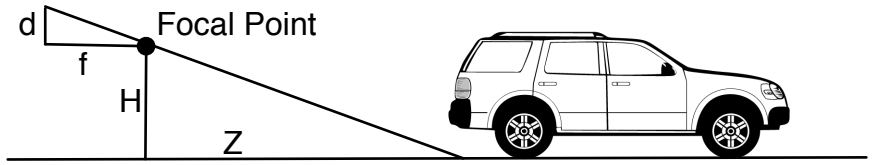
- Challenges:
  - Reliable speed estimation
  - Implement reliable car detection

Detection of Frontal Car

- Train classifier for car detection under different environments
  - Use HOG features of the back end
  - Training samples includes 1916 positive images and 20160 negative samples

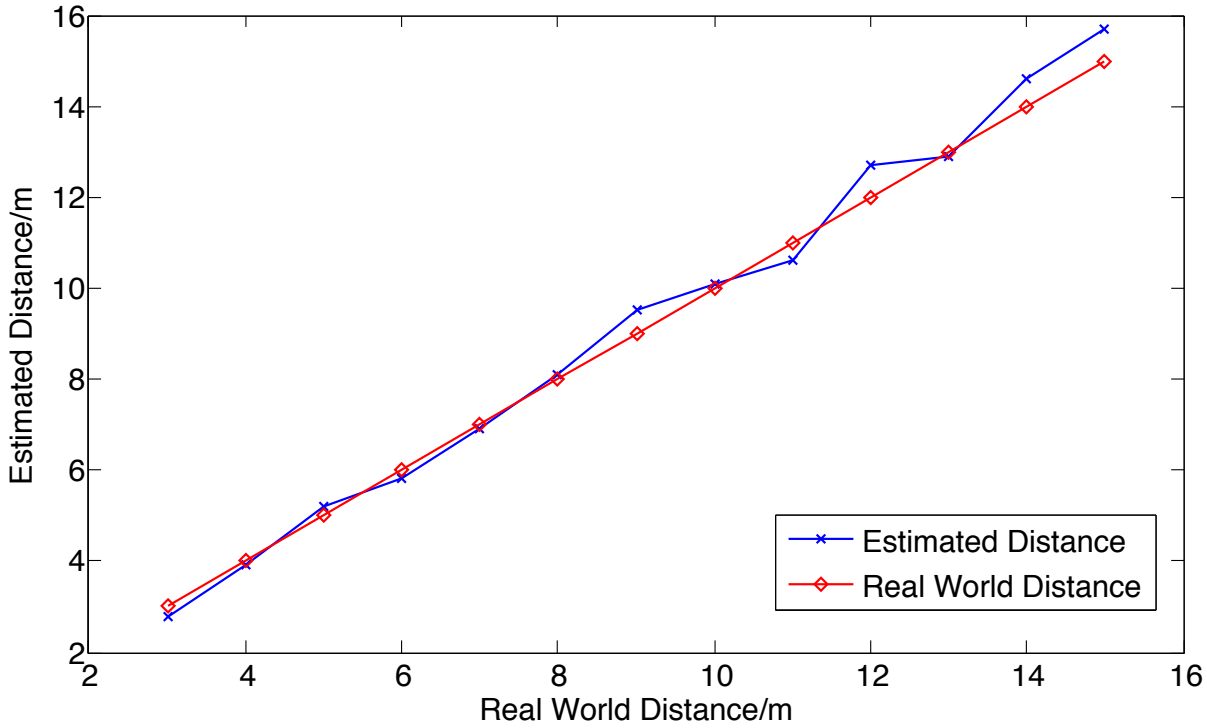
Distance Measurement

- Get the real-world from the pixel distant distance d



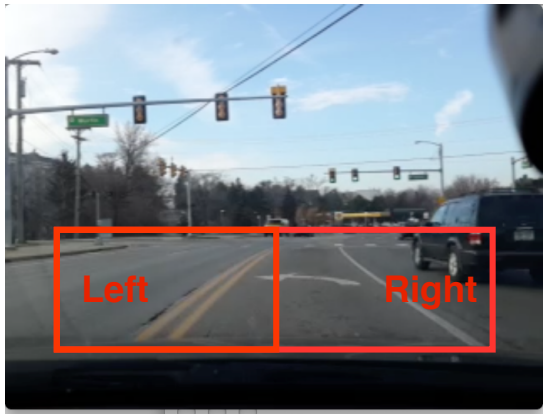
$$Z = f\frac{H}{d}$$

- Performance:
  - Accuracy



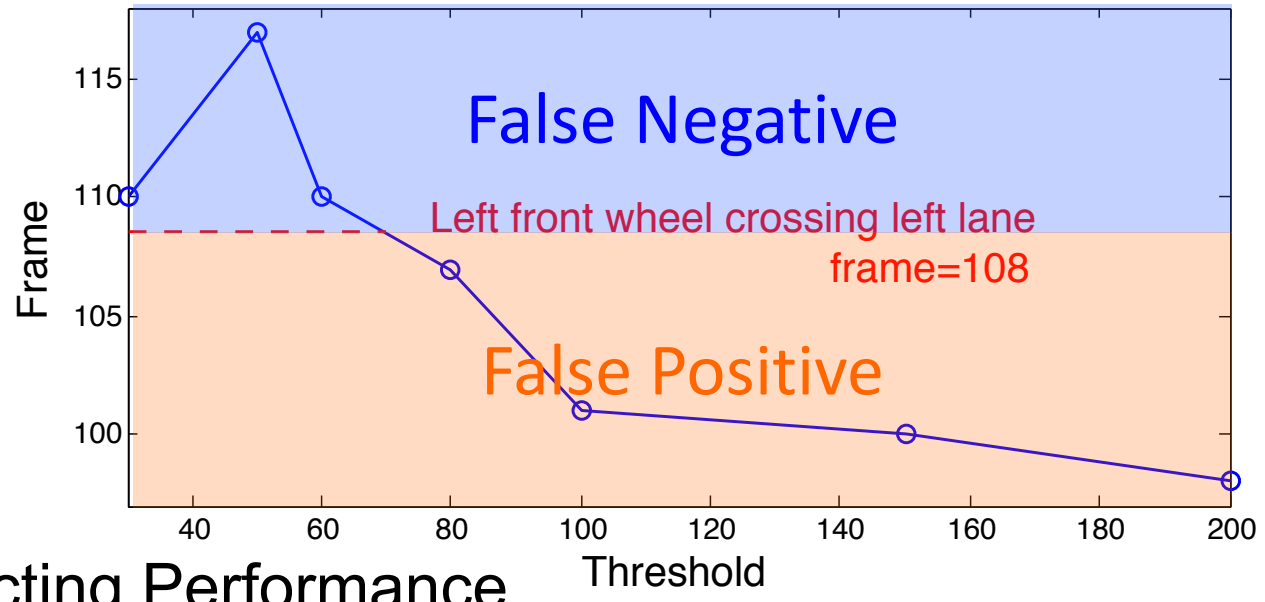
Lane Detection

- Detection the lane from two windows
  - Detecting two lanes concurrently by running threads on smartphone
  - Counting lanes in each window



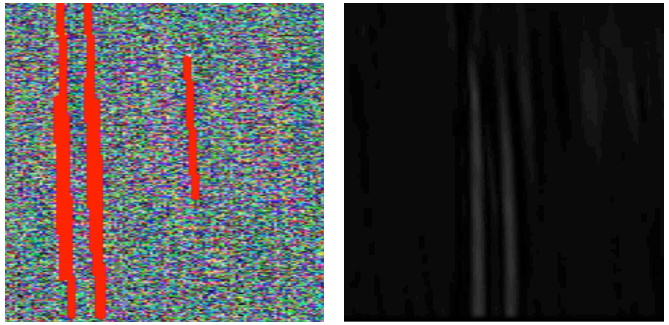
Lane Detection

- Detection result related to the threshold of probability Hough Transmission

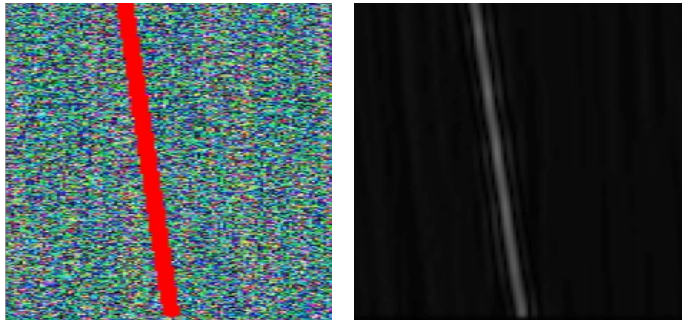


- Detecting Performance

Left window: 5 lines

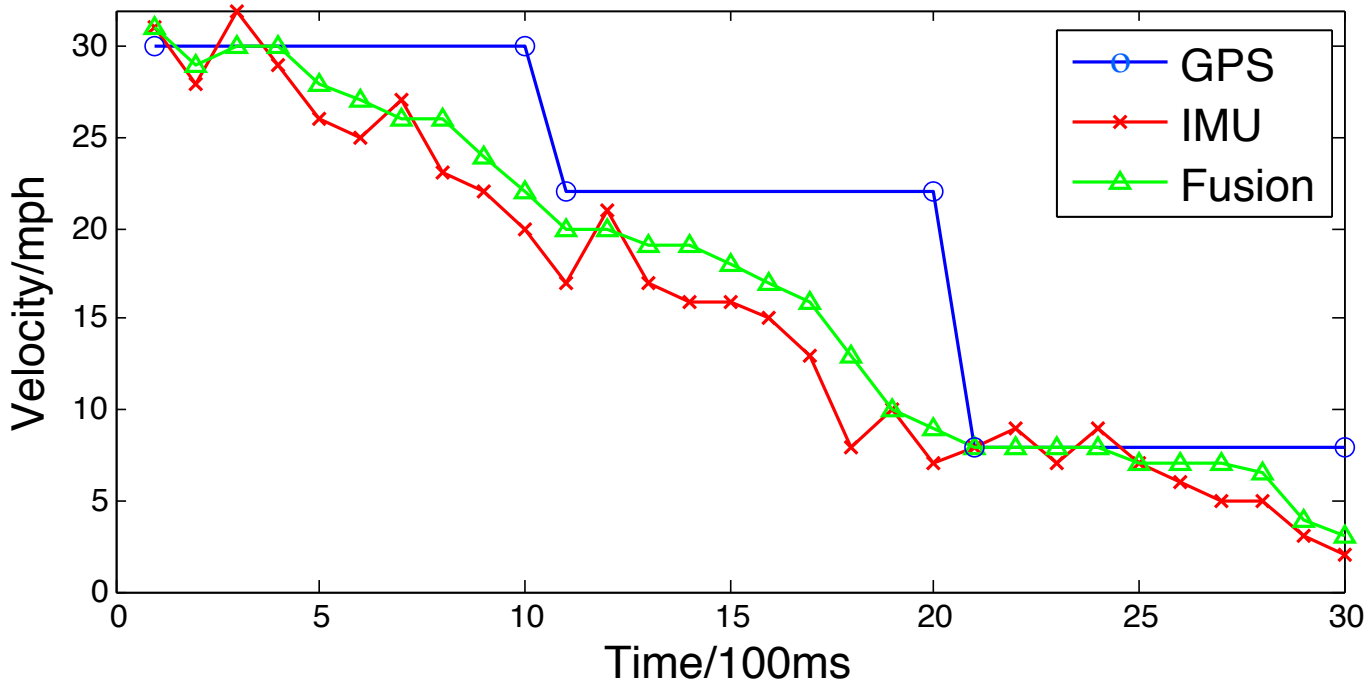


Right window: 2 lines

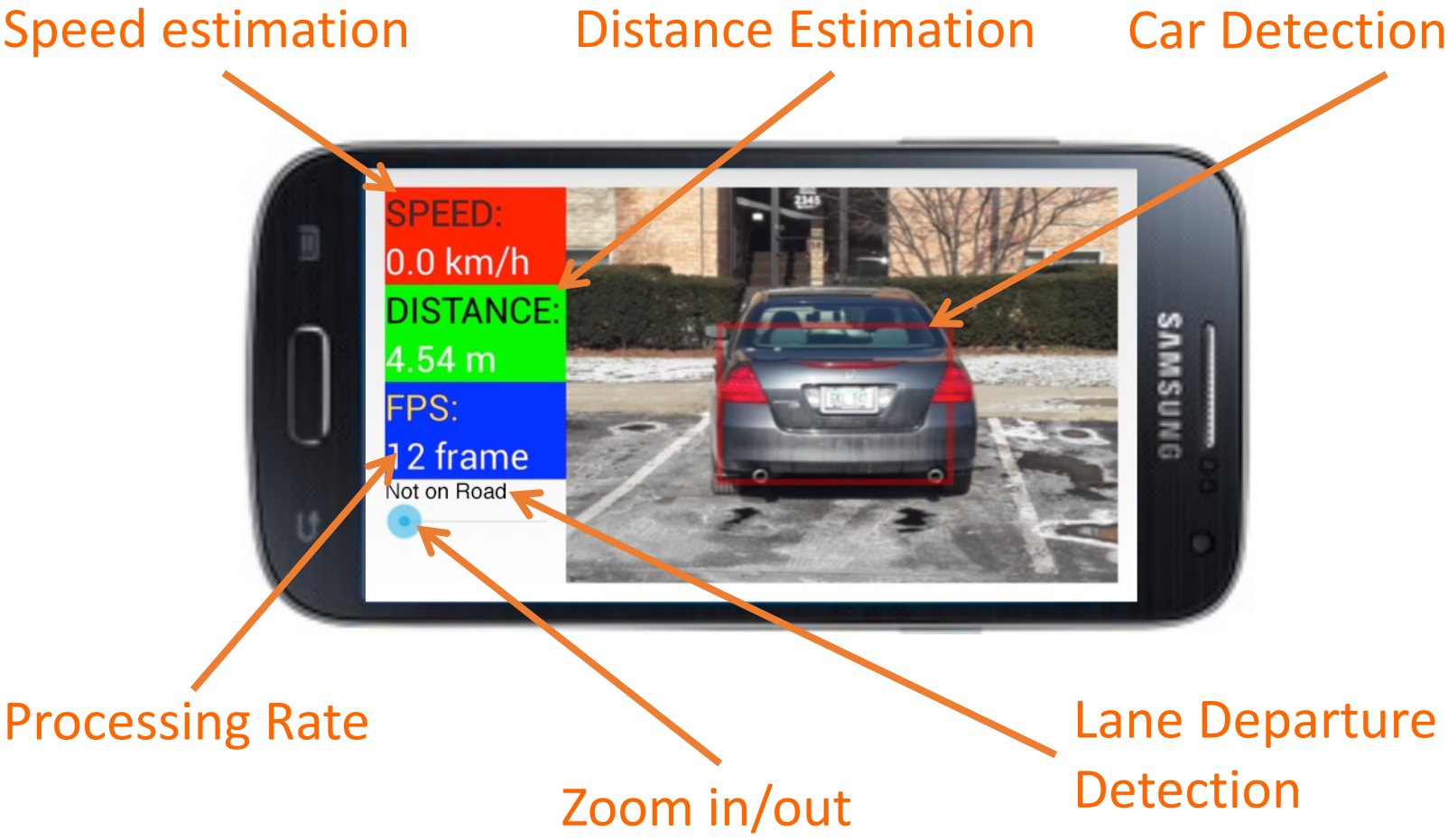


Speed Estimation

- Problem:
  - Speed information from GPS is slow.
  - Speed information from IMU is noisy
- Solution: better result from fusing the data from these two sensors



User Interface

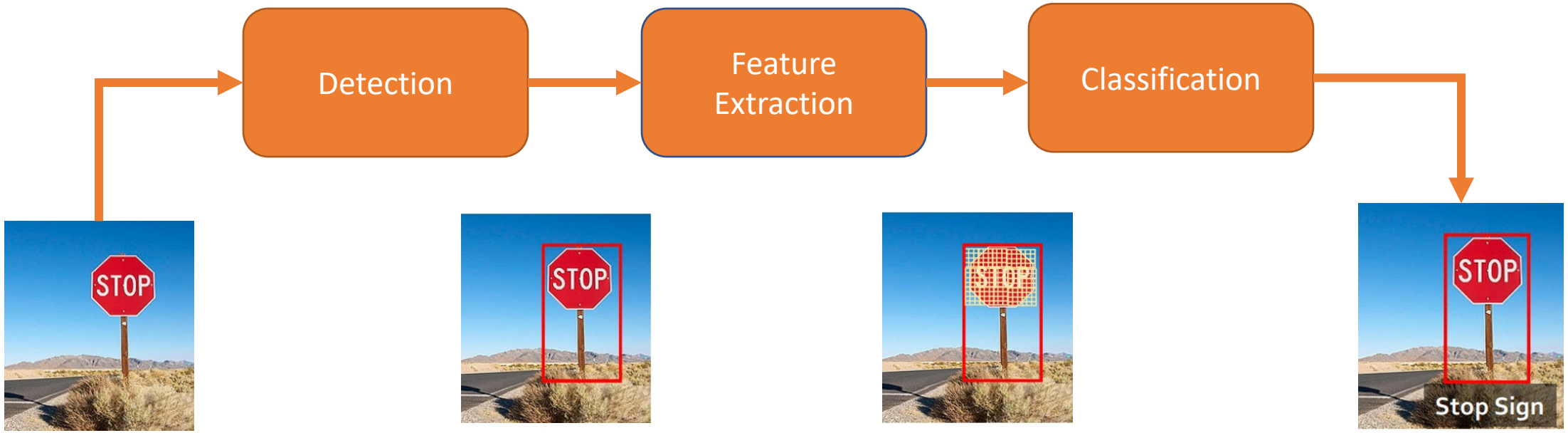


Task Offloading

Glimpse: Continuous, Real-Time Object Recognition on Mobile Devices

Chen, T. Y. H., Ravindranath, L., Deng, S., Bahl, P., & Balakrishnan, H. (2015, November). Glimpse: Continuous, real-time object recognition on mobile devices. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems* (pp. 155-168).

Motivation: The Legacy Classification Pipeline is not Suitable for Mobile Devices



Motivation: The Legacy Classification Pipeline is not Suitable for Mobile Devices

- Current status of mobile-based camera sensing:
  - Take a picture, then run the sensing algorithm. However

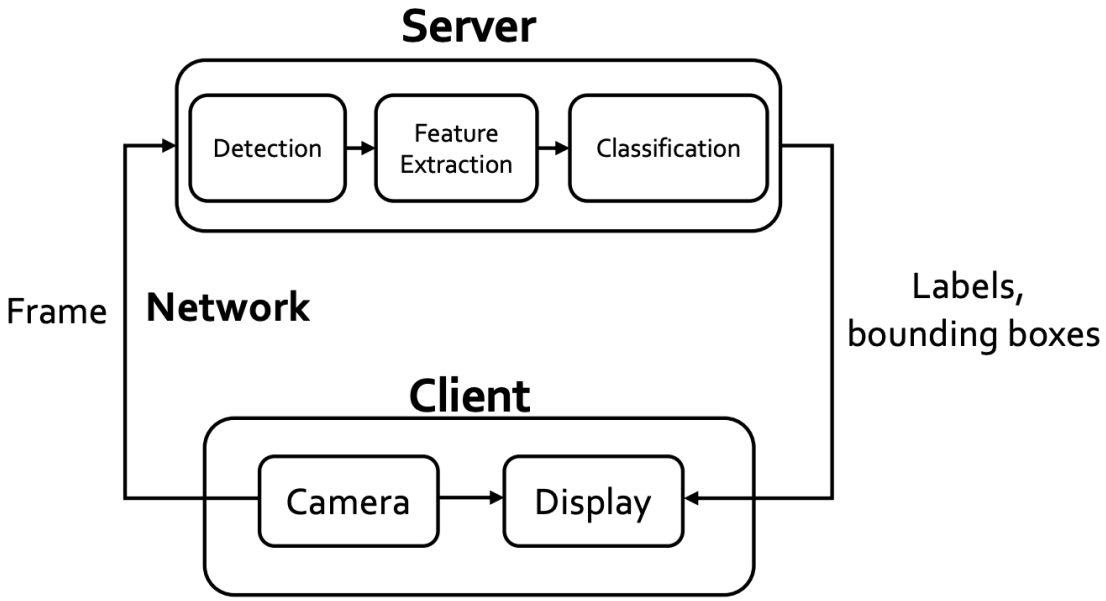
Early design: picture-based object recognition

Existing Solution

- Offload the sensing tasks to the server?
- Computationally expensive and memory-intensive
  - Server is 700x faster than Google Glass

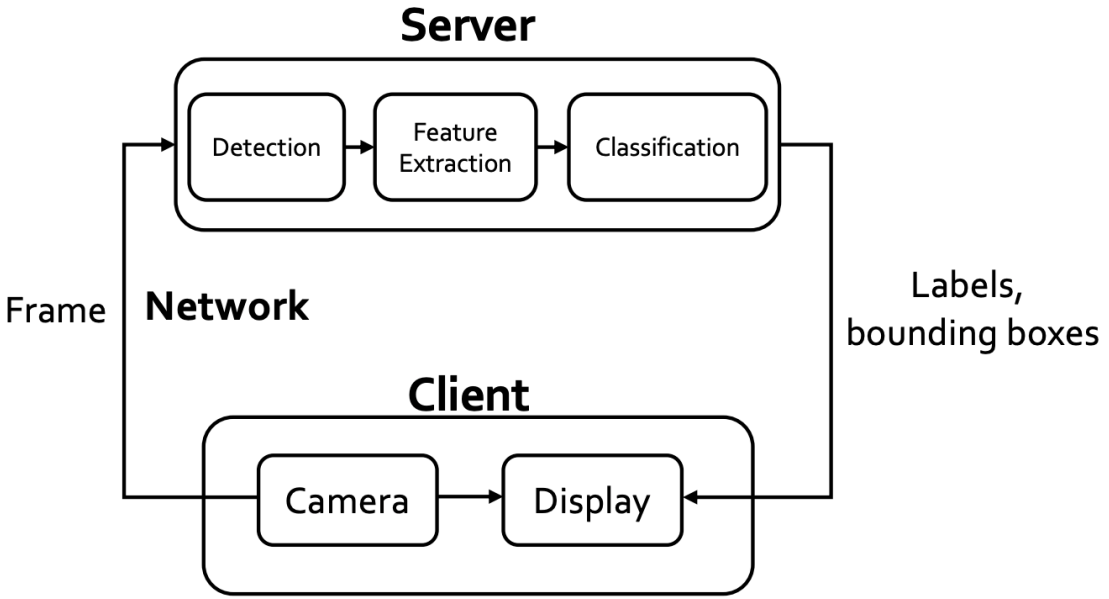
Existing Solution

- Offload the sensing tasks to the server?
- Computationally expensive and memory-intensive
  - Server is 700x faster than Google Glass



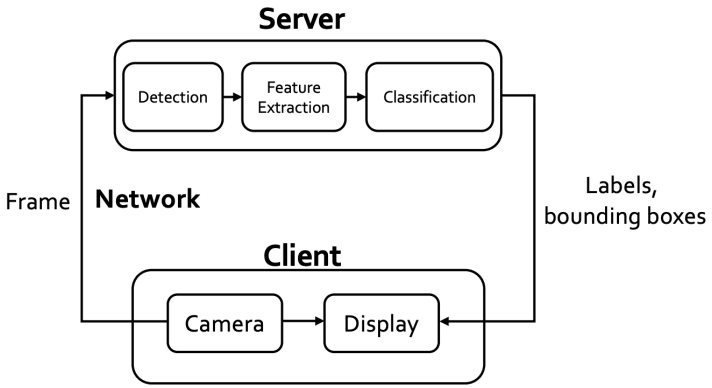
Challenges

1. End-to-end latency lowers object recognition accuracy
2. Bandwidth and energy-efficiency



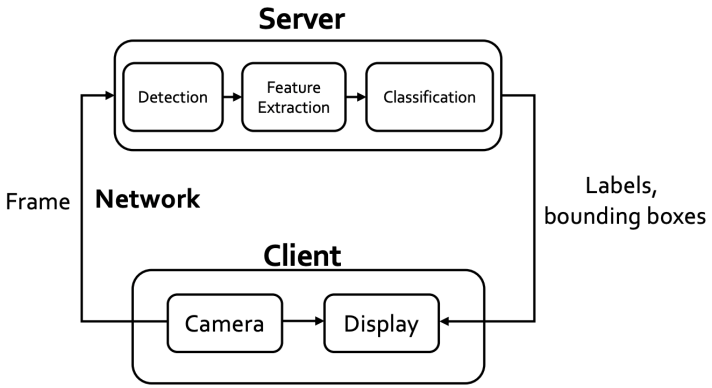
Solution

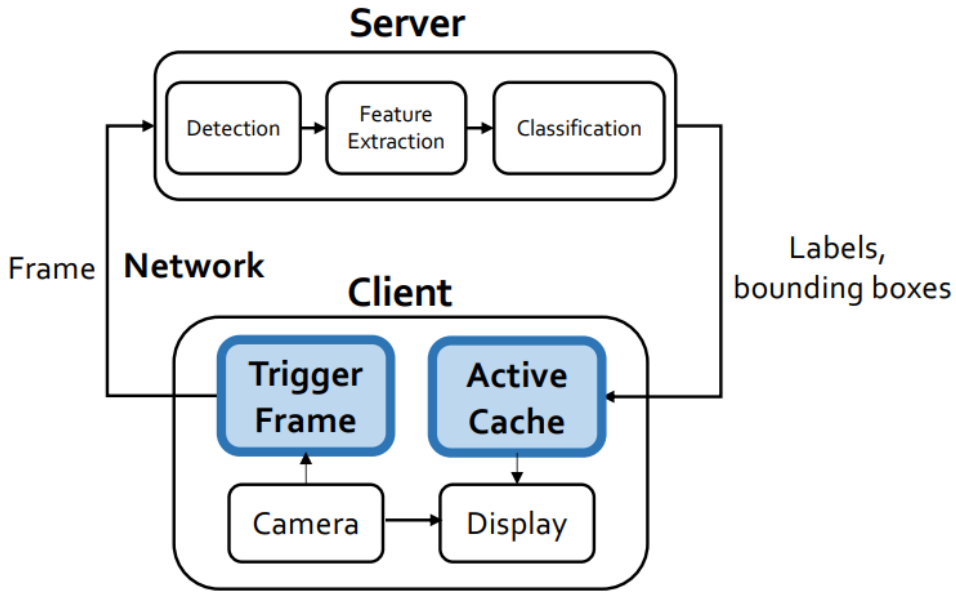
- Strategically send certain trigger frames to the server.
  1. Measuring scene changes from the previously processed frame.
  2. Detecting tracking failure.



Solution

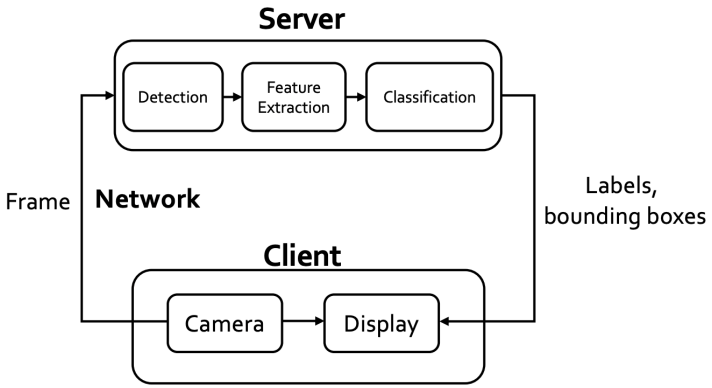
- Strategically send certain trigger frames to the server.
  1. Measuring scene changes from the previously processed frame.
  2. Detecting tracking failure.

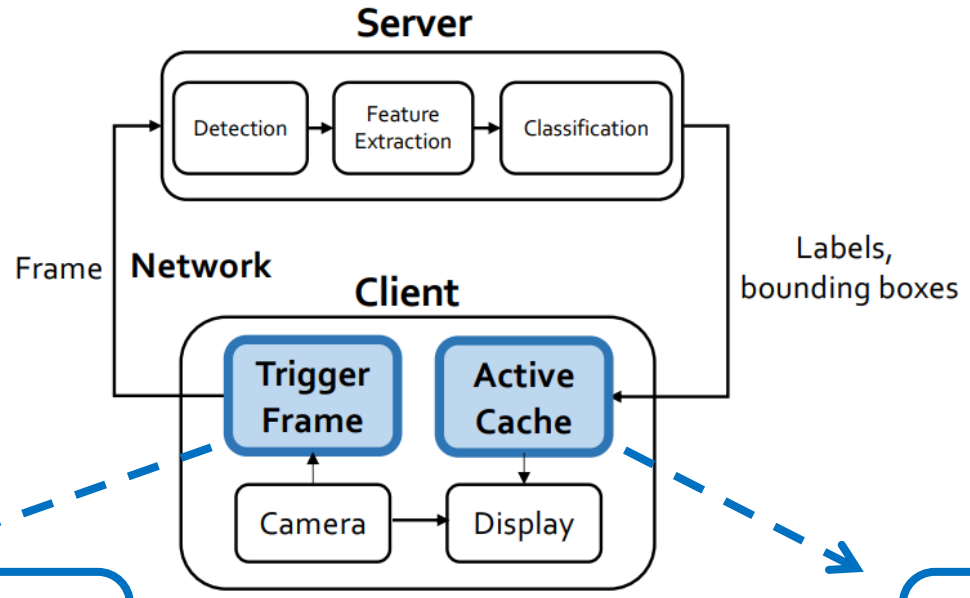




Solution

- Strategically send certain trigger frames to the server.
  1. Measuring scene changes from the previously processed frame.
  2. Detecting tracking failure.





Lower bandwidth and energy cost

Sustain the accuracy

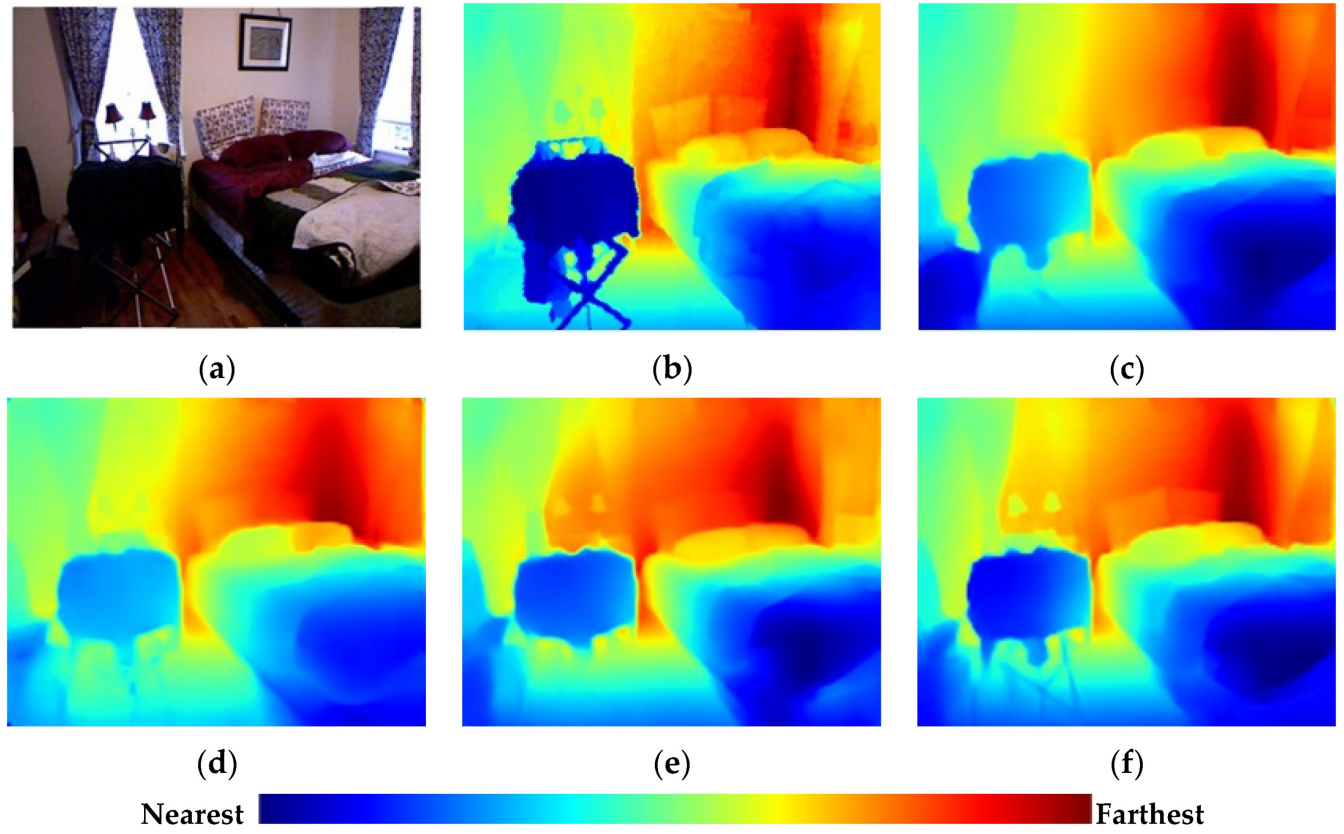
Depth Estimation

MobiDepth: Real-Time Depth Estimation Using On-Device Dual Cameras

https://www.microsoft.com/en-us/research/uploads/prod/2022/09/mobicom22-final138.pdf

Motivation

- Depth estimation is essential for AR / VR applications. But it is expensive



Challenges

- Dual-camera system is common on smartphones. But it is not ready for depth sensing
- Can we achieve depth sensing with ONLY hardwares on smartphones?

Example 3: MobiDepth

Three novel technologies:

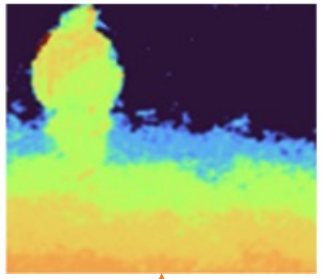
1) Iterative field-of-view cropping

2)Heterogeneous camera synchronization

3)Mobile GPU-friendly stereo matching

Example 3: MobiDepth

Depth Map



Iterative field-of-view cropping

Offline

Left Camera
(Tele FoV)

Right Camera
(Wide FoV)

Input data

Heterogeneous camera synchronization

Mobile GPU-friendly stereo matching

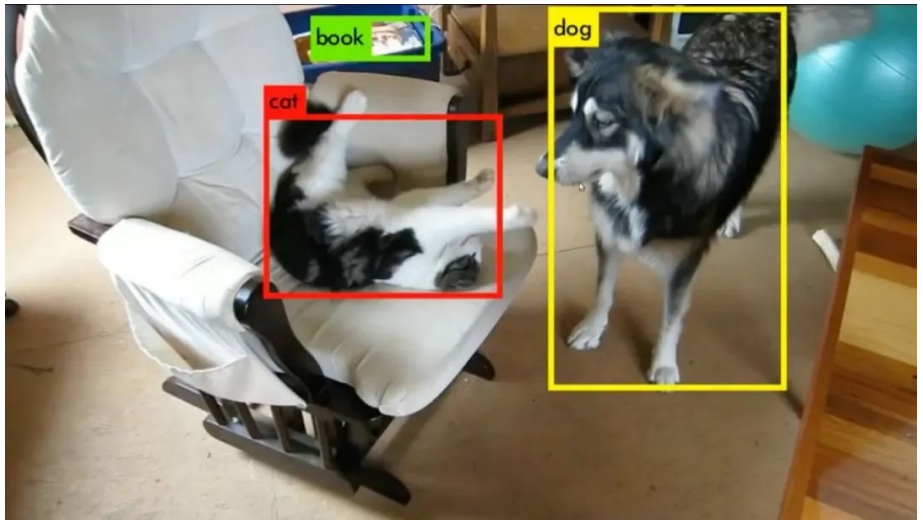
Online

Design Your Own Camera-based System

- Know the goal of the system.
- Collect the data for benchmarking test
- Design the computer vision algorithm
- Integrate into the mobile device.

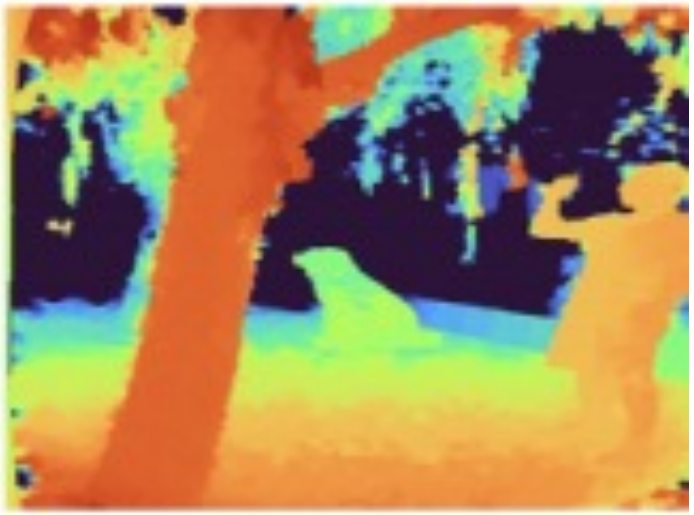
App 1: Object Recognition

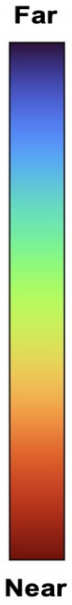
- What are the challenges of real-time object recognition based on the camera?



App 2: Depth Estimation

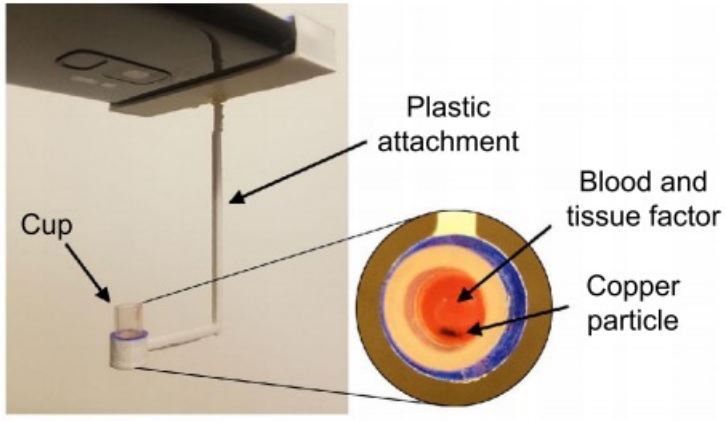
- What is the depth estimation?
- What are the implementation methods?
  1) Dedicated depth sensors.
  2) Learning-based depth prediction.
  3) Using monocular camera or dual cameras on mobile devices.





App 3: Healthcare

- Are there any other application scenarios?
- Can we find other solutions?



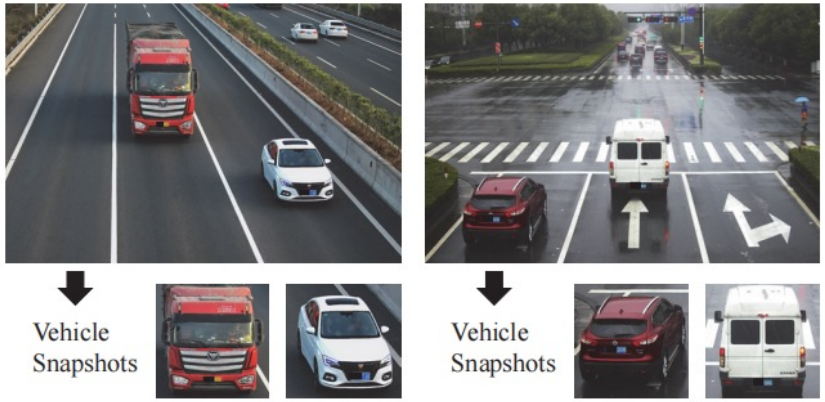
Blood clot testing

Measurement of physiological vital signs

App 4: Traffic Monitoring

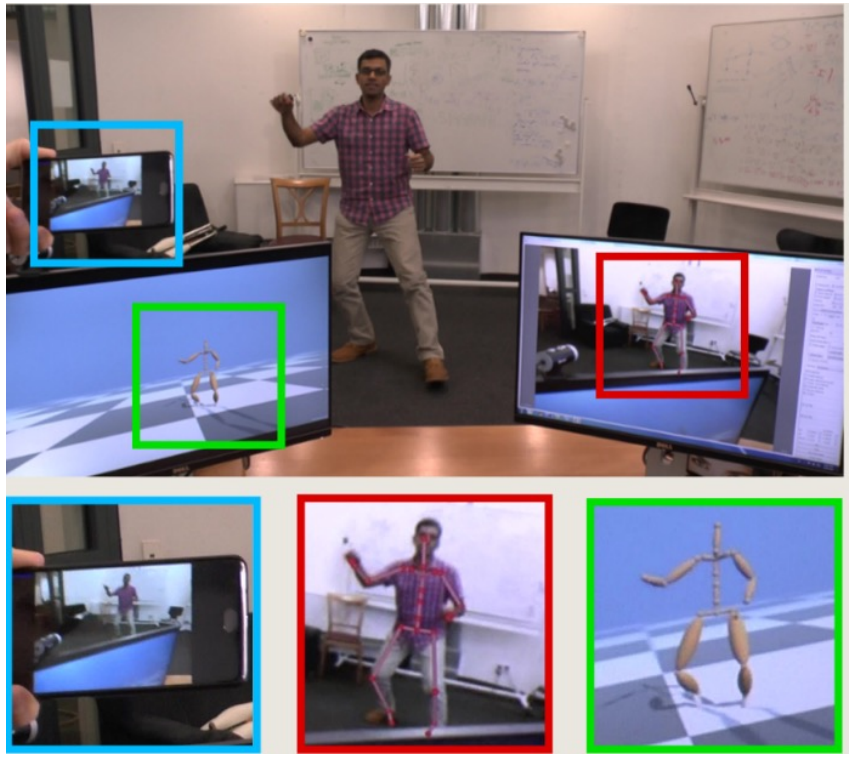
Safe driving monitoring

Driver emotion recognition



Vehicle trajectory reconstruction

App 5: Pose Estimation

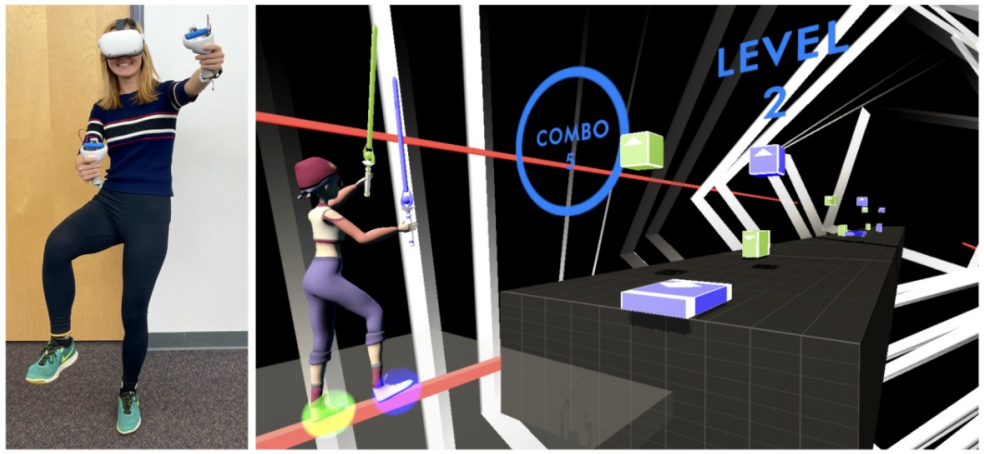


Pose reconstruction

Motion capture

App 6: Hard Problem in Virtual Reality

- One method to solve the problem of "no legs" in VR applications.



Controller Pose： https://www.figlab.com/research/2022/controllerpose

Useful Tools

- OpenCV for
  - Android: https://opencv.org/android/
  - Raspberry Pi: https://raspberrypi-guide.github.io/programming/install-opencv
- TensorFlow lite (for Android only):
  - https://www.tensorflow.org/lite

- **特征提取的方法总结：**
  - https://blog.csdn.net/qq_29462849/article/details/121965502