Real-Time Cyber Physical Systems
Application on MobilityFirst

Winlab Summer Internship 2015
Karthikeyan Ganesan, Wuyang Zhang, Zihong Zheng
Shantanu Ghosh, Avi Cooper
TEAM MEMBERS

Karthikeyan Ganesan
Wuyang Zhang
Zihong Zheng
Shantanu Ghosh
Avi Cooper
PRELIMINARY GOAL OF OUR PROJECT

CPS Application based on MF

Server side:
Implement server application for object recognition; Return the result

Client side:
Run an instance of camera system; Transmits video in standard format; Simple graphical interface to display results
CURRENT FRAME

Image Recognition:
Done essential time analysis for different phases in Image Matching. Have some progress on Speed optimization. Also working on strategies to improve Accuracy.

Cloud Computing:
Set up the Hadoop Cluster, constructed by Master and Slaves that could run Map & Reduce jobs. Now working on Hadoop Image Processing Interface (HIPI).

Application:
Tried to debug some MF Android applications done before (mfstack, mfping, etc.)
Learning how to set up the WIFI access point on Router to enable the MF connection.
IMAGE PROCESSOR

• Server Startup
  • Load Descriptors into memory

• Image Matching
  • Isolate Descriptors from test image
  • Find Matching Descriptors from Database using the Flann Class
  • Algorithm to determine Object from matched descriptors
IMAGE PROCESSOR OPTIMIZATION: SPEED

- Server Startup
  - Load Descriptors into memory

- Image Matching
  - Isolate Descriptors from test image
  - Find Matching Descriptors from Database
    - Build a KD tree
    - Perform a knn search
  - Algorithm to determine Object from matched descriptors
IMAGE PROCESSOR OPTIMIZATION: SPEED

Old

New
IMAGE PROCESSOR OPTIMIZATION: SPEED

• Descriptor Isolation
  • Largest portion of computing time.
  • Fixes:
    • Try utilizing the GPU to perform SURF
    • Try a method that better utilizes multiple CPU cores
    • Try different feature detector other than SURF
IMAGE PROCESSOR OPTIMIZATION: ACCURACY

• True Positives
  • Detects objects in test image that exist in Database with close to 100% accuracy

• False Positives
  • Often Falsely recognize objects that don’t exist in the image (Close to 50%)

• Fix
  • Improve the Algorithm used to determine object from Matched Descriptors with additional checks to confirm the actual existence of the object.
MFStack is used to install the MF stack on phone.

Also a launcher to start and stop the mf stack service.

Devices under MF network is actually communicating through the MF stack.

MFPing achieves the basic ping function for MF such as the ping for TCP/IP.

We plan to use it to test the MF connection after we set up the access point on router.
PLAN TO SET UP THE CONNECTION

Glass captures video

Android phone handles networking

Bluetooth

MF

Access Point on router

HADOOP Slaves nodes

Master node as Server manages/allocate image recognition jobs

ORBIT outdoor nodes
Background:
Optimized image processing algorithm, current system process 1,000 images within 113 ms.

Impediment:
Load the large size of descriptors in the database into a local memory before matching images.
1,000 images : 1 GB, linear accumulation -> 10,000 images : 10 GB

HIPI:
Image processing interface under the framework of HADOOP. Distribute the database into several machines so as to support over 10,000 images.
Next Week Plan

- Set up the connection between Android Phone and orbit nodes through an access point.
- Continue develop the client program based on google glass and Android phone.
- Use HIPI to embedded Image Processing Program and Hadoop Framework.
Questions?