Feature extraction using video processing and QoI
Manisha Mukherjee, Thomas F. La Porta

INSR

PENNSTATE 1855

Background

The difference in the Quality of Information (QoI)[1,2] arising due to the limitations in processing times on various devices has been analyzed. Videos obtained from various traffic cameras have been processed to extract information on:

- the number of vehicles passing by in a certain amount of time,
- the color of vehicles,
- differentiating amongst the different types of vehicles, and
- the speed of the vehicles passing by.

The videos processed on a server (with higher processing power) have been compared with videos processed on an android phone (with lesser processing power).

Objective

- The QoI that we get from the server is not achieved in the information received from the android phone in the same amount of time.
- However, a comparable QoI could be obtained from the android phone by controlling the number of frames processed per second.

The purpose of this project is to determine when the video file can be locally processed to get a desired QoI (hence save bandwidth) compared to when the video has to be streamed to an alternate powerful server.

Methodology

OpenCV[3] was used to process the traffic videos. The number of vehicles were detected by:
1. Extracting frames from the video
2. Comparing each extracted frame with an existing image of vehicles by:
    1. Matching key points in the two images.
    2. Assuming the match was correct, calculate homography (projection matrix).
    3. Using the homography to project the corners of original image to draw a quadrilateral shape (which we refer to as rectangle under) perspective transformation.
3. Counting the number of rectangles to get the number of vehicles detected.

To maintain consistency over the languages and the codes used in both platforms, we have reused the server C++ code on the android platform

Results

Technologies Used

- Eclipse as the IDE,
- OpenCV for the video processing
- C++ for the video processing (In both cases of the server and the android phone).

To run the C++ code on the android phone we have used:
- Android SDK,
- Android Native Development Kit(NDK),
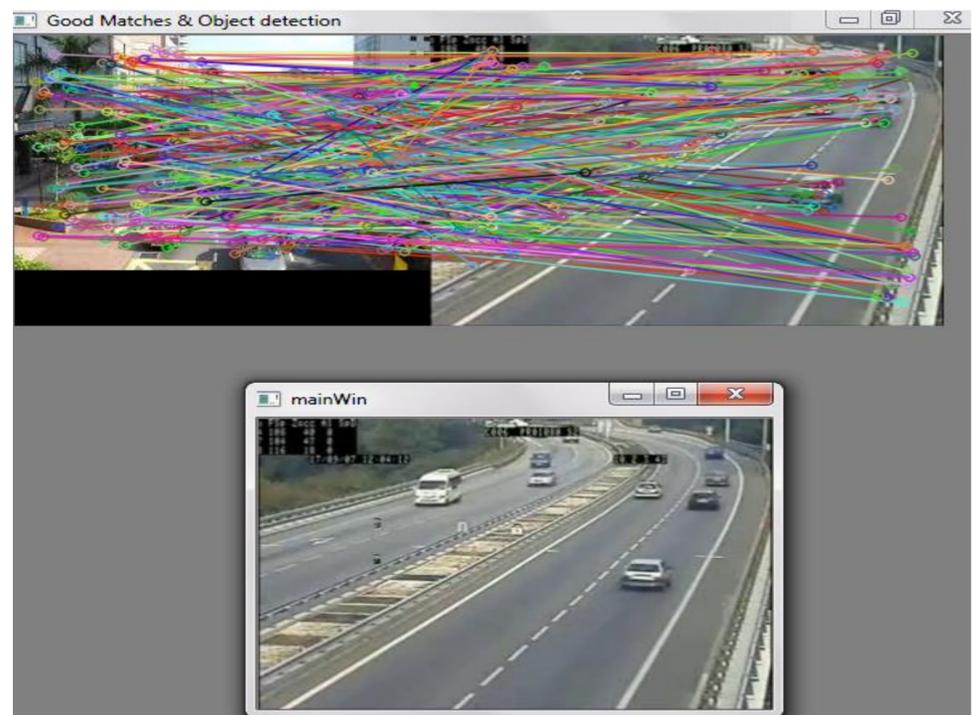- the ADT plugin for Eclipse, and
- OpenCV4Android SDK.

In both cases of the server and the android phone, we have analyzed video of the format MPEG-4 Part 14 or MP4.

References

1. Edwards, James, Ahmed Bahjat, Yurong Jiang, Trevor Cook, and Thomas F. La Porta. "Quality of information-aware mobile applications." *Pervasive and Mobile Computing* (2013).

2. Bar-Noy, Amotz, Greg Cirincione, Ramesh Govindan, S. Krishnamurthy, T. F. LaPorta, Prasant Mohapatra, M. Neely, and Aylin Yener. "Quality-of-information aware networking for tactical military networks." In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pp. 2-7. IEEE, 2011.

3. Gossow, David, Peter Decker, and Dietrich Paulus. "An evaluation of open source SURF implementations." *RoboCup 2010: Robot Soccer World Cup XIV*. Springer Berlin Heidelberg, 2011. 169-179.