Odessa: Enabling Interactive Perception Applications on Mobile Devices

Moo-Ryong Ra, Anmol Sheth, Lily Mummert, Padmanabhan Pillai', David Wetherall, Ramesh Govindan
Emerging Mobile Perception Applications

Sensing

GPS
Accelerometer

Computation
Dual-Core CPU

Communication
Cloud Infrastructure

Sensing Applications
Activity Recognition
Health, Traffic Monitoring
Location-Based Service
Participatory Sensing
Vision-based Interactive Mobile Perception Applications

Motivation

Problem

Measurement

Design

Evaluation

Face Recognition

Object and Pose Recognition

Gesture Recognition

Type your name and press [enter] when done.
Common Characteristics

Interactive
- Crisp response time (10 ms ~ 200 ms)

High Data-Rate
- Processing video data of 30 fps
- Stream based

Compute Intensive
- Computer Vision based algorithms
Enabling Mobile Interactive Perception

**Performance**

<table>
<thead>
<tr>
<th>Application</th>
<th>Throughput</th>
<th>Makespan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face Recognition</td>
<td>2.50 fps</td>
<td>2.09 s</td>
</tr>
<tr>
<td>Object and Pose Recognition</td>
<td>0.09 fps</td>
<td>15.8 s</td>
</tr>
<tr>
<td>Gesture Recognition</td>
<td>0.42 fps</td>
<td>2.54 s</td>
</tr>
</tbody>
</table>

*All running locally on mobile device*

*Video of 1 fps*
How to speed things up?
Main Focus

Data Flow Structure

Offloading + Parallelism

System Support

Enable Mobile Interactive Perception Application
Contributions

What factors impact offloading and parallelism?

Measurement

How do we improve throughput and makespan simultaneously?

Odessa Design

How much benefits can we get?

Evaluation
Measurement

- Input Data Variability
- Varying Capabilities of Mobile Platform
- Network Performance
- Effects of Parallelism
Lesson I : Input Variability

The system should adapt to the variability at runtime

Impact of input variability
## Lesson II: Effects of Data Parallelism

### Object and Pose Recognition

<table>
<thead>
<tr>
<th># of Threads</th>
<th>% Frames With faces</th>
<th>Mean Exec time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>61.66</td>
<td>149</td>
</tr>
<tr>
<td>2</td>
<td>24.87</td>
<td>15.6</td>
</tr>
<tr>
<td>3</td>
<td>38.11</td>
<td>18</td>
</tr>
</tbody>
</table>

The level of data parallelism affects **accuracy** and performance.
Latency vs. Throughput

- Makespan ~ latency
  - How long it takes to process an input
- Throughput
  - Rate at which inputs are processed
- Both are important
Summary: Major Lessons

Offloading decisions must be made in an adaptive way.

The level of data parallelism cannot be determined a priori.

A static choice of pipeline parallelism can cause sub-optimal performance.
Offloading DECision System for Streaming Applications

Application
- Odessa Profiler
- Sprout
- Cloud Infrastructure

Network

Runtime
- Profiler
- Decision Engine
- Odessa
- Sprout
- Mobile Device

Input Video Stream
Incremental decisions adapt quickly to input and platform variability.
Data-Flow Graph

Face Recognition
- Source
- Copy
- Tiler
- Feature merger
- Graph Splitter
- Classify
- Reco. Merge
- Display

Object Pose Estimation
- Source
- Copy
- Scaler
- Tiler
- Feature splitter
- Feature merger
- Model matcher
- Match joiner
- Cluster splitter
- Cluster joiner
- RANSAC
- Display

Gesture Recognition
- Source
- Copy
- Scaler
- Tiler
- MotionSIFT
- Feature merger
- Descaler
- Copy
- Display
- Classify
- Face detect
- Face merge
- Descaler
- Display
- Copy
Odessa finds a desirable configuration automatically.
Resulting partitions are often very different for different client devices.

<table>
<thead>
<tr>
<th>Client Device</th>
<th>Stage Offloaded and Instances</th>
<th>Degree of Pipeline Parallelism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile Device</td>
<td>Face detection (2)</td>
<td>3.39</td>
</tr>
<tr>
<td>Dual Core Notebook</td>
<td>Face Detection (1) Motion-SIFT Feature (4)</td>
<td>3.06</td>
</tr>
<tr>
<td></td>
<td>Face Detection (1) Motion-SIFT Feature (9)</td>
<td>5.14</td>
</tr>
</tbody>
</table>
## Performance Comparison with Other Strategy

### Object and Pose Recognition Application

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Throughput (FPS)</th>
<th>Makespan (Latency)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offline-Optimal</td>
<td>6.49</td>
<td>430 ms</td>
</tr>
<tr>
<td>Odessa</td>
<td>6.27</td>
<td>807 ms</td>
</tr>
</tbody>
</table>

Odessa performs 4x better than the partition suggested by domain expert, close to the offline optimal strategy.

Mobile Device