Ubiquitous and Mobile Computing
CS 403x: Visage: A Face Interpretation Engine for Smartphone Applications

Robert Esposito, Christopher Knapp, and Doruk Uzunoglu

Computer Science Dept.
Worcester Polytechnic Institute (WPI)
Introduction

- Mobile phone camera is a ubiquitous sensor like the microphone, accelerometer, etc.
  - However, it has not been exploited to nearly the same extent
- Emotion categories represented by universal facial expressions
Vision

- Set of sensing, tracking, and machine learning algorithms on smartphones
- Engine that interprets head poses and facial expressions to provide automatic feedback
Related Work

- Computer vision algorithms on mobile applications
  - SenseCam
  - MoVi
  - Recognizr
  - Google Goggles
- Mobile head pose trackers
  - PEYE
Design Considerations

- User mobility
  - Angle, motion, and light exposure level of mobile phones are unpredictable
- Limited phone resources
  - Video camera produces 50 times more data than microphone, 800 times more than accelerometer
  - Visage processes video streams in real time
System Architecture Design

- Sensing
- Preprocessing
- Tracking
- Inference

Fig. 1. Visage System Architecture
Preprocessing Stage

- Phone posture: Estimates gravity and motion using accelerometer and gyroscope readings
- Face detection with tilt compensation: Scans with decreasing window size until face detected
- Adaptive exposure: Corrects exposure level based on local lighting information within face region
Tracking Stage

- Feature points tracking
  - Identifies possible feature points on first frame and tracks their locations
- Pose estimation
  - POSIT algorithm estimates 3D geometry of user’s head using 2D feature points

Fig. 4. (a) Cylinder model and (b) Tracking with pose estimation.
Inference Stage

- Active appearance model
  - Describes 2D image as triangular mesh of landmark points
  - Uses pixel color to enhance model accuracy
- Seven expression classes: angry, disgust, fear, happy, neutral, sad, surprise

Fig. 5. Visage expression classification on the iPhone 4
Implementation

- Prototyped on iPhone 4 in C and Objective C
- Downsampling images
  - Minimal performance penalty
- Drops oldest frames when processing cannot keep up with video stream

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>640 x 480</td>
<td>4090</td>
</tr>
<tr>
<td>480 x 360</td>
<td>2123</td>
</tr>
<tr>
<td>320 x 240</td>
<td>868</td>
</tr>
<tr>
<td>192 x 144</td>
<td>298</td>
</tr>
<tr>
<td>160 x 120</td>
<td>203</td>
</tr>
<tr>
<td>96 x 72</td>
<td>68</td>
</tr>
<tr>
<td>80 x 60</td>
<td>53</td>
</tr>
</tbody>
</table>

*Table 1. Computational costs of face detection*
CPU and Memory Benchmarks

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Avg. CPU usage</th>
<th>Avg. memory usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI only</td>
<td>&lt; 1%</td>
<td>3.18 MB</td>
</tr>
<tr>
<td>Pose estimation</td>
<td>58%</td>
<td>6.07 MB</td>
</tr>
<tr>
<td>Expression inference</td>
<td>29%</td>
<td>4.57 MB</td>
</tr>
<tr>
<td>Pose estimation &amp; expression inference</td>
<td>68%</td>
<td>6.28 MB</td>
</tr>
</tbody>
</table>

**Table 2.** CPU and memory usage under various tasks

<table>
<thead>
<tr>
<th>Component</th>
<th>Average processing time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face detection</td>
<td>53</td>
</tr>
<tr>
<td>Feature points tracking</td>
<td>32</td>
</tr>
<tr>
<td>AAM fitting</td>
<td>92</td>
</tr>
<tr>
<td>Facial expression classification</td>
<td>3</td>
</tr>
</tbody>
</table>

**Table 3.** Processing time benchmarks
Accuracy of Head Pose Estimation

- Tested with angles from -90 to 90 degrees in increments of 15 degrees
- Mean absolute error = $5.51° \pm 1.9°$

**Fig. 6.** Images captured by the front-facing camera assuming varying phone tilted angles from -90 to 90 degrees, separated by an angle of 15 degrees. The red boxes indicate the detection results. The first row is detected by the standard Adaboost face detector. The second row is detected by Visage’s detector.
Fig. 7. Phone motion and head pose estimation errors (a) without motion-based reinitialization, and (b) with motion-based reinitialization
Accuracy of Facial Expression Classification

![Graph showing head pose estimation error](image)

**Fig. 8.** Head pose estimation error

<table>
<thead>
<tr>
<th>Expressions</th>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Happy</th>
<th>Neutral</th>
<th>Sadness</th>
<th>Surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>82.16</td>
<td>79.68</td>
<td>83.57</td>
<td>90.30</td>
<td>89.93</td>
<td>73.24</td>
<td>87.52</td>
</tr>
</tbody>
</table>

**Table 4.** Facial expression classification accuracy using the JAFFE dataset
Applications: Streetview+

- App tracks user’s head rotation and GPS location to provide a panorama view from Google Streetview
- 12-15 frames per second

Fig. 9. Streetview+ enhanced with awareness of user head rotation
Applications: Mood Profiler

- Tracks user’s mood in the background while they use other apps
- One classification per second

![Histograms for YouTube and Email](image)

**Fig. 10.** Expression histogram when using (a) the YouTube mobile application and (b) an email client.
Conclusions

- Visage is able to track head position and facial expressions
- Carries out all sensing and classification tasks on the phone without server
- Relatively low computational cost
References

● Visage: A Face Interpretation Engine for Smartphone Applications