EyePhone

Activating Mobile Phones With Your Eyes

Felix Nwaobasi
CS525: Mobile & Ubiquitous Computing

Mobile Interactions

- Lots of different ways to measure user gestures on a phone (accelerometers, cameras, etc)
- Touchscreen was a major advancement
- Can we go further?

HCI vs HPI

HCI

- Tends to focus on ideal settings
- Takes full advantage of the a computer's computational power
- External sensors used, often encouraged

HPI

- **Deals with users on the move under different contexts & conditions**
- **Huge concern regarding energy consumption and computational power**
- **External sensors frowned upon**

Creating the EyePhone

EyePhone Features

- Maps a user's eye movement to a position on the display
- Blinks from the user correspond to a click
- Hands-free!

The EyePhone Algorithm

- Eye detection
- Eye template creation
- Eye tracking
- Blink detection

Eye Detection

- **Uses motion analysis operations on consecutive frames**
- **Focused on eye contours**

Eye Detection (cont.)

Figure 1: Left figure: example of eye contour pair returned by the original algorithm running on a desktop with a USB camera. The two white clusters identify the eye pair. Right figure: example of number of contours returned by EyePhone on the Nokia N810. The smaller dots are erroneously interpreted as eye contours.

Eye Template Creation

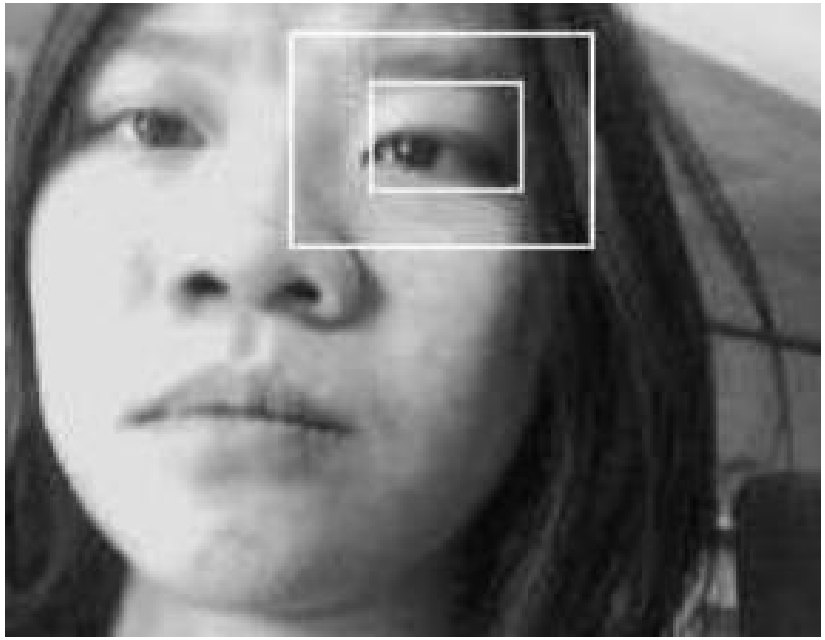
- Used whenever eye gets lost
- Created first time user uses EyePhone
- Reduces computation time & preserves battery life
- Not very effective if lighting changes

Eye Tracking

- **Based on eye template matching**
- **Correlation score between search window and open eye template**
- **Use correlation coefficient to improve accuracy**

Eye Tracking (cont.)

- **Correlation coefficient of .4 works great for eye template matching**



Blink Detection

- Uses thresholding
- Created four different thresholds to account for phone's bad quality camera
- If correlation coefficients are within these thresholds, then eye is closed

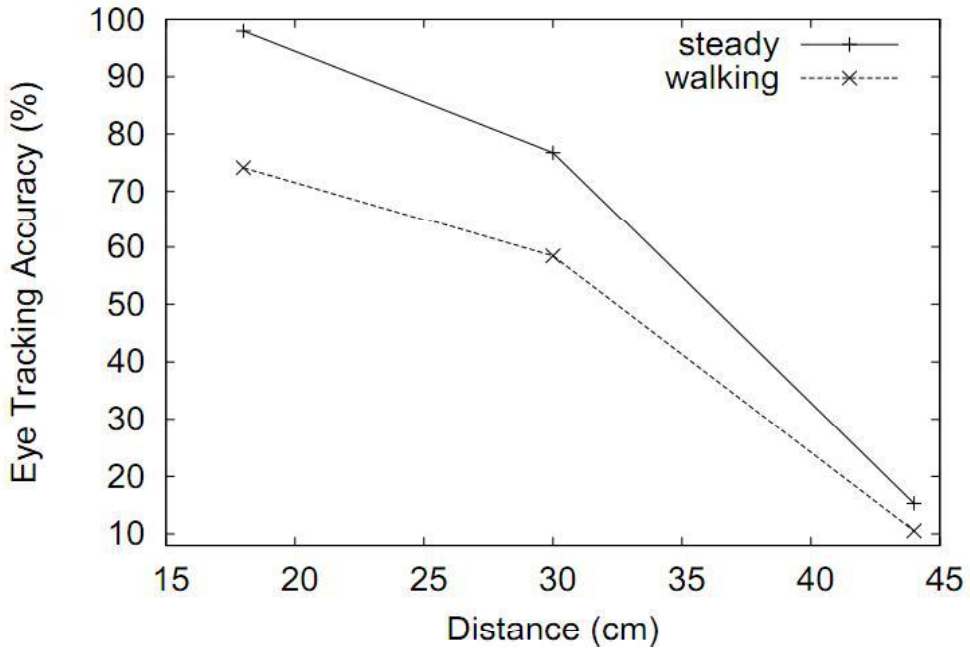
Evaluation

Experiments

- **Daylight Exposure/Stationary Subject**
- **Artificial Light/Stationary Subject**

Experiments

- **Daylight Exposure/Person Walking**
- **Distance/Tablet correlation**



Experiment Results

Table 1: EyePhone average eye tracking accuracy for different positions of the eye in different lighting and movement conditions and blink detection average accuracy. Legend: DS = eye tracking accuracy measured in daylight exposure and being steady; AS = eye tracking accuracy measured in artificial light exposure and being steady; DM = eye tracking accuracy measured in daylight exposure and walking; BD = blink detection accuracy in daylight exposure.

Eye position	DS	AS	DM	BD
Top left	76.73%	74.50%	82.81%	84.14%
Top center	79.74%	97.78%	79.16%	78.47%
Top right	80.35%	95.06%	60%	82.17%
Middle left	98.46%	97.19%	70.99%	74.72%
Middle center	99.31%	84.09%	76.52%	79.55%
Middle right	99.42%	75.79%	65.15%	80.1%
Bottom left	98.36%	93.22%	78.83%	74.53%
Bottom center	90.76%	71.46%	85.26%	67.41%
Bottom right	84.91%	93.56%	78.25%	72.89%

EyePhone & HPI

- Lightweight application
- Camera obtained 15 frames/sec
- Application only runs when user is looking at display
- Three hours of battery life if used continuously

EyePhone & HPI (cont.)

Table 2: Average CPU usage, RAM usage, and computation time for one video frame. The front camera supports up to 15 frames per second. The last column reports the percentage of used battery by EyePhone after a three hour run of the system.

CPU	RAM	Computation time	Battery used after 3h
65.4%	56.51%	~100 msec	40%

Possible Applications

EyeMenu

- Maps eye position to 1 of 9 buttons
- Blink to click!
- Great for people with disabilities

Future Work

- Increase battery life
- Improve eye template creation
- Minimize false positives

Questions?