MobileASL: Making Cell Phones Accessible to the Deaf Community

Richard Ladner

University of Washington

MobileASL

American Sign Language (ASL)

- ASL is the preferred language for about 500,000 - 1,000,000 Deaf people in the U.S and most of Canada.
- ASL is not a code for English
- Signs usually occur within the “sign-box”
- Composed of location, orientation, shape of hands and arms + facial expressions
- Usually uses 2 hands, but one-handed signing not uncommon

Current Technology for Deaf People (text)

TTY

Sidekicks and Blackberries
(text, pictures, non-real-time video)

Benefits:

Low bandwidth
Mobile (PDAs)

Problems:

English, not ASL

Current Technology for Deaf People (video phones)

Set-top boxes

Web cams

Photo / Peter Thompson

<u>Benefits:</u>

ASL, not English

<u>Problems:</u>

Requires high bandwidth

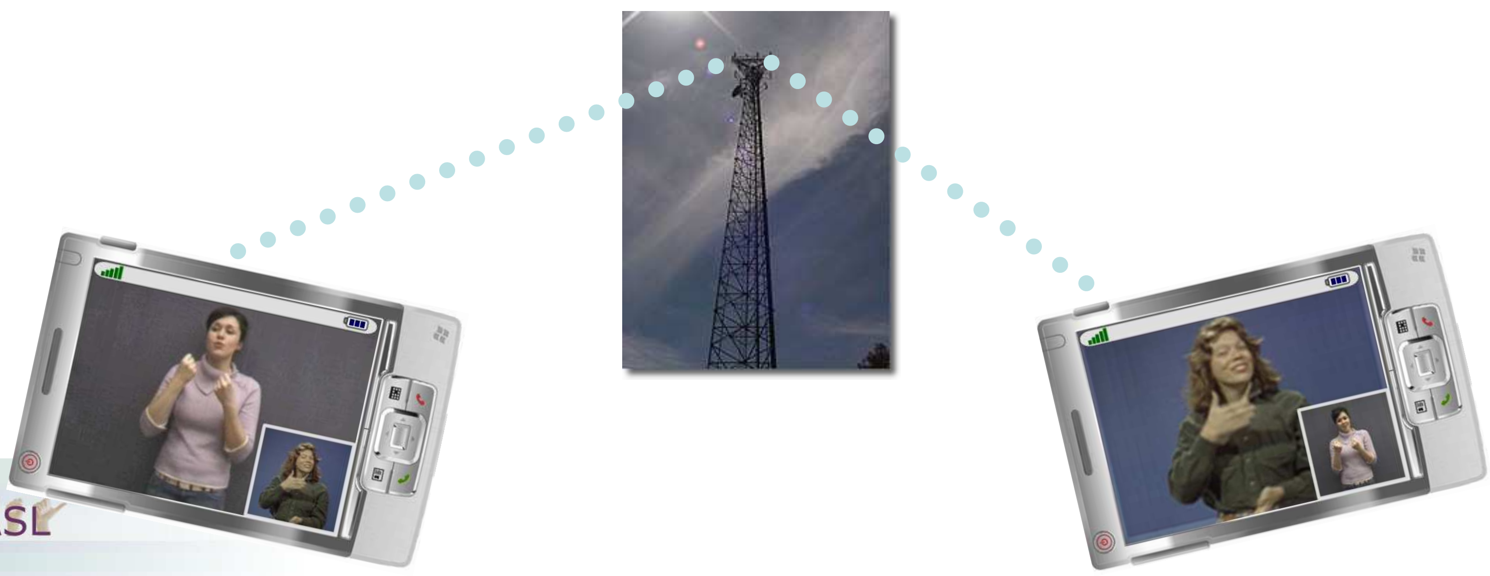
Not mobile

Our goal:

- ASL communication using video cell phones over current U.S. cell phone network

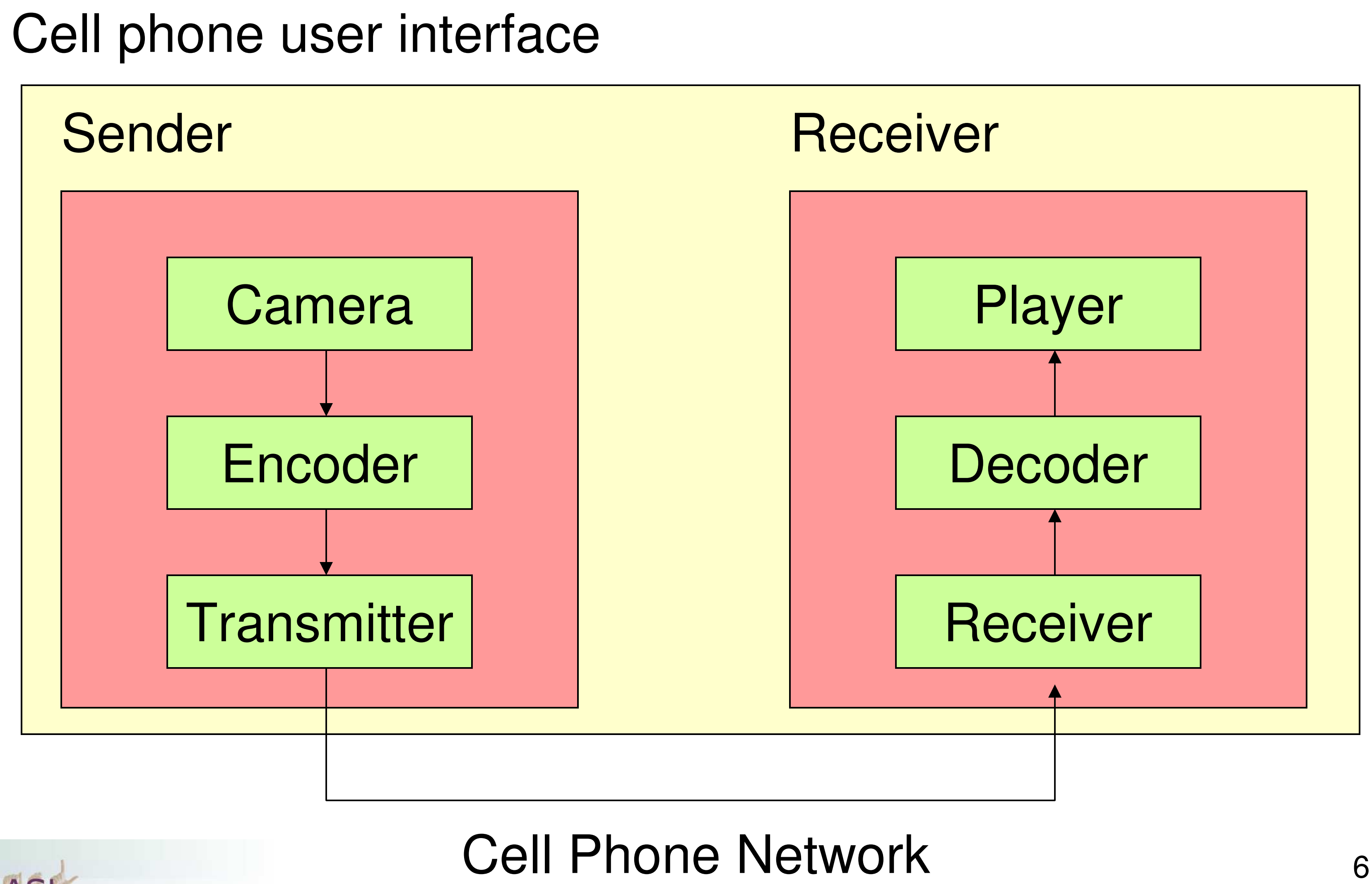
Challenges:

- Limited network bandwidth
- Limited processing power on cell phones



MobileASL

Architecture



MobileASL

Cell Phone Network Constraints

- MobileASL is about fair access to the current network
  - As soon as possible, no special accommodations
- Low bit rate constraint
  - GPRS - Ranges from 30kbps to 80kbps (download)
- Low Power
  - Cell phones run at much lower Hz then PCs
- New mobile broadband services
  - Higher bandwidth for download, not upload.

MobileASL

What about 3G?

MobileASL

Portrait

- Special Codec from Microsoft Asia
- Low Bandwidth, Low Power, small size video (160 x 120)
- May not be suitable for sign language

Keman Yu, Jiangbo Lv, Jiang Li and Shipeng Li, 2003

Codec Used: x264*

- Open source implementation of H.264 standard
- Doubles compression ratio over MPEG2
- x264 offers faster encoding
- Main profile
- Off-the-shelf H.264 decoder can be used

*The code is written from scratch by Laurent Aimar, Loren Merritt, Eric Petis, Min Chen, Justin Clay, Mans Rullgard, Radek Czyz, Christian Heine, Alex Izvorski, and Alex Wright. It is released under the terms of the GPL license.

Outline

- Motivation
- Introduction
- User Studies
- Rate, distortion, complexity optimization
- X264 implementation
- User Interface
- Current and future research

MobileASL Focus Group

- 4 Deaf people, mid-20s to mid-40s,
- Open ended questions:
  - Physical Setup
    - Camera, distance, …
  - Features
    - Compatibility, text, …
  - Privacy Concerns
    - ASL is a visual language
  - Scenarios
    - Lighting, driving, relay services, …

Implications of Focus Group

- “I don’t foresee any limitations. I would use the phone anywhere: the grocery store, the bus, the car, a restaurant, … anywhere!”
- There is a need within the Deaf Community for mobile ASL conversations
- Existing video phone technology (with minor modifications) would be usable

Eyetracking Studies

- Participants watched ASL videos while eye movements were tracked

- Important regions of the video could be encoded differently

* Muir et al. (2005) and Agrafiotis et al. (2003)

MobileASL

Eyetracking Results

- 95% of eye movements within 2 degrees visual angle of the signer's face (**demo**)
- Implications: Face region of video is most visually important
  - Detailed grammar in face requires foveal vision
  - Hands and arms can be viewed in peripheral vision

* Muir et al. (2005) and Agrafiotis et al. (2003)

Mobile Video Phone Study

- 3 Region-of-Interest (ROI) values
- 2 Frame rates, frames per second (FPS)
- 3 different Bit rates
  - 15 kbps, 20 kbps, 25 kbps

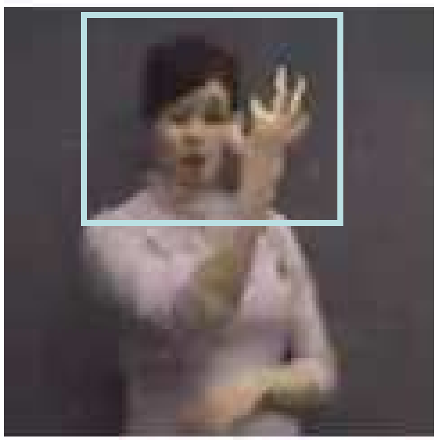
- 18 participants (7 women)
  - 10 Deaf, 5 hearing, 3 CODA*
  - All fluent in ASL

* CODA = (Hearing) Child of a Deaf Adult

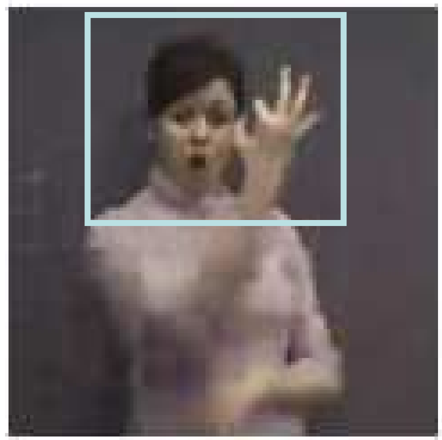
MobileASL

Example of ROI

Varied quality in fixed-sized region around the face

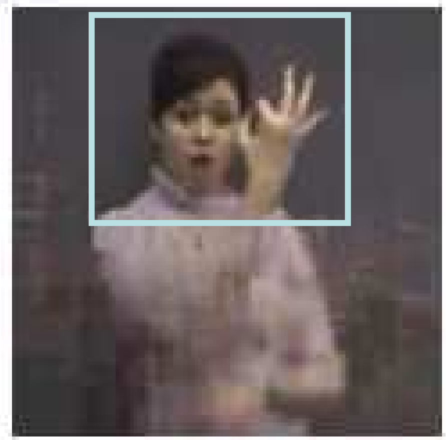


0 ROI



6 ROI

2x quality in face



12 ROI

4x quality in face

- (demo)

Examples of FPS

- Varied frame rate: 10 fps and 15 fps
- For a given bit rate:

  Fewer frames = more bits per frame

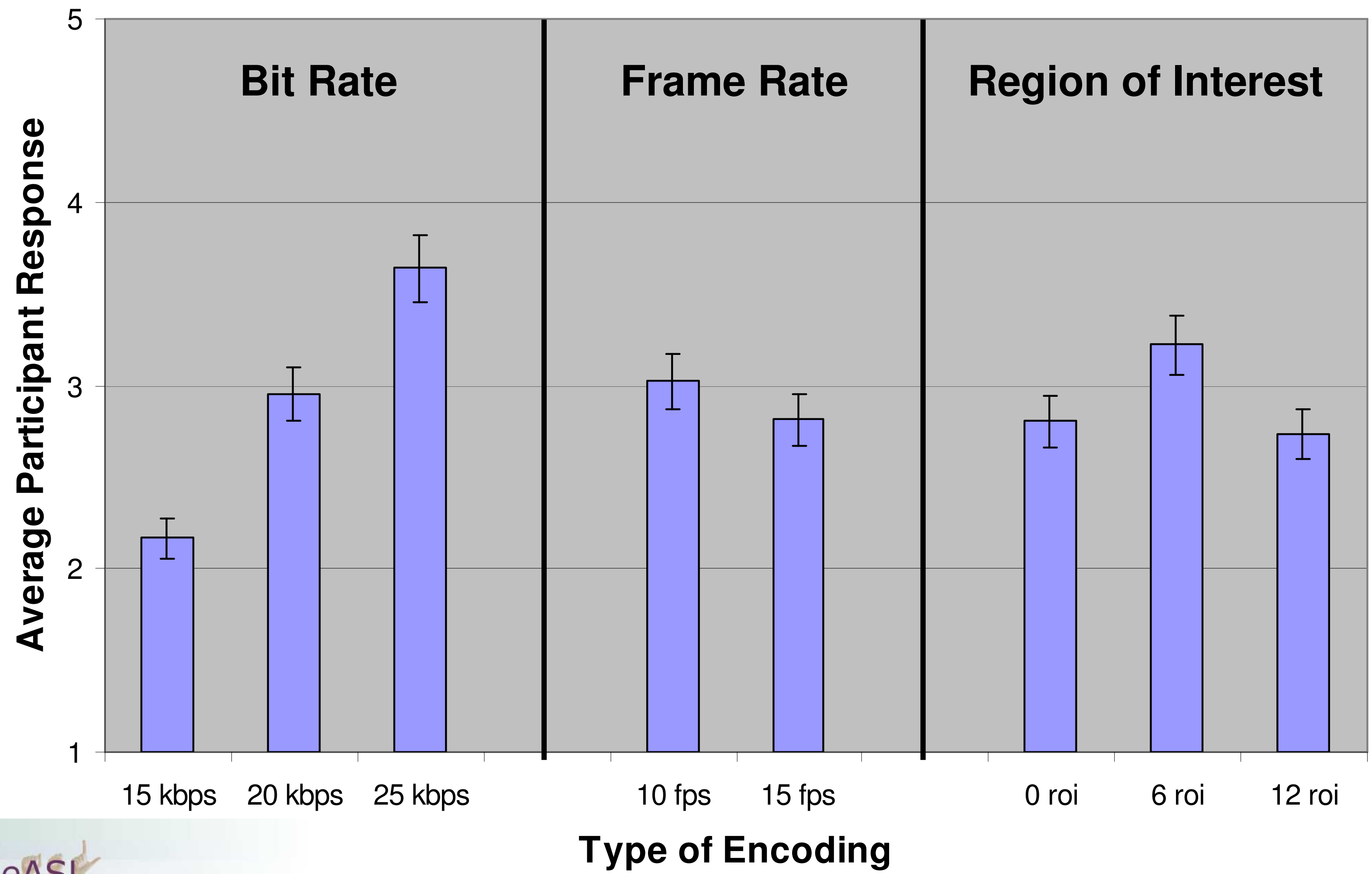
- (demo)

MobileASL

Questionnaire

MobileASL

User Preferences Results



MobileASL

Implications of results

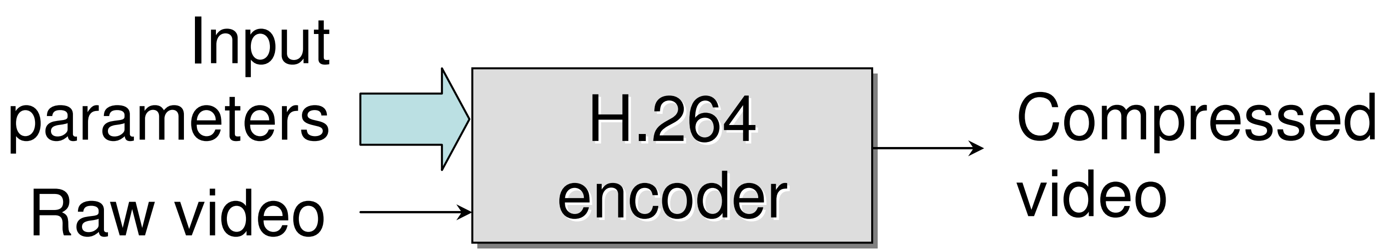
- A mid-range ROI was preferred
  - Optimal tradeoff between clarity in face and distortion in rest of "sign-box"
- Lower frame rate preferred
  - Optimal tradeoff between clarity of frames and number of frames per second
- Results independent of bit rate

Outline

- Motivation
- Introduction
- User studies
- Rate, distortion, complexity optimization
- X264 implementation
- User Interface
- Current and future research

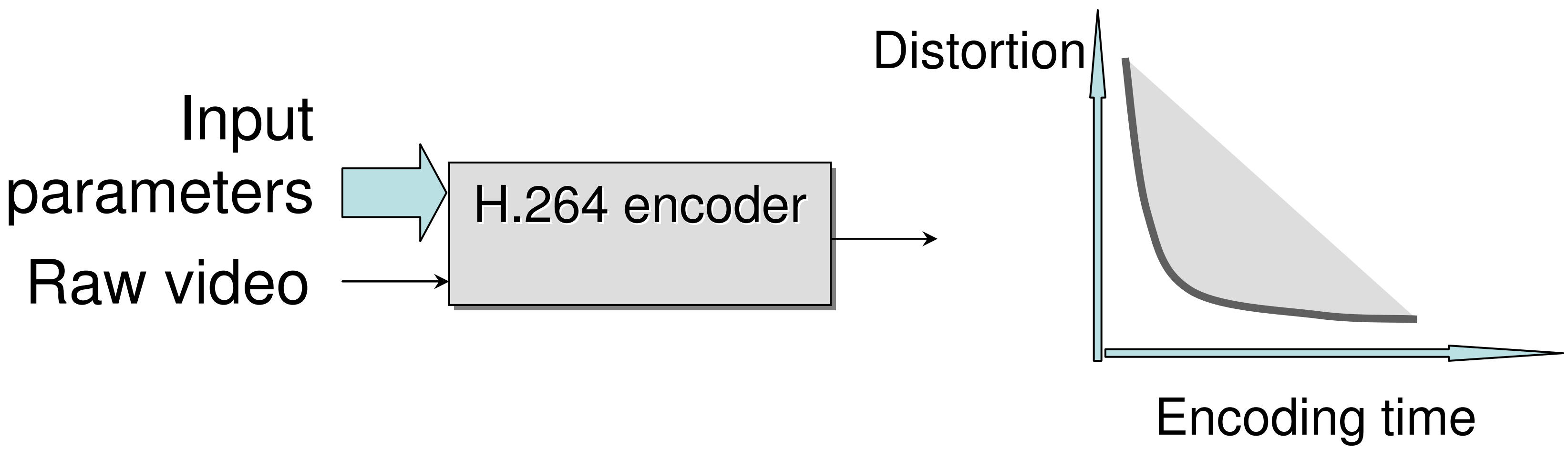
MobileASL

Rate, distortion and complexity optimization



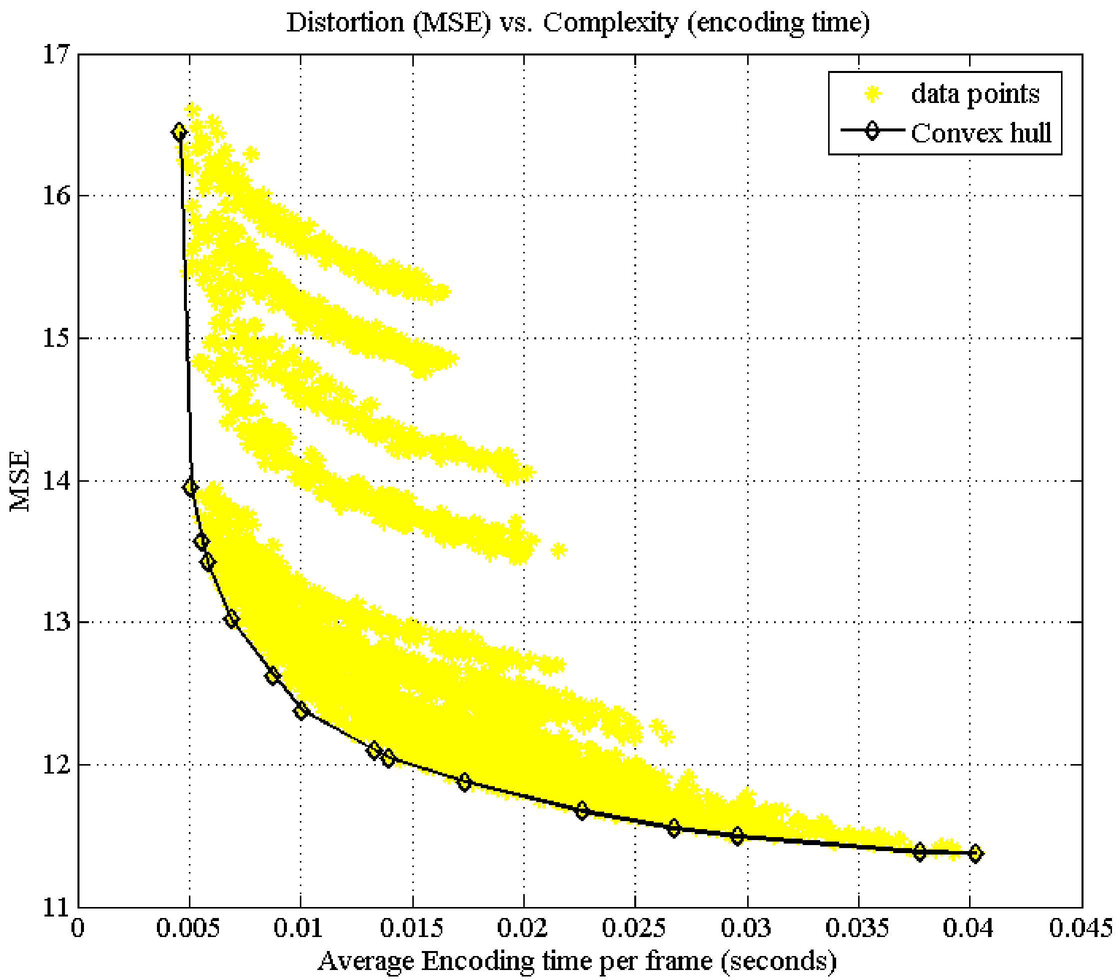
- Objective: Achieve best possible quality for least encoding time at a given bitrate

Parameter Settings



input parameters	# of options
# of reference frames	16
motion estimation	7
partition size	10
quantization method	3
Total = 16x7x10x3 =	**3360** tests/video clip

Time – Complexity Tradeoff



30 kbps
10 ASL videos

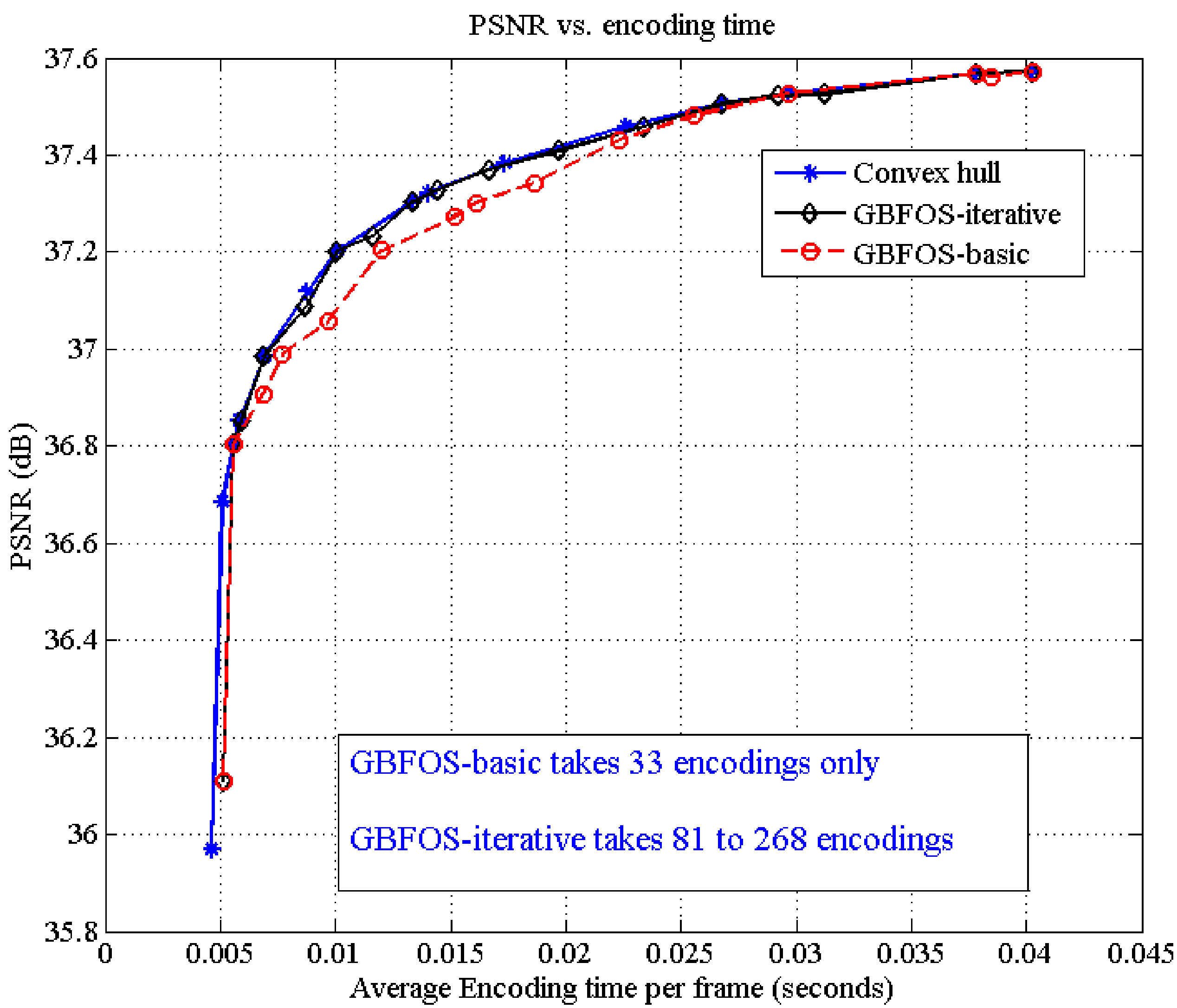
GBFOS Approach

Chou, Lookabaugh, Gray, 1989

- Choose input parameter that minimizes the slope on the convex hull and repeat.
- Parameter settings are not independent.
- Basic – Compute slopes once.
- Iterative – Recompute slopes after each parameter is chosen.

MobileASL

PSNR vs. Average Encoding Time



MobileAS

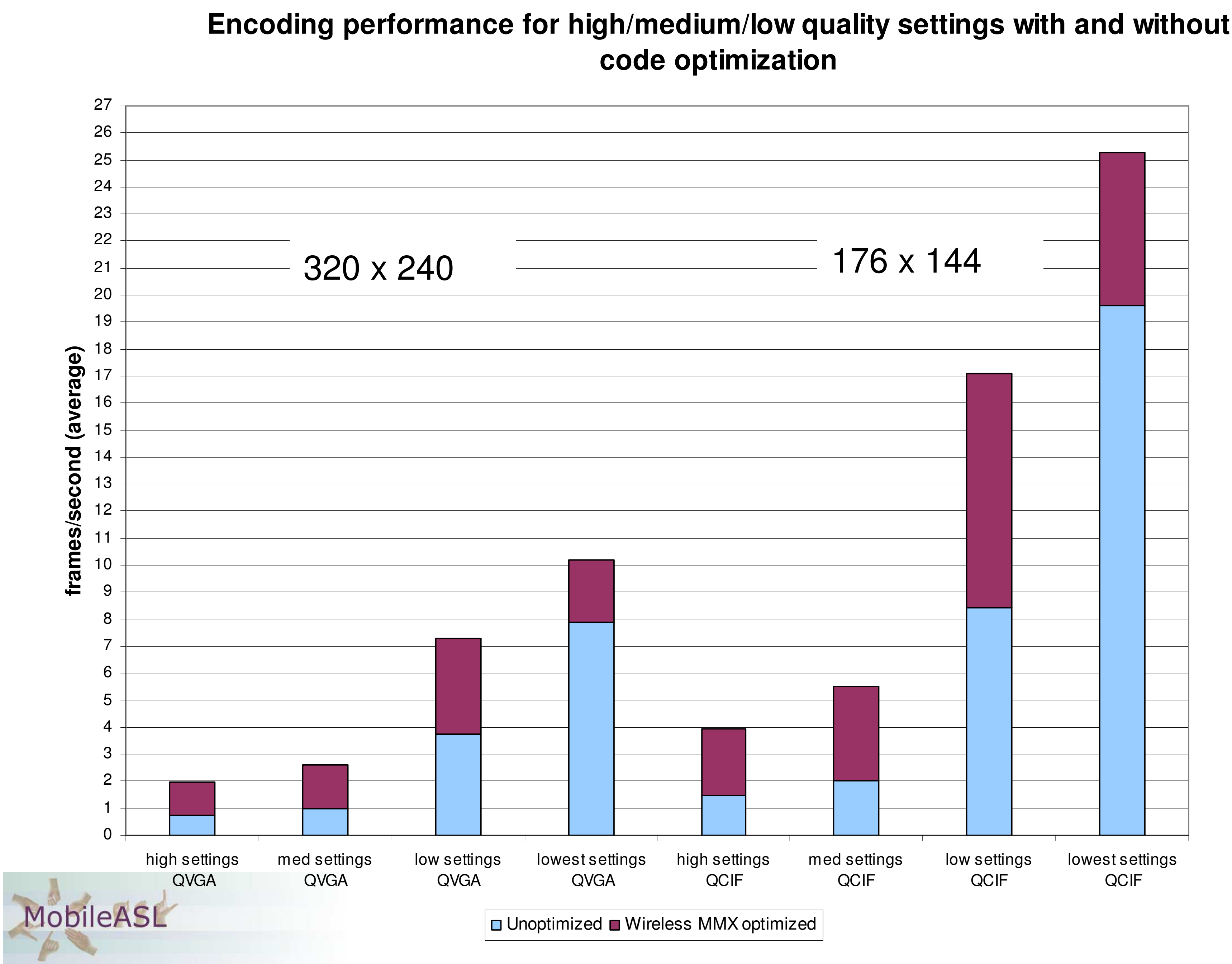
Outline

- Motivation
- Introduction
- User studies
- Rate, distortion, complexity optimization
- X264 implementation
- User Interface
- Current and future research

MobileASL

Encoding/Decoding on the Cell Phone

- Implemented a command-line version of x264 on a cell phone using Windows Mobile Edition 5.0.
- Required significant modifications to the Linux based x264 codec.


Encoding performance for high/medium/low quality settings with and without code optimization
frames/second (average)
320 x 240
176 x 144
high settings QVGA
med settings QVGA
low settings QVGA
lowest settings QVGA
high settings QCIF
med settings QCIF
low settings QCIF
lowest settings QCIF
Unoptimized
Wireless MMX optimized
MobileASL

Examples of Low Frame Rates

- Demo

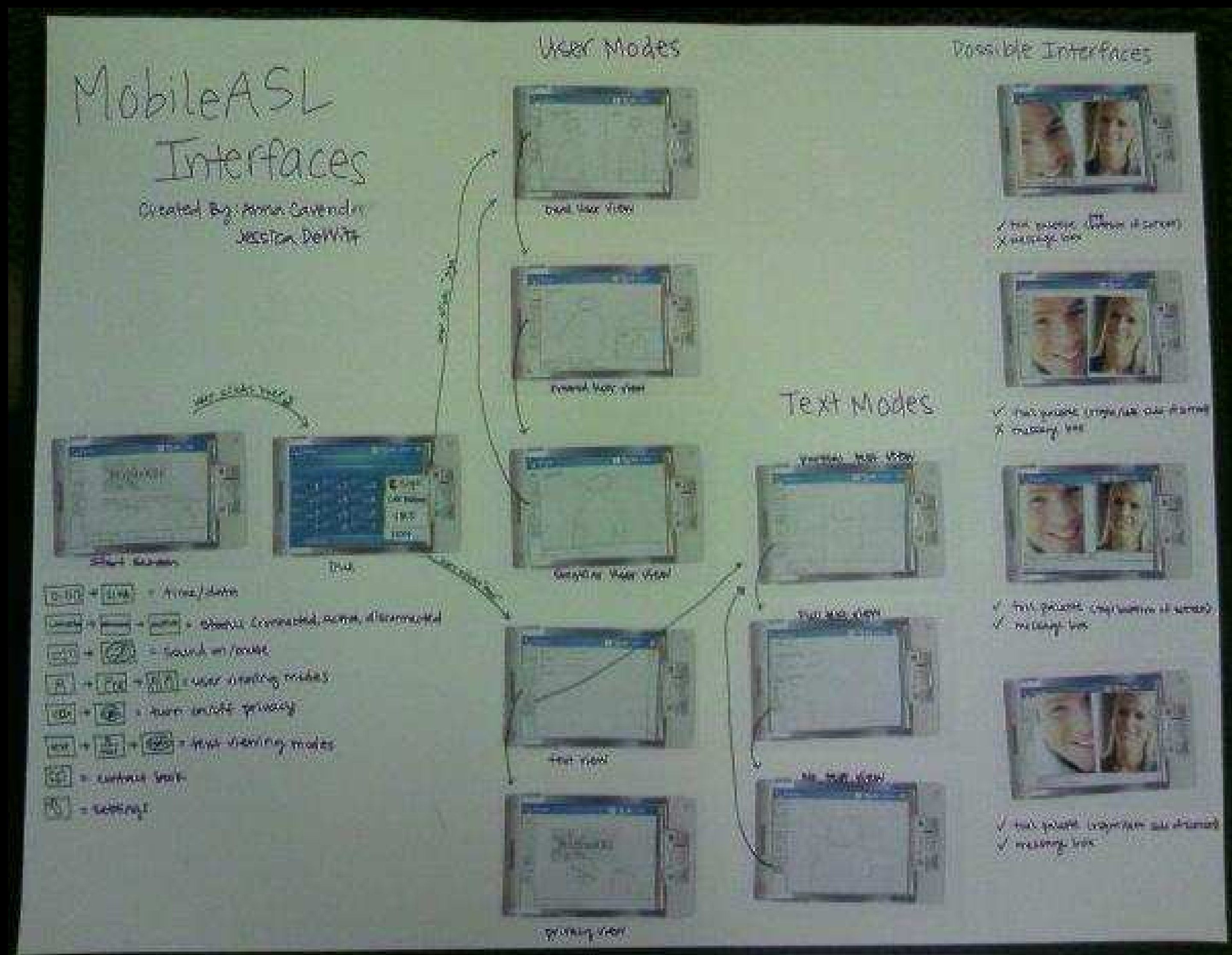
Outline

- Motivation
- Introduction
- User studies
- Rate, distortion, complexity optimization
- X264 implementation
- User Interface
- Current and future research

User Interface Design: Goals

- Usable, intuitive, easy to learn
- Inspired by Deaf users
- Utilize existing knowledge (VP, Webcam, Sorenson …)
- Design stages:
  - Story boards
  - Paper prototype testing
  - Digital prototyping

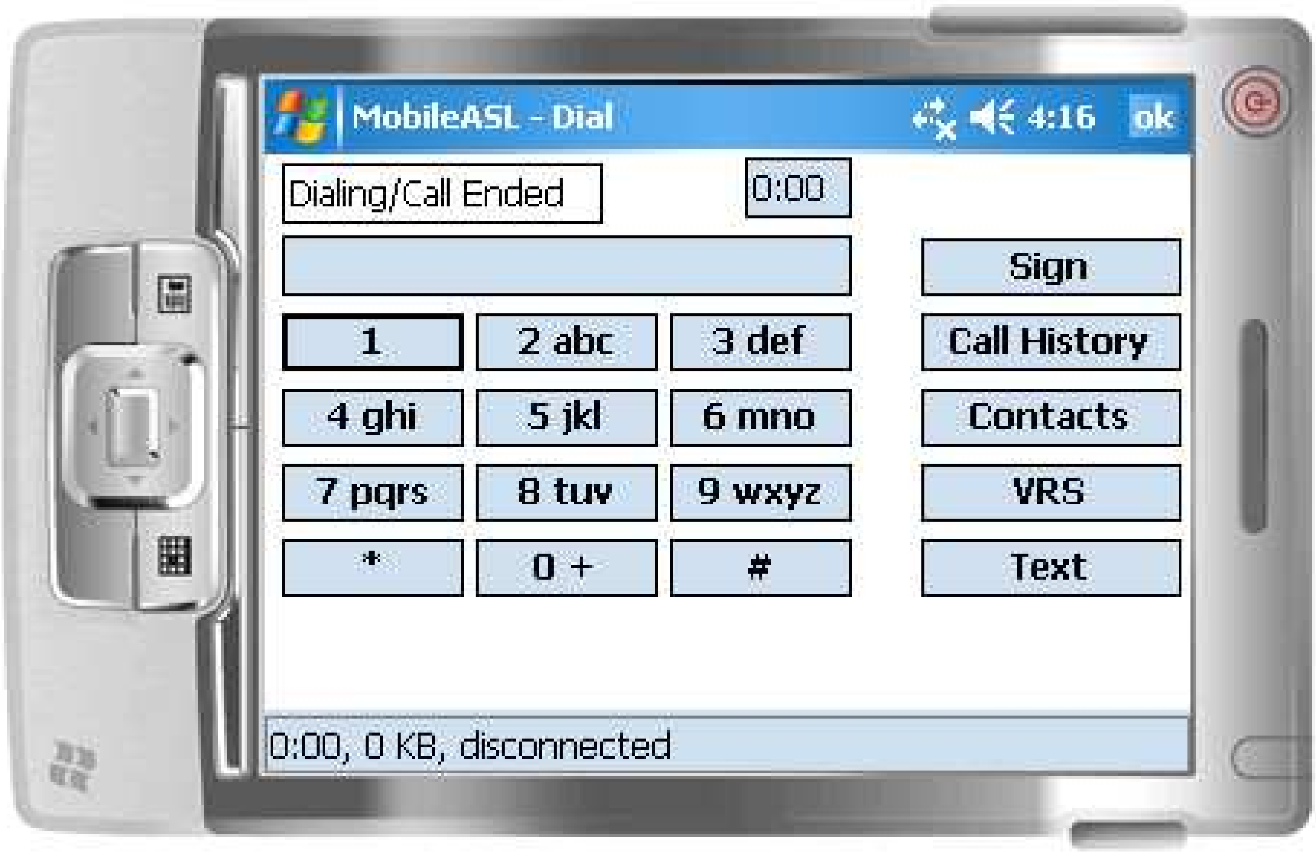
MobileASL


MobileASL Interfaces
Created By: Anna Cavender
Jessica DeWitt
User Modes
Possible Interfaces
Text Modes

Basic Interface

Split Screen with Text

Call Set-up



MobileASL

Outline

- Motivation
- Introduction
- User studies
- Rate, distortion, complexity optimization
- X264 implementation
- User Interface
- Current and future research

MobileASL

Current Work

- Dynamic Region-of-Interest
  - Skin detection algorithms
- Objective Metrics
  - For ASL Understandability
- Activity Recognition
  - Fingerspelling, signing, "listening"
- Building the System
  - Transmission, Receiving, Playing
  - Packet loss on GPRS

MobileASL

Dynamic Region of Interest

- Use skin detection algorithms to drive region of interest.
- Fast skin detection algorithms exist
- Demo

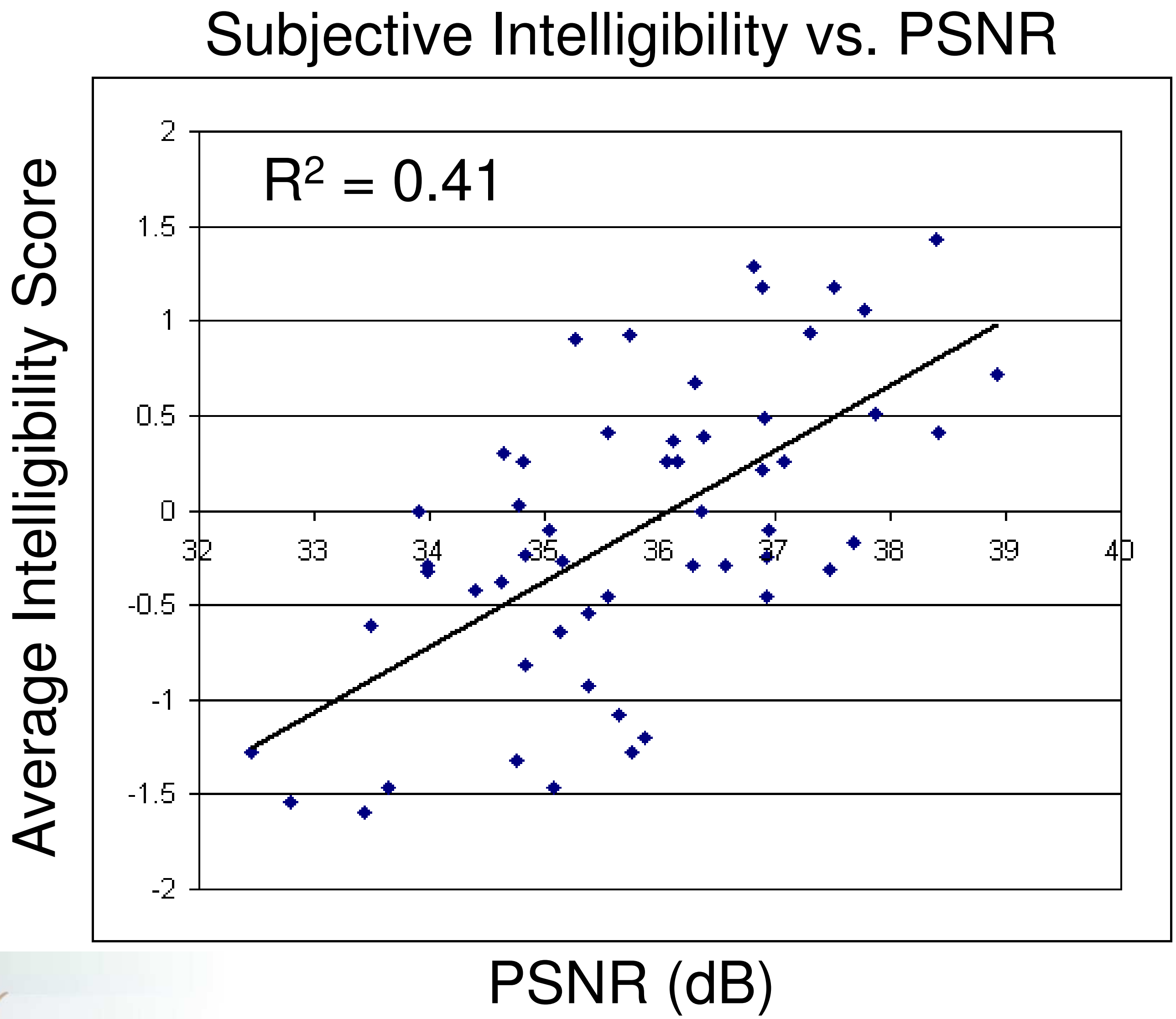
MobileASL

Objective Metric

- Importance
  - Face
  - Hands
  - Signing Box
- Weighted MSE based on where the pixels are

MobileASL

Objective Intelligibility Metric



MobileASL

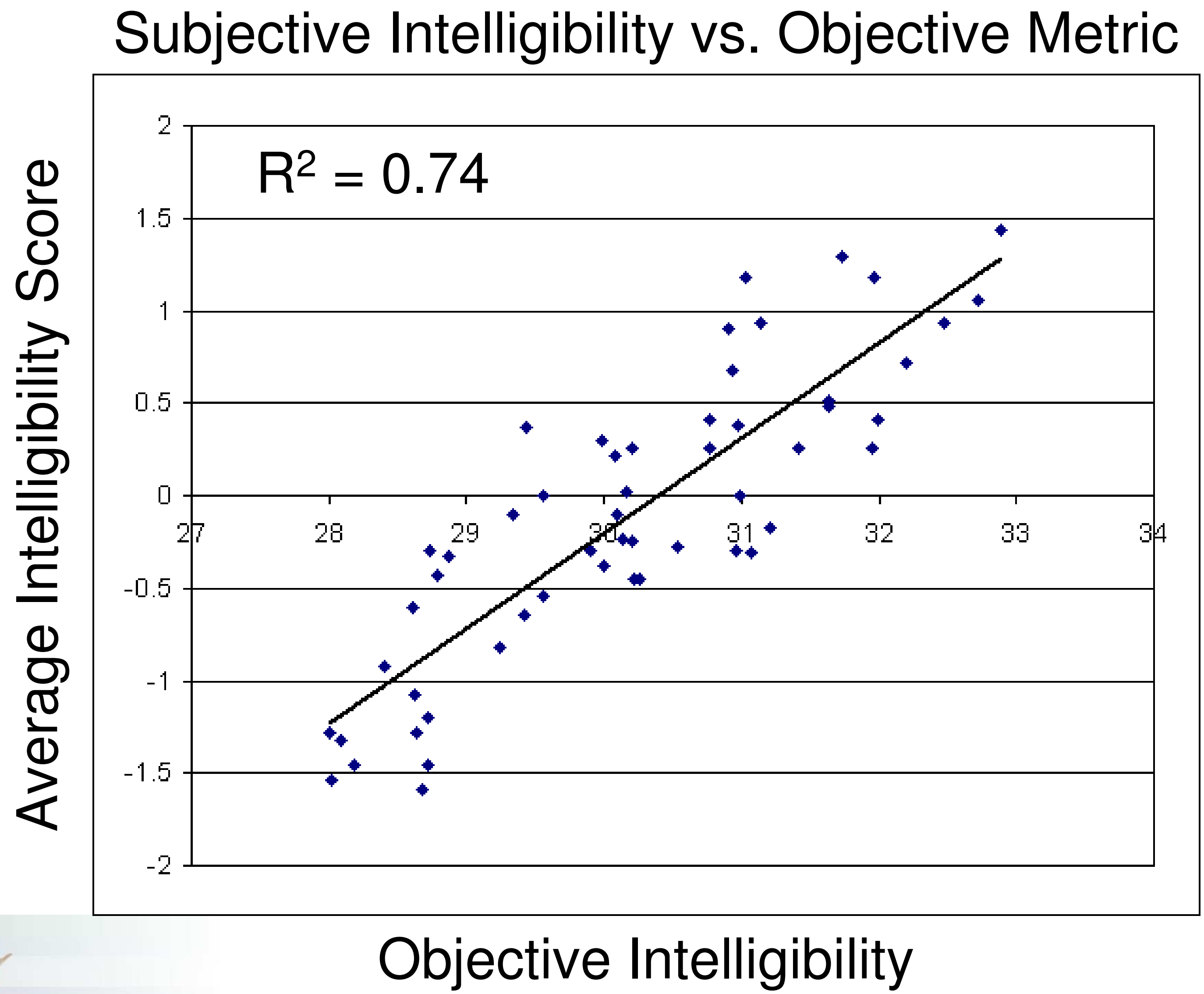
Objective Intelligibility Metric

$$I = 10\log_{10}\frac{255^2}{F \times MSE_F + H \times MSE_H}$$

where $F = 0.6$ and $H = 0.4$

MobileASL

Objective Intelligibility Metric



MobileASL

Activity Recognition

- Motivation:
  - Finger spelling requires a higher bit rate and/or frame rate for intelligibility than signing
  - We want to minimize encoding complexity when not signing.
- Goal:
  - Recognize these three states: finger spelling, signing, not signing
  - Perform recognition in real time

MobileASL

Possible Solution

- Use H.264 motion vectors as features
- Use probabilistic techniques to automatically recognize activity
  - Hidden Markov Models
  - Kalman filters or particle filters

MobileASL

Building the System

- in C#:
  - Really easy to develop GUIs.
  - Developers can only use their predefined interface for the camera. The interface is simple, but extremely limited.
- In C++:
  - GUI development much more complex.
  - Accessing camera requires knowledge of windows COM system.

MobileASL

Thanks

- Co-PIs
  - Eve Riskin and Sheila Hemami
- Graduate Students
  - Anna Cavender, Rahul Vanam, Neva Cherniavsky, Frank Ciaramello, Dane Barney, Carl Hartung
- Undergraduate Students
  - Jessica DeWitt, Loren Merritt, Sam Whittle
- National Science Foundation