Mobile Touch UI Optimization

Prof. Parham Aarabi
MOBILE USER INTERFACE OPTIMIZATION
What is a good mobile UI design?

1. Button size
2. Button location
3. Visual (i.e. color, font, style, ...)

MOBILE USER INTERFACE OPTIMIZATION
Novice vs. Expert Users

**Novice Users**
- Button selection is based on curiosity
- Higher touch error
- Takes time to find specific button

**Expert Users**
- Button selection is based on purpose
- Lower touch error
- No need to find buttons
Novice vs. Expert Users

Novice Users

Ideal UI is simple, instructional, with large buttons

In practice, we need one UI for both

Expert Users

Ideal UI is efficient, quick, with mid-sized buttons
What is the ideal button size?
What is the ideal button size?

Experimental results based on the evaluation of 20 individuals
What is the ideal button size?

The bigger the better?

- Smaller: Less user friendly, More efficient
- Bigger: More user friendly, Less efficient
MOBILE USER INTERFACE OPTIMIZATION

UI Design

Smaller
Less user friendly
More efficient

Bigger
More user friendly
Less efficient

Information Theory

Higher Rate
Less noise robust
More efficient

Lower Rate
More noise robust
Less efficient
Information Theoretic Framework


Information Theoretic Framework

Alternatively:

\[ \text{prob. of touch error} = ks_i \]

\[ E[\text{touches for UI } i \text{ until correct selection}] = \frac{1}{ks_i} \]

\[ E[\text{total touches}] = \sum_i \frac{p_i}{ks_i} \]

\[ \arg \min_{s_i} E[\text{total touches}] = c \sqrt{p_i} \]

\[ S = cP^{0.5} \]
MOBILE USER INTERFACE OPTIMIZATION

\[ S = c \]

\[ S = c P \]

\[ S = c P^{0.5} \]
UI Design (space limited)

Information Theory (bandwidth limit)

Rate-limited Channel
Information Theoretic View of Touch UI

A touch-screen is a communication channel with 1 of N possible touch locations

N=total touchable pixels

(i.e. a codebook with a total of N codes)

Ideally, there would be $\log_2 N$ bits of info per touch
Information Theoretic View of Touch UI

In practice, there is touch noise (i.e. intent to touch location X is measured as a touch on location Y)

If total touch-area is $S_T$, and if minimum detectable touch area is 100mm$^2$ (about the size of a finger tip), then:

$$\text{Info per touch} = \log_2 \left( \frac{S_T}{100} \right)$$
Information Theoretic View of Touch UI

Only a portion of the total touch space is UI (u% of $S_T$). We then have:

$$\text{Info per touch} = \log_2 \left( \frac{uS_T}{100} \right)$$

On iPhone 5, screen area = 4413 mm²
Assuming $u=30\%$, we get:

$$\text{Info per touch} = 3.73 \text{ bits (typical)}$$

For fun, at $u=100\%$:

$$\text{Info per touch} = 5.46 \text{ bits}$$
Information Per Touch

-Every touch selection provides some info

e.g. choosing 1/2 equiprobable buttons = 1 bit of info

e.g. choosing 1/8 equiprobable buttons = 3 bits of info

But usually, buttons are not equiprobable! How do we find the probability distribution to calculate Entropy?

$$H = - \sum_{i} p_i \log p_i$$
Information Per Touch

\[ S = cP^{0.5} \quad \Rightarrow \quad P = kS^2 \quad \Rightarrow \quad H_I = \log \sum_i s_i^2 - \frac{\sum_i s_i^2 \log s_i^2}{\sum_i s_i^2} \]

We call this “Interface Entropy”
Interface Entropy

Equation for getting amount of info per touch based on the size of the buttons

\[ H_I = \log \sum_i s_i^2 - \frac{\sum_i s_i^2 \log s_i^2}{\sum_i S_i^2} \]
Interface Entropy

Less info required of user = simpler

1.86 bits

1.86 bits

1.12 bits
Interface Entropy

Less info required of user = simpler

3.58 Bits

3.33 Bits
Visual (Image) Entropy

Measures only colorfulness of image (no spatial measurement)

\[ H_V = - \sum_{r,g,b} p(r,g,b) \log p(r,g,b) \]
App Showdown (iStanford vs. MIT Mobile)
MOBILE USER INTERFACE OPTIMIZATION

Interface Entropy:
Avg. Button Size:
Visual Entropy:
Button Placement:

Interface Entropy:
Avg. Button Size:
Visual Entropy:
Button Placement:
MOBILE USER INTERFACE OPTIMIZATION

Interface Entropy: 3.59 bits
Avg. Button Size: 120mm²
Visual Entropy: High
Button Placement: Average

Interface Entropy: 2.65 bits
Avg. Button Size: 100mm²
Visual Entropy: Low
Button Placement: Average
Fun topic 1: Dynamic Keyboards

Touch-screen keyboards are not great!
Dynamic Keyboards

If you have typed “WHIC”, there is a high probability you will type another “H”
Dynamic Keyboards

Idea: since probability of selection has changed, why not change size of “H” key
Dynamic Keyboards

Based on the current probability of selection, dynamically redraw keyboard
Dynamic Keyboards

e.g. Starting keyboard with key sizes matching selection frequencies in English
Dynamic Keyboards

- More efficient than a regular keyboard
- Can be more annoying to use
Dynamic Keyboard Example: Dasher

Dynamic Keyboard Example: Dasher
Dynamic/Smart Keyboard Example: Fleksy
Dynamic/Smart Keyboard Example: Swype
The key is error modelling on the words QwertyuiuhgCvbghjK
QuICK
wICK
...
Other example: Minuum
Fun topic 2: Extended Touch

What if you could touch outside the screen?
Fun topic 2: Extended Touch
Fun topic 2: Extended Touch
Fun topic 2: Extended Touch