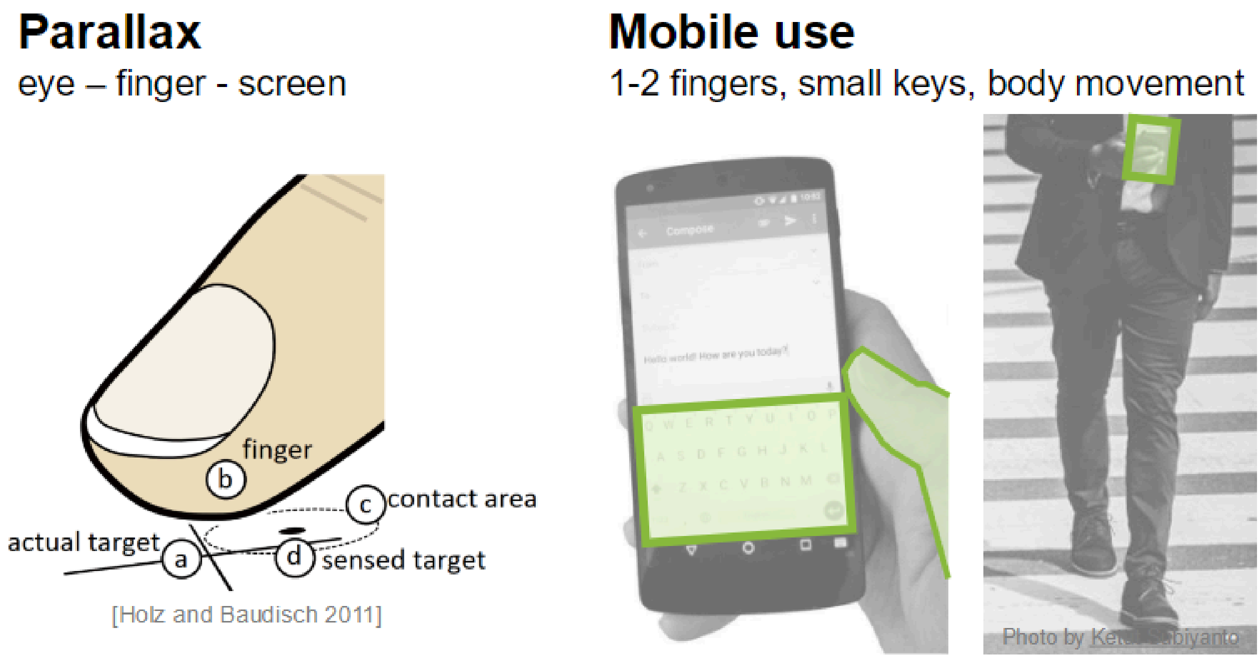
Lecture 9 - Text Entry

Mobile Text Entry

Why is typing on mobile devices difficult?



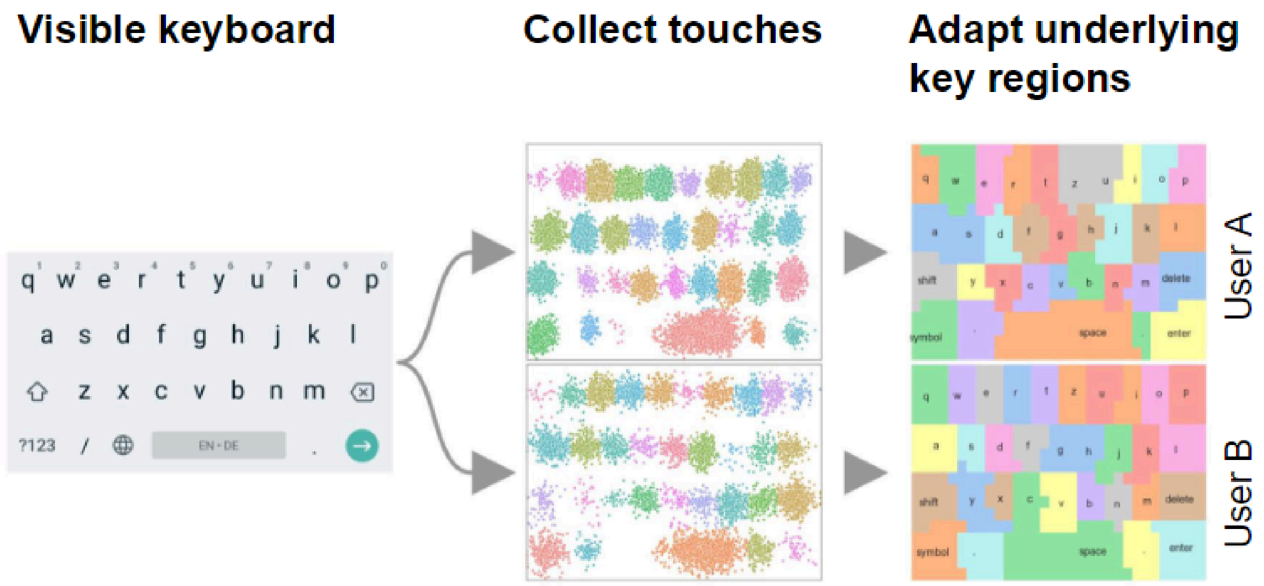
- **Parallax effect:** Misalignment between finger, screen, and eyes leads to inaccurate touches.
- **Small key sizes:** Higher chance of pressing the wrong key.
- **Mobile use conditions:** Users type in various postures (one-handed, walking, sitting), affecting accuracy.

> **Parallax and Typing Accuracy - Holz & Baudisch (2011)**
>
> "Parallax between the user's eye, finger, and screen affects the precision of touch input."

Touchscreen Keypress Variability

- Studies show that users **do not touch keys at their geometric centers**.
- Typing patterns vary by individual and context (e.g., smartphone vs. tablet).
- Gaussian distributions can model touch location variation.

Probabilistic Keyboard Models



- **Touch points follow a probabilistic distribution** around key centers.
- **Bayes' Rule** is used to infer the most likely intended key.

What is a Probabilistic Keyboard Model?

Instead of treating touches as exact, probabilistic models **assign likelihoods** to different keys based on touch input and language context.

Keyboard Adaptation Strategies

User-Specific Adaptations

Adaptation Type	Description
Touch Behavior	Adjusts key regions based on user's past typing patterns.
Context Awareness	Changes layout based on hand posture or movement (e.g., walking).
Language Model	Predicts likely words based on previous input.

Gaussian Key Model

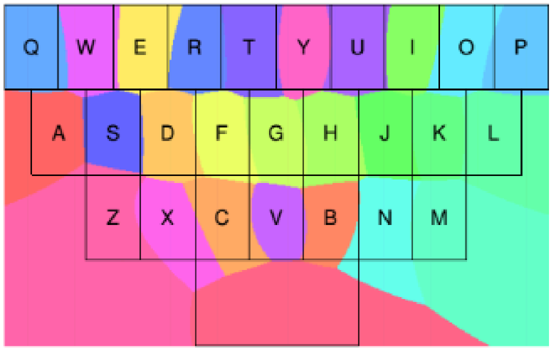
- Each key is modeled as a **Gaussian distribution**.
- The system updates key regions dynamically based on **historical touch data**.

Error Correction & Prediction

Language Model Influence

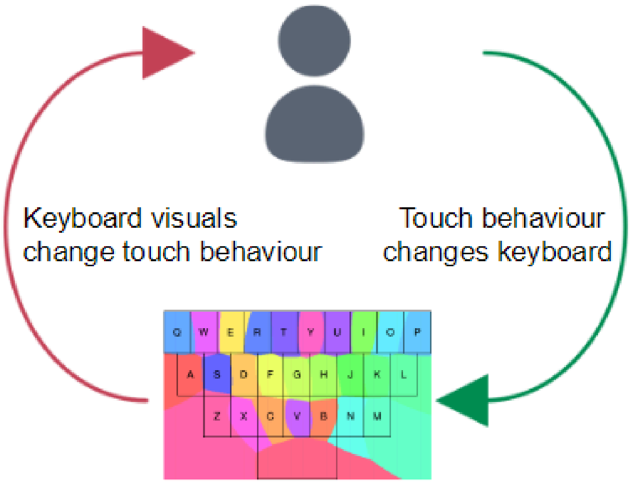
- After a key is detected, the system applies **language models (e.g., bigram models, neural networks)** to improve accuracy.
- Example: If a user types "th", the system predicts "the" over "thu".

Why do our keyboards not look like this?



[Yin et al. 2013]

→ Avoid co-adaptation of user and system

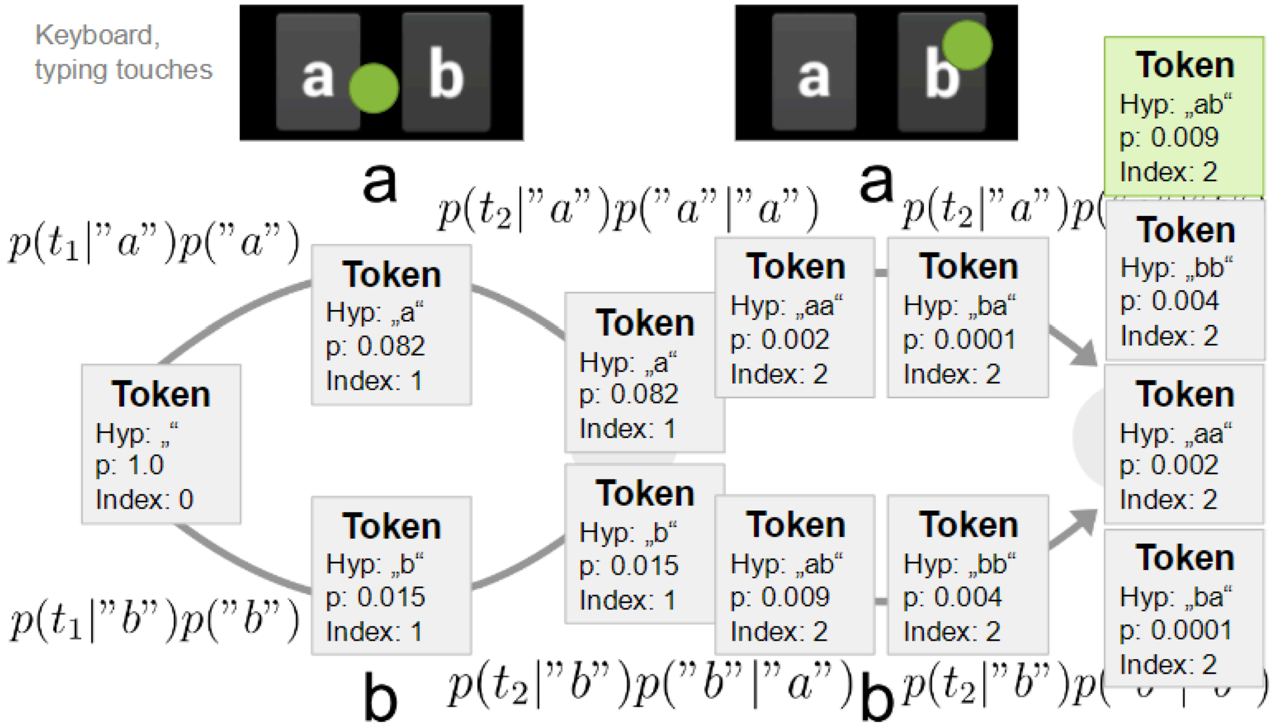


> ♨ **How Language Models Improve Typing**
>
> Even if a touch input is slightly off, language models help select the **most likely** intended word.

Decoding Typing Sequences

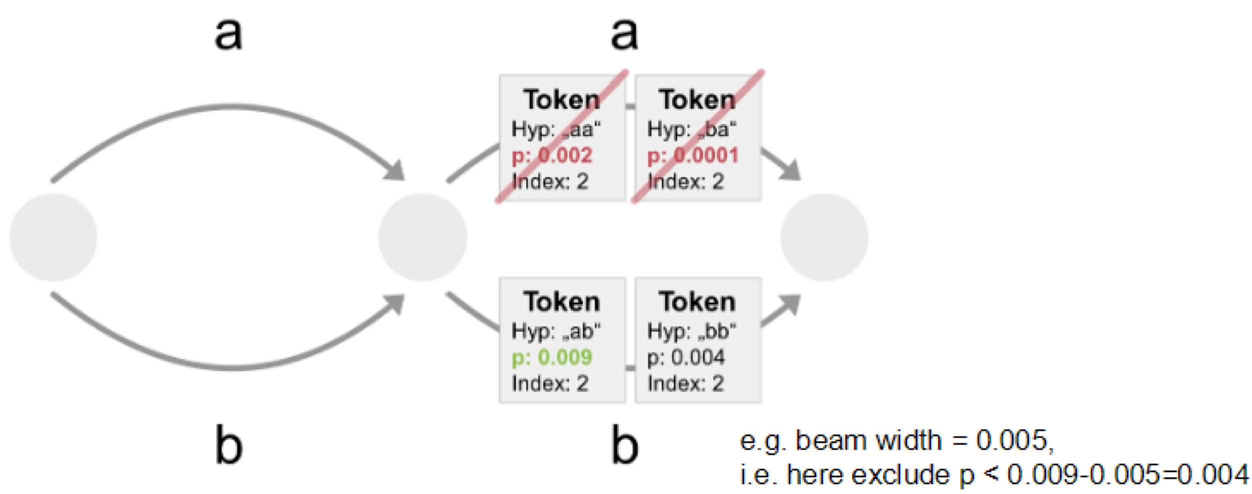
Method	Description
Token Passing	Iteratively refines possible words based on touch input.
Beam Search	Limits search space to the most probable word sequences.

Token passing algorithm



With beam search / pruning

- Problem: Large search space
  Substitution-only → exponential, Insertion → infinite
- Solution: **Beam search / pruning**
  Per index, only propagate tokens that are within a certain range (=„beam width“) of the probability of the most likely token.



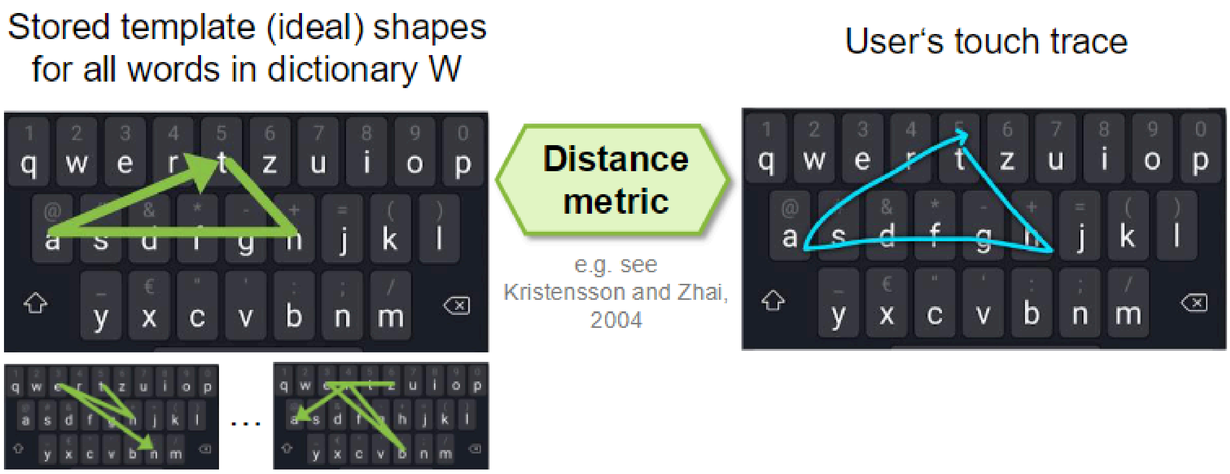
Gesture-Based Typing

- Swiping from letter to letter forms a trace that is matched to stored **word shapes**.
- Uses **distance metrics** to compare user input with predefined templates.

- Infer intended word from shape of finger trace on the keyboard

Shape model ↘ ↙ Language model

$$w' = \text{argmax}_{w \in W}(p(trace|w)p(w))$$



> **Gesture-Based Typing - Kristensson & Zhai (2004)**
>
> "Users can enter text by drawing word shapes, reducing individual key presses."

Optimization-Based Keyboard Design

Why Optimize Keyboard Layouts?

- QWERTY is **historically optimized for typewriters**, not touchscreens.
- Alternative layouts reduce **finger movement** and improve efficiency.

Optimization Strategies

Method	Description
Random Search	Generates random layouts and selects the best.
Simulated Annealing	Gradually refines layouts by accepting probabilistic changes.
Heuristic Methods	Use AI models to find the best design based on constraints.

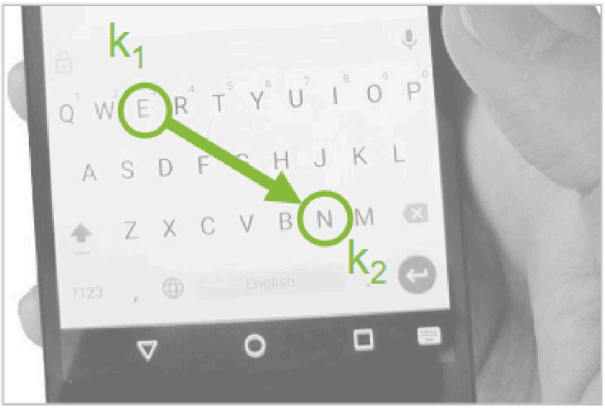
Objective Function: How to judge a layout?

- Finger movement time (e.g. Fitts' law)

$$t(k_1, k_2) = a + b\log_2\left(\frac{D}{W} + 1\right)$$

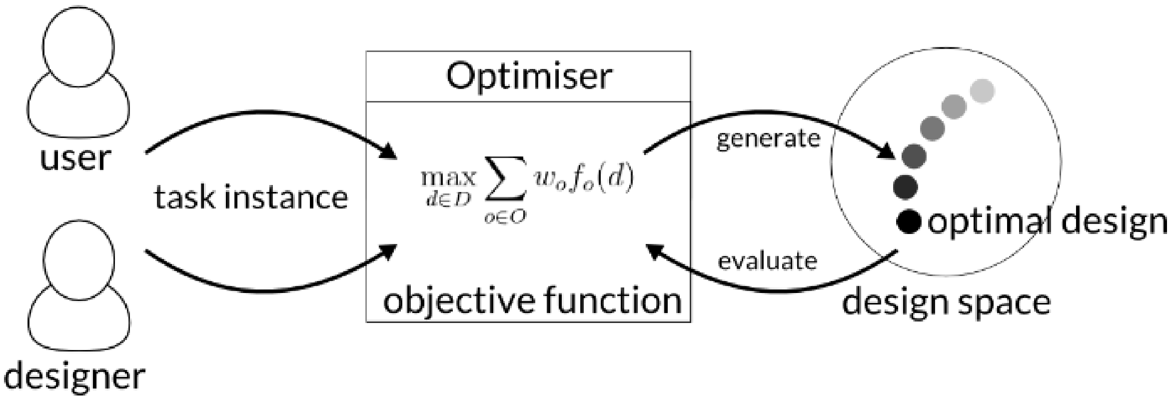
- Language properties (e.g. bigram frequencies)

e.g. $p(\text{"}n\text{"}|\text{"}e\text{"}) = 0.001$



- Combined: mean time between two key presses

$$f(d) = \sum_{k_1 \in K} \sum_{k_2 \in K} p(d(k_2)|d(k_1))\,t(k_1, k_2) \quad \text{where the design } d \text{ maps from keys to characters}$$



Simulated Annealing Example

- **Starts with a random layout**.
- Gradually refines by testing **small changes**.
- "Cools down" to **converge on an optimal design**.

Potential of Optimization-based Design

- Obtaining information on the design problem and a formal specification
- Exploring a large design space comprehensively
- Improving quality and robustness of designs
- Estimating possible improvements
- Supporting human designers
- Optimization during use, personalised UIs
- Requires: Models of user behaviour, formal problem definition / objective function, computational capacity, …

Why is QWERTY Still Used?

Even though optimized layouts exist, users are **resistant to change** due to familiarity.

Questions & Answers

Name and explain the key components of optimization-based UI design.

Key components include **objective functions** (define what is being optimized), **design constraints** (limit solutions to practical options), and **search algorithms** (find the best possible solution).

How can designers influence obtained designs in this approach?

Designers can adjust **constraints, weighting factors, and optimization goals** to guide the system towards practical and user-friendly solutions.

Explain Simulated Annealing. Can you consider a design resulting from this method as "optimal"? Why (not)?

Simulated Annealing is an optimization technique that explores different solutions by gradually reducing randomness in search. The result is **near-optimal**, but not guaranteed to be the absolute best.

If it is possible to find better designs than QWERTY, why are we not using them widely?

User habits, resistance to change, and infrastructure dependence make it difficult to transition away from QWERTY despite superior alternatives.

Beyond keyboard layouts, which other UI design problems could be addressed with this approach? And which are hard to address in this way?

Optimization can improve **menu layouts, gesture-based interfaces, and adaptive UI designs**. However, **social interaction and cultural usability** are harder to optimize due to subjective factors.