# Toward a universal text input method for the Ubiquitous Computing Age

*Toshiyuki Masui*

Keio University
masui@pitecan.com
http://pitecan.com/
http://3memo.com/masui/wtim

November 13, 2011

# Self introduction

- Working on user interface technologies @ Keio Univ, Japan.
- Worked at various companies and organizations
  - Fujitsu, Sharp, Sony, AIST, Apple,
- Developed various text input systems
  - For Japanese mobile phones
  - For iPhone

# Today's talk

- My experience on developing Japanese text input systems

# “POBox” on Japanese Android

# Masui's Activities

- Developing various UI systems
- Running useful Web services
- Writing papers、magazine articles、Web articles
- Developing techniques for ubiquitous computing

# Research topics

- Predictive user interface
- Information visualization
- Information retrieval
- Text input systems
- Communication systems
- Ubicomp devies
- Authentication systems
- ~~Natural language processing~~

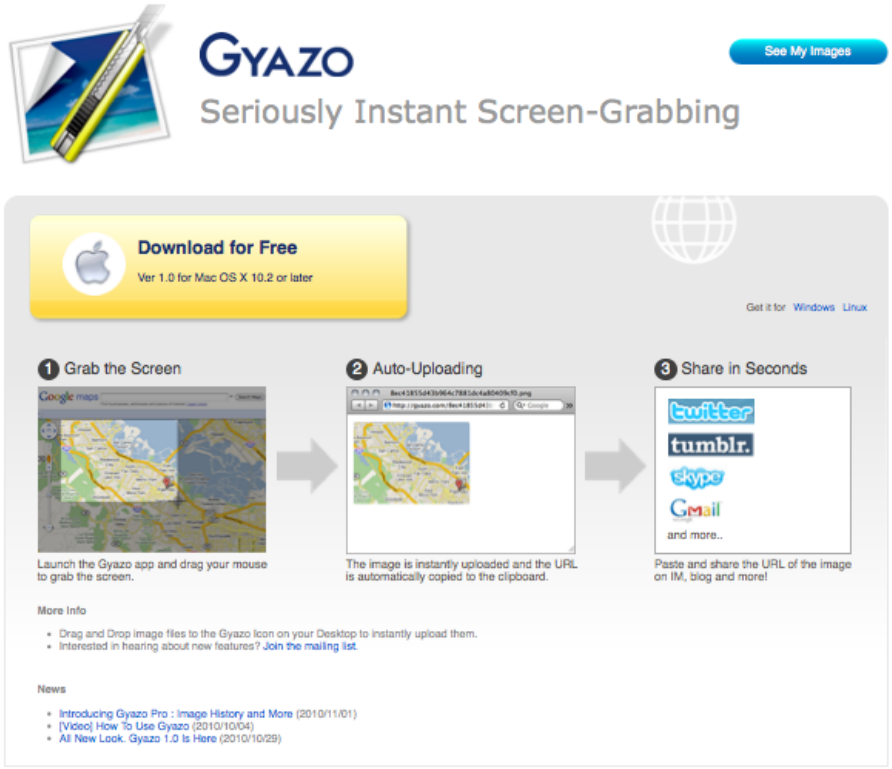
# Masui's Web services

- Practical
  - Not research-oriented
- Useful for Masui's activities
- Hopefully useful for everybody

# Masui's Web services

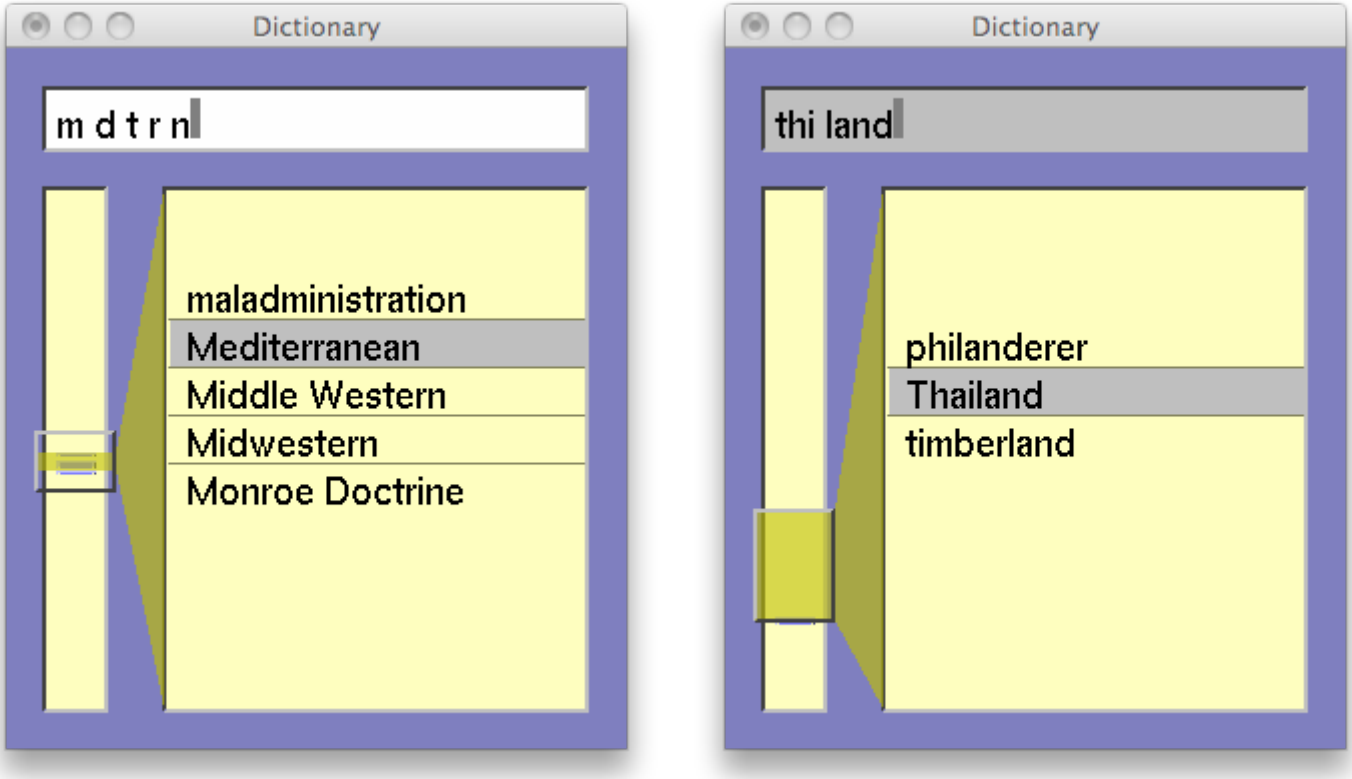
- Communication systems
  - Hondana.org, QuickML, Gyazz, Gyamm
- Authentication
  - Quiz-based authentication
- Utilities
  - Gyazo, 3Memo, Feed-TV
- Visualization / Information retrieval
  - AkimboFinder, PhotoBrowser, Lexierra, LensBar, SmoothSnap, Gyaim

# Gyazo - Instant image catpture + upload



# LensBar - Zooming information retrieval

- Easy to find a word in a dictionary



# Research trends on text input

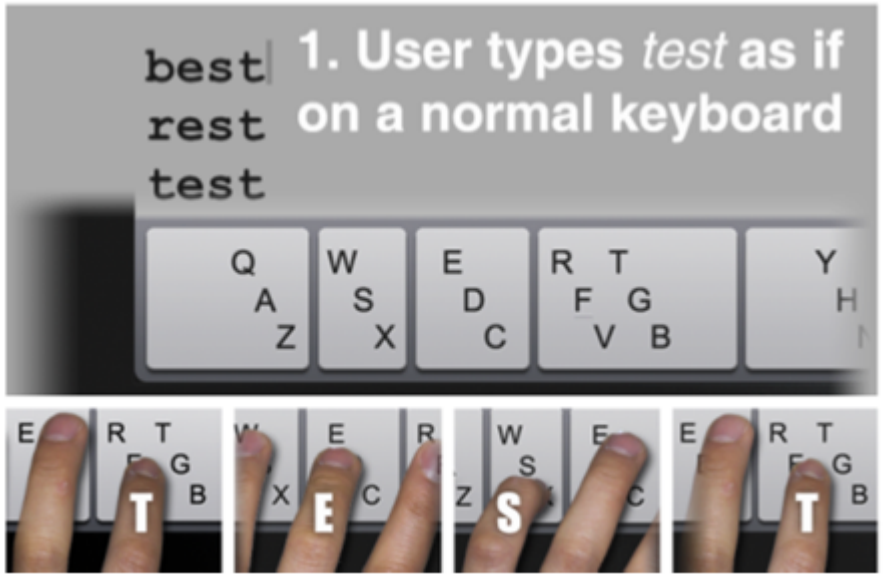
- Small devices
- Mobile environment
- Using few keys

# Papers on text input

- New papers published every year
- Same group of people writing papers
  - 2 papers at UIST
- Real innovation is rare

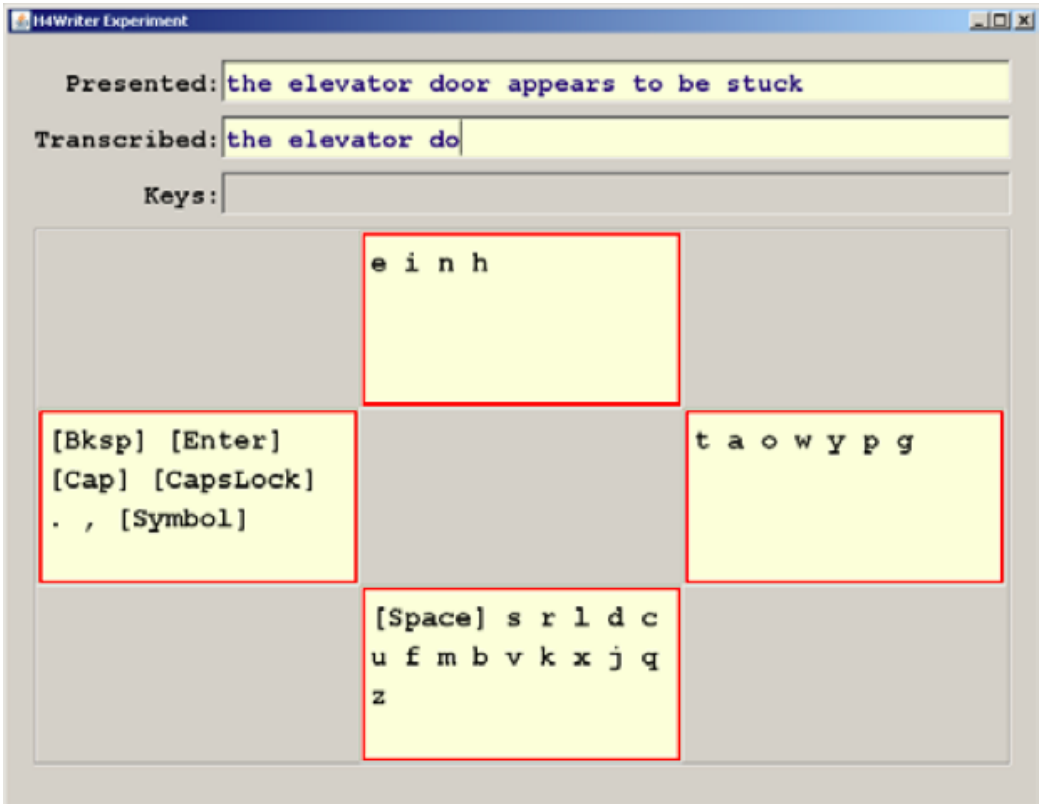
# 1Line Keyboard

- Frank Chun Yat Li, UIST2011



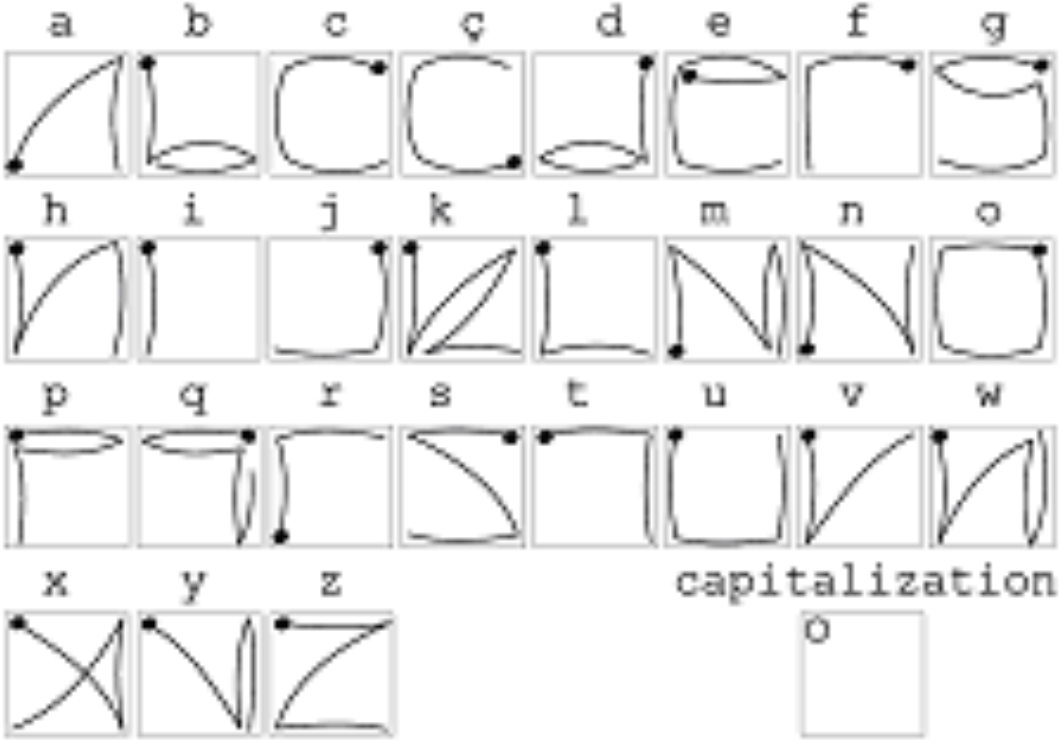

# H4-Writer

- Scott MacKenzie, UIST2011



# EdgeWrite

- J. O. Wobbrock, UIST2003 / CHI2006



# LURD-Writer

- Torsten Felzer, ASSETS 2006

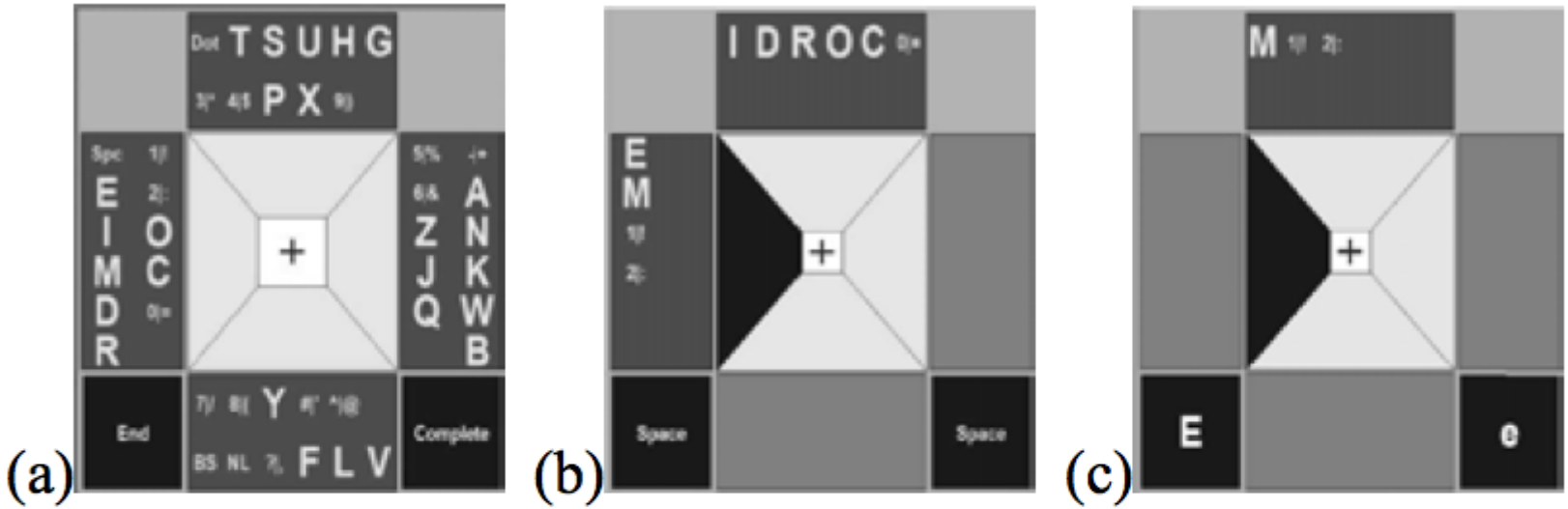



Figure 9. *LURD-Writer* input of “e” [2]. (a) Initial state. (b) After selecting left edge. (c) After selecting left edge again. Final selection of “e” occurs with a right mouse button click (left button for uppercase).

# Swype

- Video

# ShapeWriter

# Problems

- Who uses them?
- Are authors really using their systems?
- Are they really “**eating their own dog food?**”

# In my case...

- Using my own IME everywhere / every day
  - on Mac
  - on Android phone
  - on iPhone/iPad
  - on Old mobile phone
  - on Windows

# Universal Design (UD)

- Designing one system for everybody
  - People with dexterity
  - Old people
  - Small children
  - People with handicaps

# UD Princeples (Ronald Mace)

- Equitable use
- Flexibility in use
- Simple and intuitive
- Perceptible information
- Tolerance for error
- Low physical effort
- Size and space for approach and use

# Input systems should be universally designed

- Same IM for everybody, every situation
  - People with one/two hands
- Same IM for various environments
  - For PCs and phones
  - For different languages

# UD in text input

- For everybody, everywhere
- Based on a simple flexible technique
- Works for various languages
- Works for non-languages

# Restrictions in the UbiComp environment

- Slow input
- Inaccurate input
- Single-handed
- Small number of keys
- Small display

# Current Japanese text input systems

- Very complicated
  - Many buttons and function keys
  - Many conversion modes
- Different interface for different devices
  - PC vs. mobile phones
  - No shared dictionary
- Difficult to customize

# Why is it complicated?

- 30+ years of confusion
  - Integration of old interface
- Integration with application
  - Application context should be considered
  - Keystroke conflicts
- **Sentence-based conversion** (連文節変換)
  - Sentence-based text input using NL techniques
  - Very popular, but full of problems

# Sentence-based Japanese IM (SBIM)

- Convert phonems into a Japanese text
- Long years of NL research history
- Sometimes works great
  - "atsuinabe" (hot pan) => 熱い鍋
  - "atsuihi" (hot day) => 暑い日
  - "atsuihon" (thick book) => 厚い本

# Problems of SBIM (1)

- Spelling errors not allowed
- Difficult to use without a good reliable keyboard

# Problems of SBIM (2)

- Many keystrokes required
- "toukyoumade" for "東京まで(to Tokyo)"
- "tky" should be enough for "東京"

# Problems of SBIM (3)

- Error correction inevitable
- "kyouhaishaniitta"
  - "kyou haisha ni itta" (I saw a dentist today)
  - "kyouha isha ni itta" (I saw a doctor today)
- Can't tell which is correct, unless the system knows where I went
- An interface required to tell where the gap is

# Problems of SBIM (4)

- Works for a single language
- No dialect allowed
- e.g. "I couldn't eat it"
  - Tokyo: "**taberaremasendeshita**"
  - => "食べられませんでした"
  - Kyoto/Osaka: "**kuehenkatten**"
  - => "食えへんかってん"

# Problems of SBIM (5)

- Word registration not easy
- Users should know grammattical knowledge

# Problems of SBIM

- No spelling error allowed
- Many keystrokes required
- Error corrections inevitable
- Language-dependent
- Word registration difficult
- ⇒ **Far from universal**

# Advantages of SBIM

- Technically interesting
  - You can write papers!
- Intuitive to novice users
  - Just enter the pronunciation
- Impressive at first
  - Looks very intelligent
  - Annoying in the long run

# SBIM

- **Not universal**
- Was okay in the PC age
- Not ideal for UbiComp age

# Grand goal of IM

- Simple + Universal
- Use a simple and flexible technique everywhere

# Ideal text input systems

- Universal
  - All the people use the same system everywhere
  - Use the same system for various situations
    - one hand / two hands
    - standard KB / small KB
  - Use one dictionary for all devices

# Ideal text input systems (Cont'd)

- Intuitive
  - Intuition == Experience
  - Should not very different from old methods
- Simple
  - Minimum typing
  - Simple UI

# Ideal text input systems (Cont'd)

- Use resources on the Net
  - Sharing dictionaries by people
  - Using database and IR systems on the net
    - Google search
    - Wikipedia
    - Web corpus

# Ideal text input systems (Cont'd)

- Simple customization
- Same technique for various languages
  - Same method for Japanese / Chinese / Korean...
  - No special feature for each language
  - Works for dialects and even image input

# Techniques for universal text input

- IR techniques
  - Getting data from dictionaries and corpuses
- Input prediction
  - Predict next words and phrases
- Minimize user interaction
  - Use smart input devices

# IR and IM

- Result of IR can be used for IM
  - True for any input devices, any languages
- Good IR systems can be used as good IM
  - A good image retrieval system is a good image input system
- IM dictionary is good for IR

# Predictive IF

- Eliminate operations by predicting what people will do
- e.g. "tok" => "Tokyo"
  - simplest prediction
  - or called 'keyword completion'

# Example: Dynamic Macro

- If you do something twice, chances are you do it one more time

# Predictive text input


- Setup IR query with simple operations
- Dynamic query of IR results
- List the results as input candidates
- Use approximate pattern matching
  - Get "Mediterranean" from "m d t r n"

# Source of the prediction

- User's context
- Usage history
- Corpus
- Search history

# Masui's input systems

- POBox
  - Palm, mobile phones, Mac, Windows, Emacs
- Soft KB + handwriting recognition
- Few-key text input
- Image input
- IM on browsers
- IM testbed on Mac

# POBox: Predictive Operation Based On eXample

- Approximate word search
- Simple prediction
  - "tok" => "Tokyo"
  - "Hello" => "World"

# Simple word prediction

- Corpus: "Hello World"
- Input: "Hello"
- Prediction: "World"

# POBox on Palm

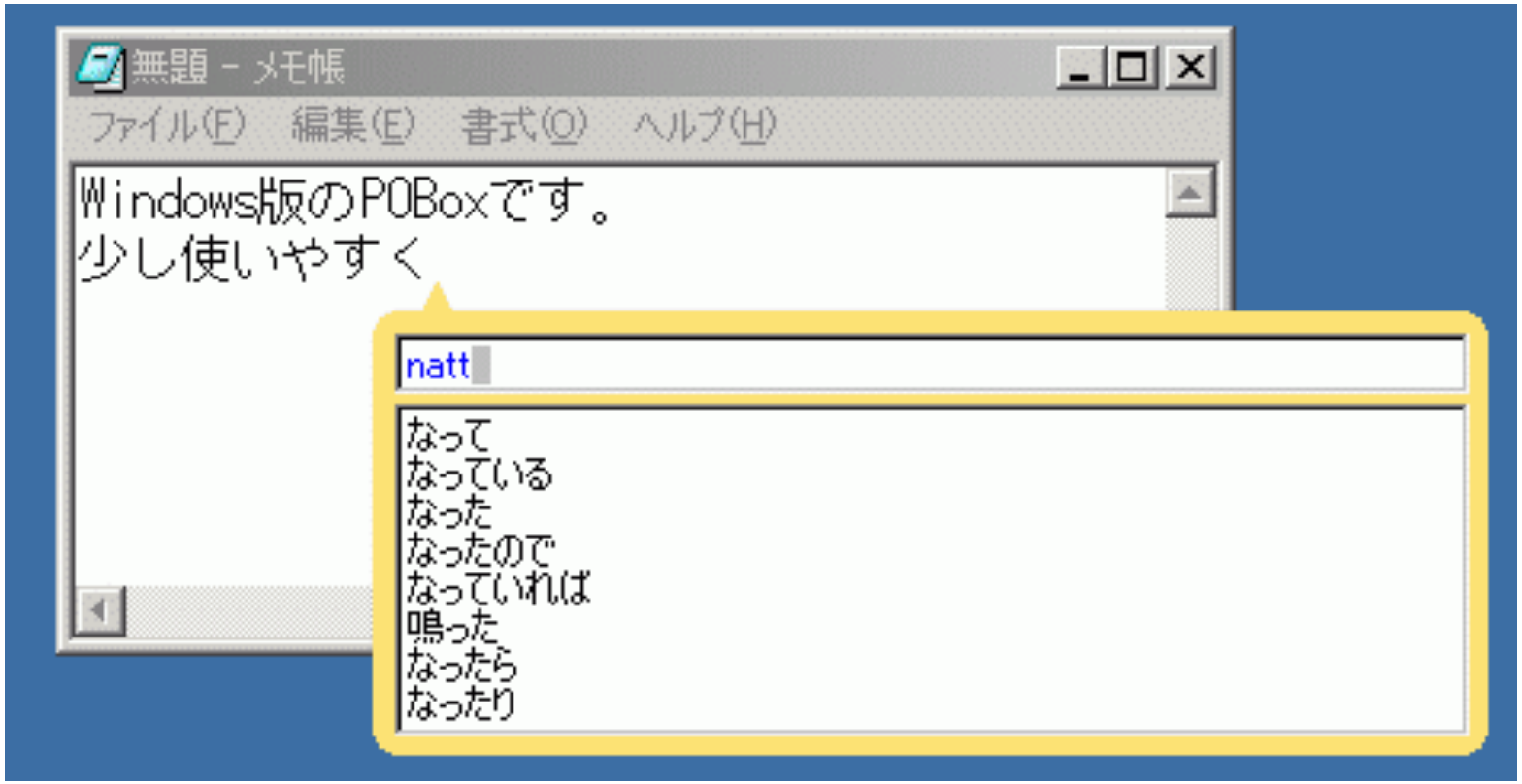
# POBox on Japanese mobile phone (2001)

# POBox on Japanese Android phone (2010)

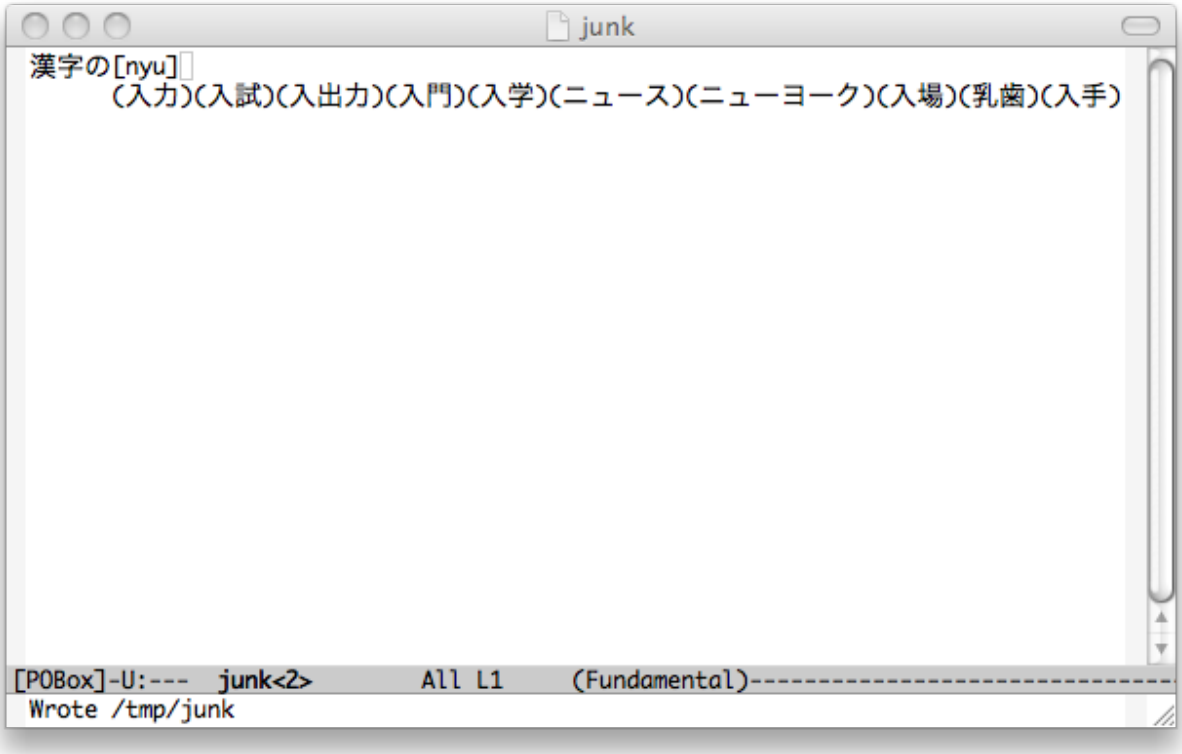
# "Flick input" on iPhone

- An implementation of the "T-Cube" system

# POBox on Windows

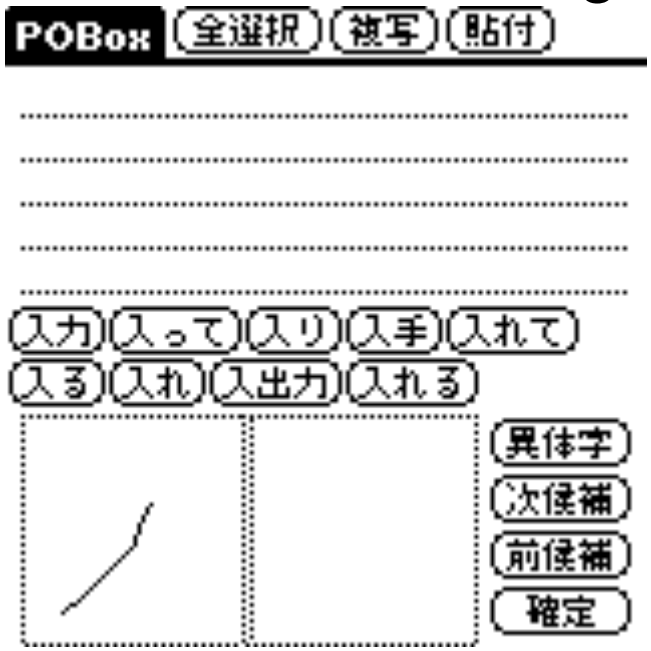


# POBox on Emacs



# POBox with handwriting recognition

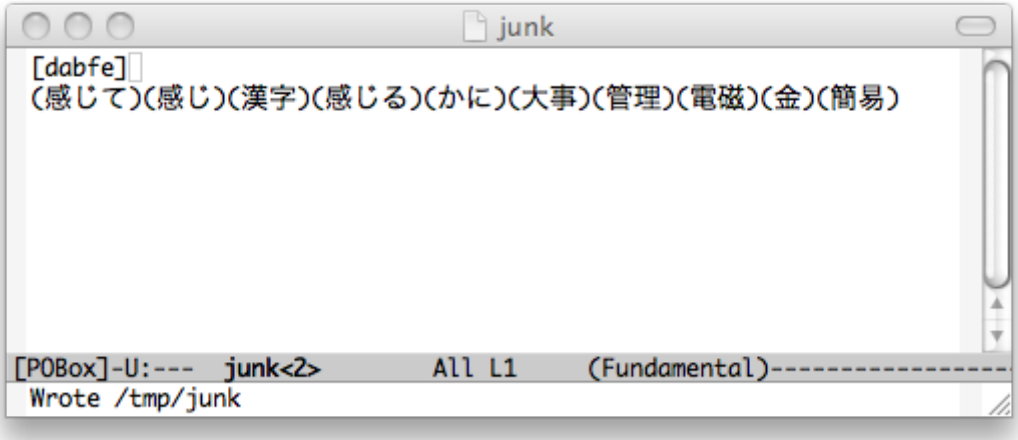
- Integration of software KB + handwriting recognition

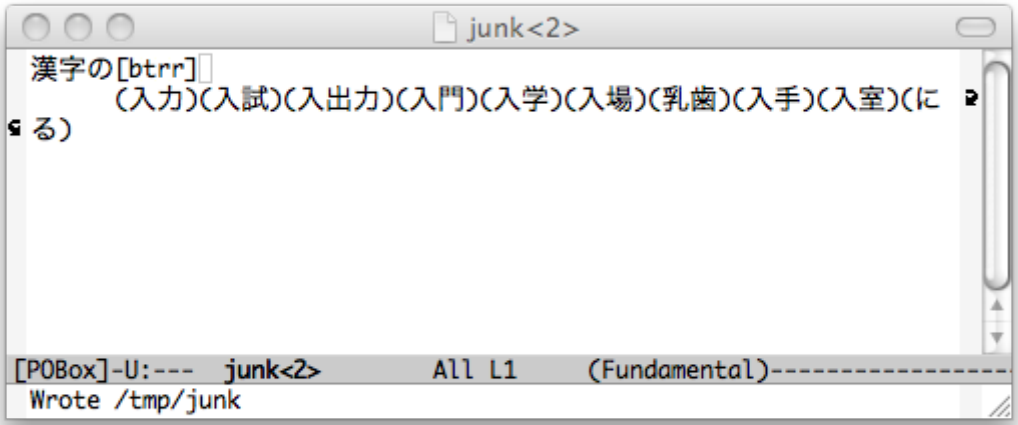


# Half-Qwerty

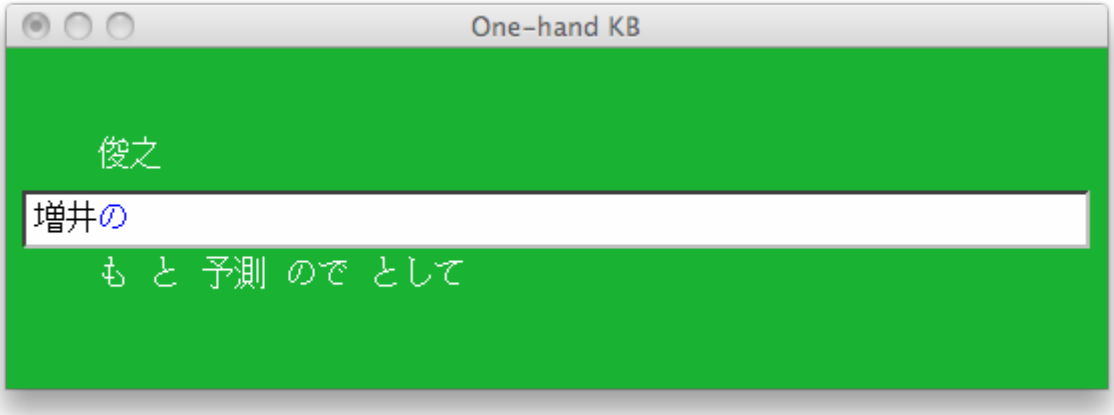

Tab
Backspace
Esc
Delete
Del
PgUp
Home
PrtSc
A-Z
0-9
Return
enter
Ctrl
ScrLK
Shift
Shift Shift
help
Ins
PgDn
clear
NumLk
End
Pause
Alt
option
Matias
halfkeyboard.com

# One-hand Emacs



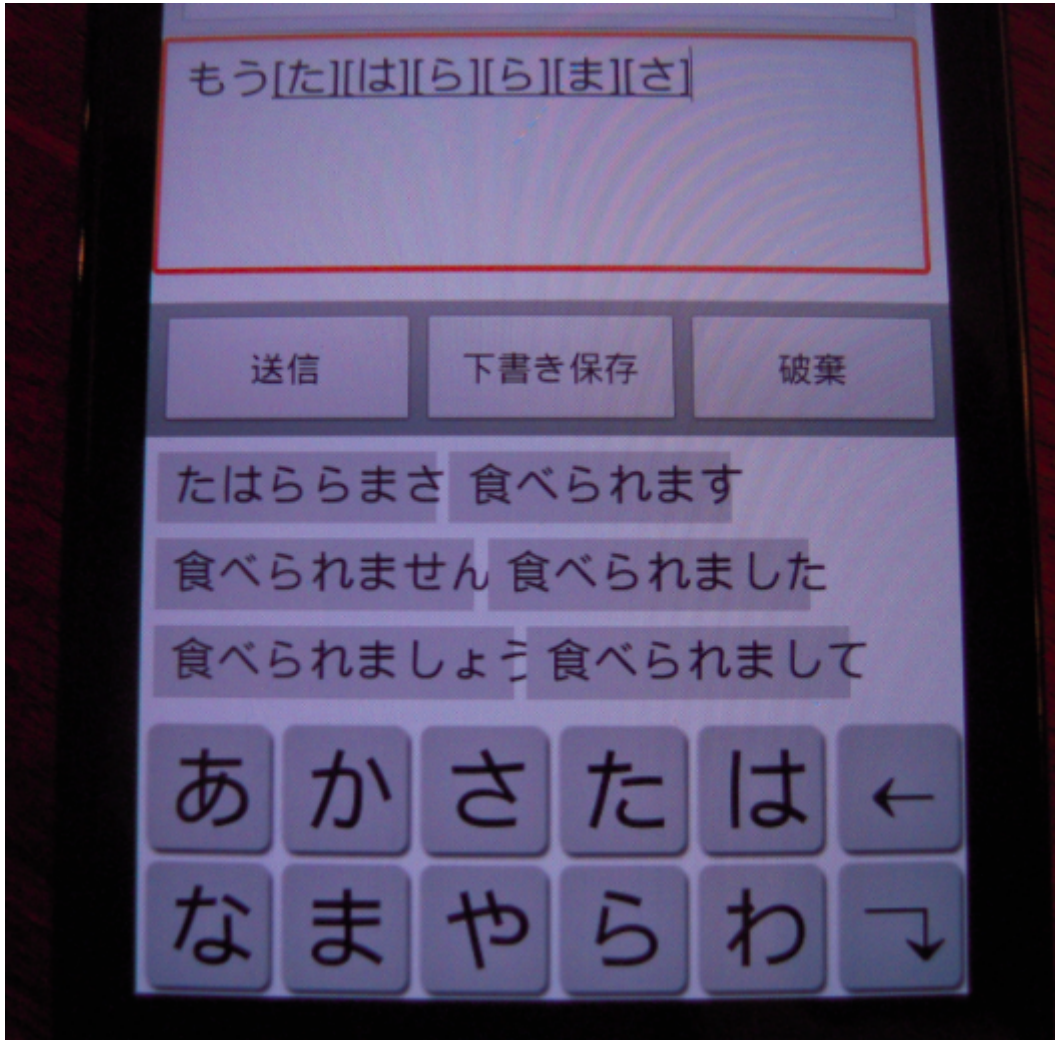



# 4-key POBox



# T9+menu

- Integration of T9-type input and menu


もう[た][は][ら][ら][ま][さ]
送信
下書き保存
破棄
たはららまさ
食べられます
食べられません
食べられました
食べられましょう
食べられまして
あ
か
さ
た
は
←
な
ま
や
ら
わ
↵

# POBox on a browser

## Gyazoの画像を使って絵文字入力

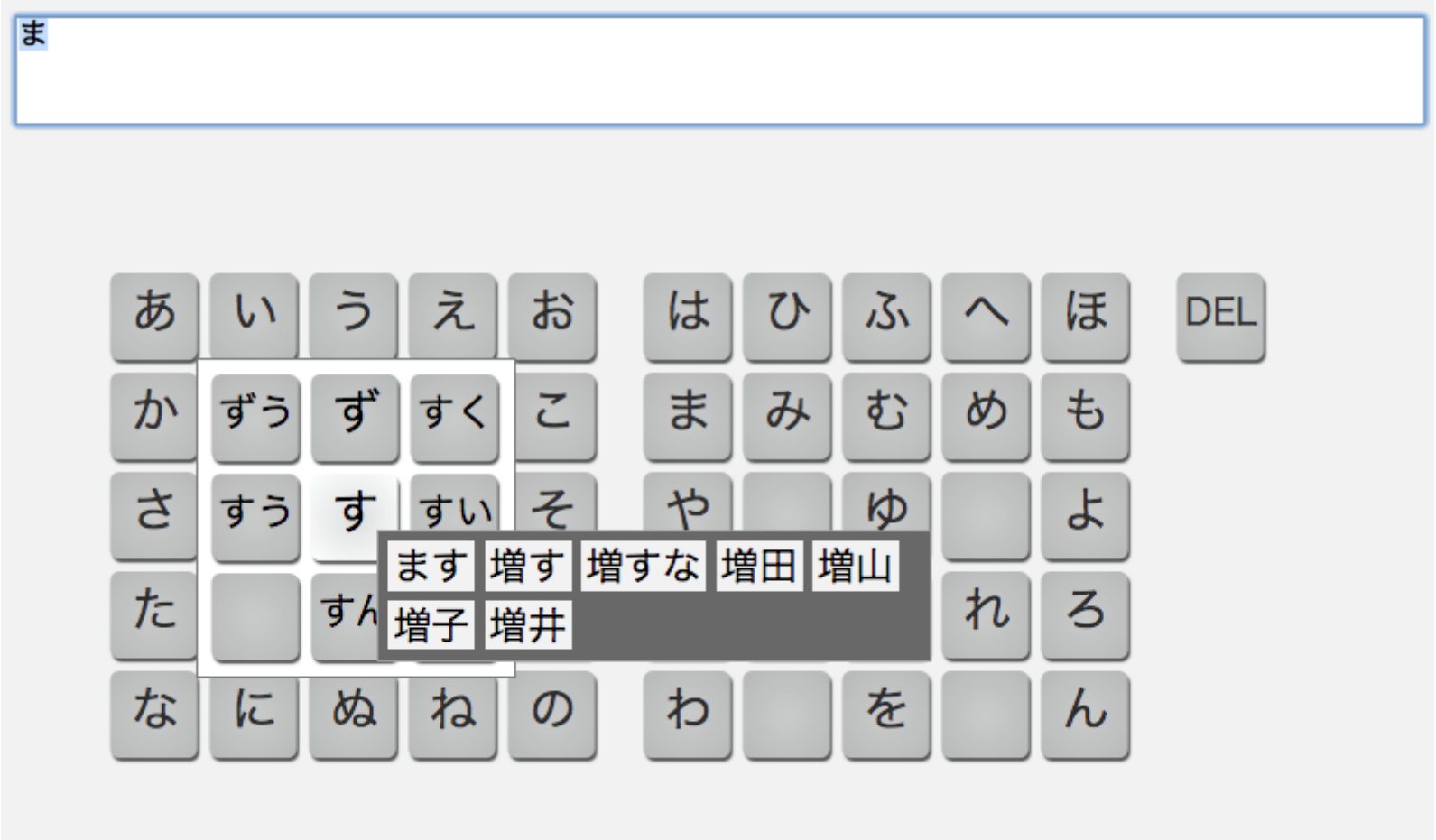
- Gyazoに登録した画像を文字入力に使います。
- 読みはGyazoで登録できるようにする予定。
- ESCで変換/非変換を切り換えます。
- 辞書(2M)のロードに時間がかかるかもしれません。ゆっくりお試し下さい。

 と  と  で宴会しましょう

# Text input system + browsers

- IM can be used as a menu
- Special IM for special service
- No need to prepare IM for each machine

# Multi-level flick



# Gyaim

- A simple IME on Mac
- **600** lines of code in MacRuby + IMKit (Cocoa framework)
- "IME development made easy"

# Findings

- Simple IME is usually enough
  - Using only simple input methods for years
  - Same dictionary / algorithm
- NL processing is sometimes very useful
  - Integration with more universal methods is important

# Unsolved problems

- Shared dictionaries
- Word registraton methods
- More ntegration of IR and IM

# Dictionary sharing in the network age

- Share high-quality dictionary entries
- When do people post a dictionary?
  - Hint: Wikipedia?

# Word registration

- Usually not easy
  - Nobody edits his dictionary
- Special registration UI required
- Integrating search and registration

# Integration of IR and IM

- Nobody writes texts from scratch
- For doctors, lawers, bloggers, ...

# Conclusion

**Let's solve all the problems for the grand goal!**