CS 338: Graphical User Interfaces

Lecture 7-1: Advanced Input

Interface Input Modalities

- How can a user give input to an interface?
  - keyboard
  - mouse, touch pad
  - joystick
  - touch screen
  - pen / stylus
  - speech
  - other…

more error!
harder! (?)

Keyboard input

- Short history of the keyboard
  - typewriter patented by Christopher Latham Sholes in 1868
  - Sholes patent sold to Remington & Sons in 1873
    - Remington adopted the QWERTY layout
  - companies held contests to see whose typewriters & typists could enter the fastest
  - why did the QWERTY keyboard win out?
    - people disagree... seemingly part marketing, part luck

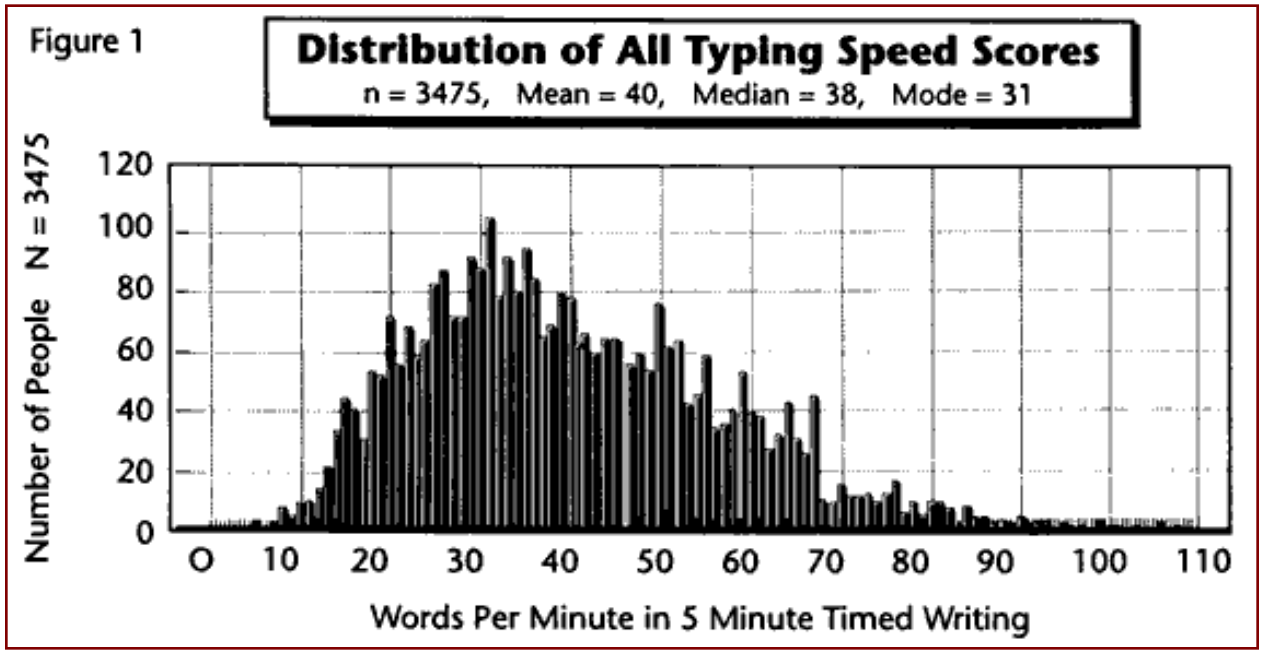
"Typing errors," by Stan Liebowitz and Stephen E. Margolis.
Available at http://reason.com/9606/Fe.QWERTY.shtml

Keyboard input

- Short history of the keyboard (cont.)
  - Dvorak Simplified Keyboard patented by August Dvorak in 1936
    - "key" feature: common keys on home row

Keyboard input

- So how fast can people type anyway?



- Of course, this is touch-typing normal text...

"Typing Speed: How Fast is Average," by Teresia R. Ostrach.
Available at http://www.fivestarstaff.com/publication_typing.htm

Mouse input

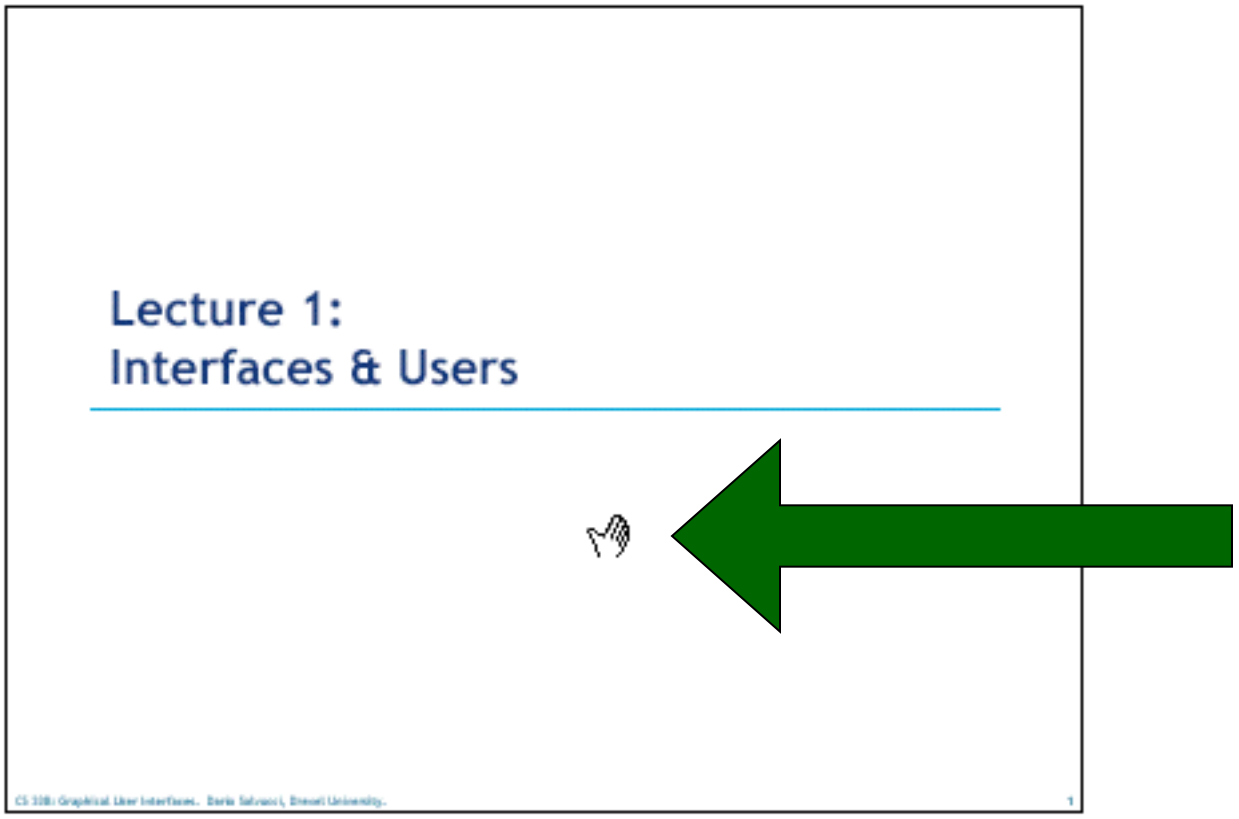
- Short history of the mouse
  - Douglas Engelbart invented the mouse in 1963 in a larger project to "augment human intellect"
    - and thereby invented "point and click"

Direct manipulation

- Why is the mouse so effective?
  - at least part of the answer lies in the facilitation of "direct manipulation"
- Direct manipulation involves...
  - visual representation of the manipulated objects
  - physical actions instead of text entry
  - immediately visible impact of the operation

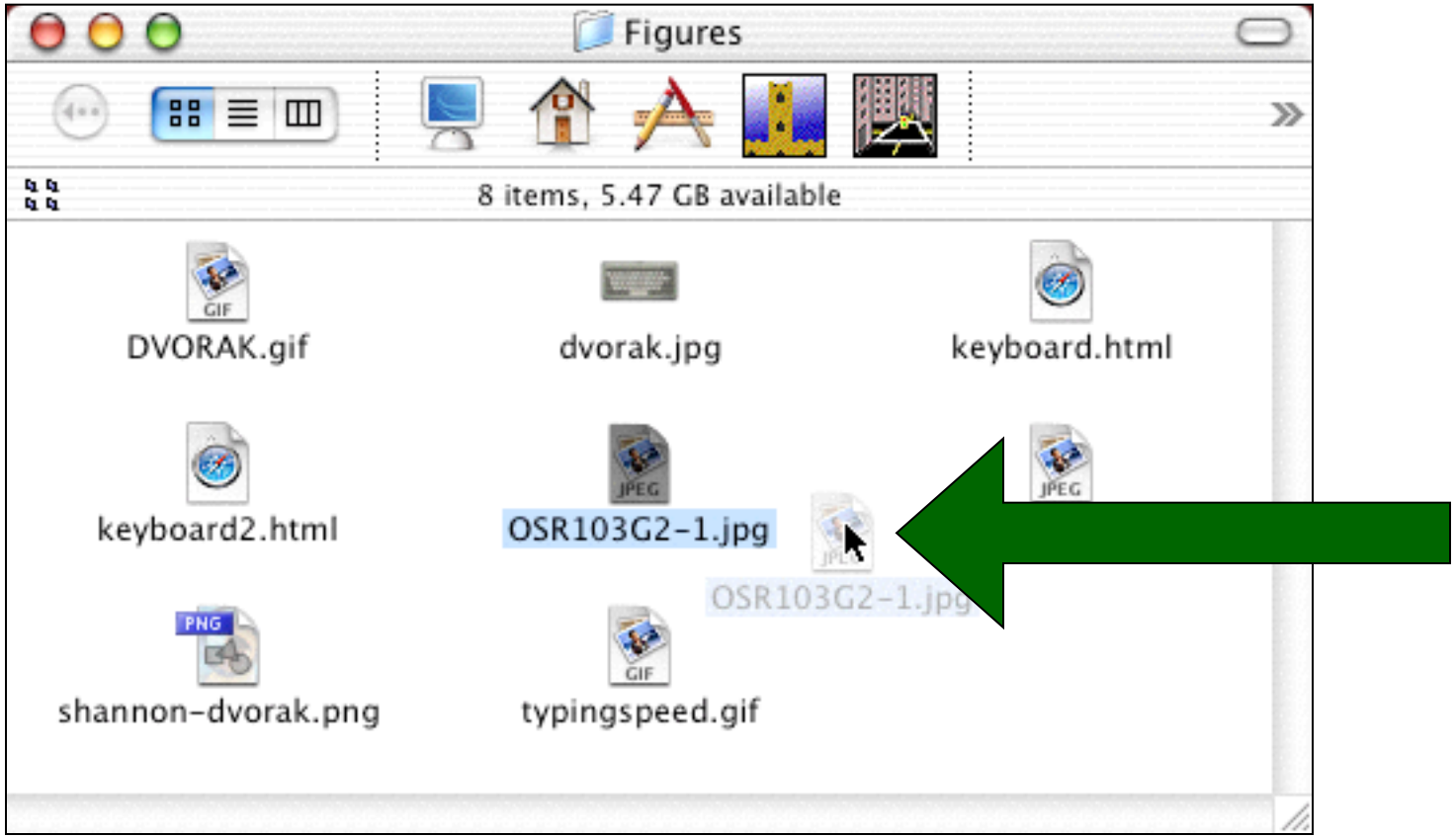
Direct manipulation

- Example: "Hand" dragging



Direct manipulation

- Example: Desktop folder & file manipulation



Related input

- Trackball
  - nice for games, small footprint
  - why aren't they mainstream? metaphor?
- Trackpad
  - small & thin — perfect for laptops
  - integrates notions of mouse & pen
  - now, multitouch trackpads common
- Trackpoint (IBM)
  - pencil-eraser nub
  - (more or less out of use now, but still exist)

Pen input

- Pen-based interfaces in computing

Back in the day...

Today...

Pen input

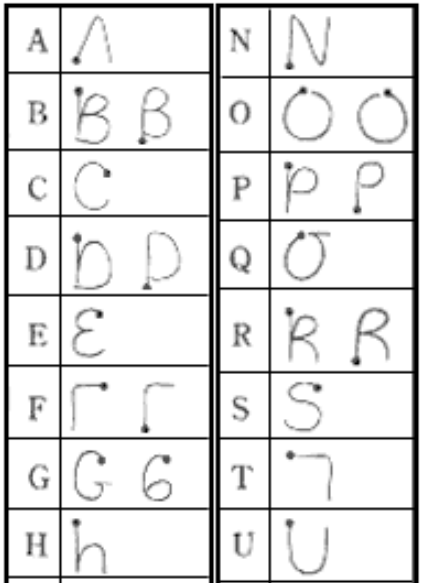
- Pen input has lots of advantages...
  - users can hold a familiar input device
  - users can write in a familiar input language
  - users can carry with them
- ... and some major disadvantages:

Kearney

Newton

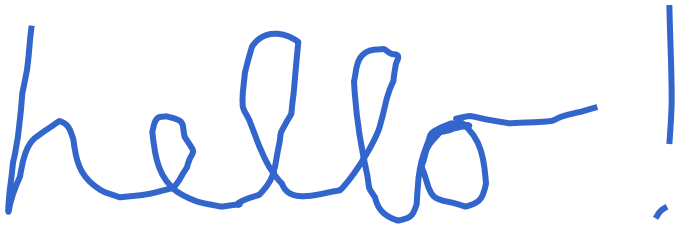
Pen input

- Handwriting
  - very general, well-developed human skill
    - thus, make use of what users can already do!
  - but hard to recognize (for people & machines)
- Gestures
  - gesture alphabets
    - Palm Pilot graffiti
  - editing gestures
  - easier to recognize



Off-line vs. on-line recognition

- Off-line recognition
  - examine static output of handwriting, i.e., the end result of the writing

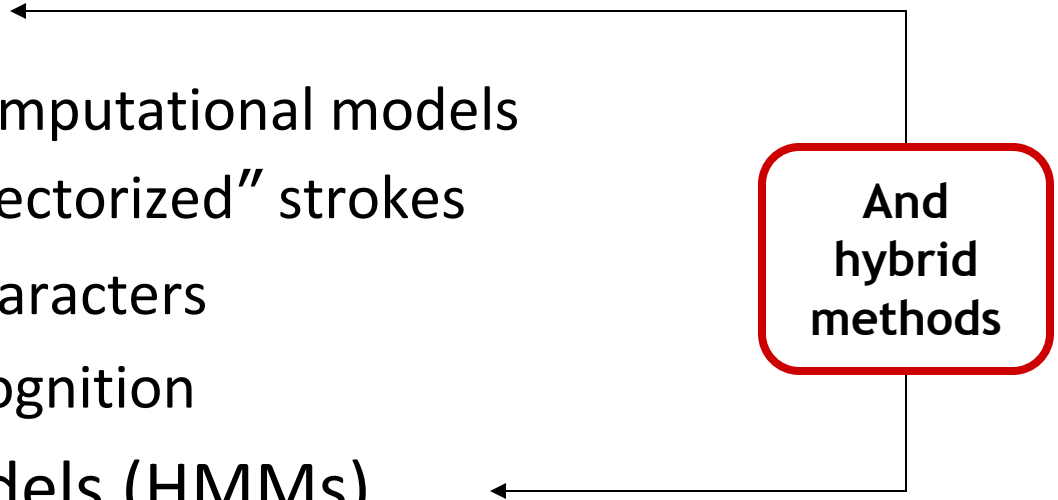


Like OCR, but much harder — Why?

- On-line recognition
  - examine dynamic movement of handwriting, i.e., the strokes, pen up/downs involved
- Which is more "informed"? more useful?

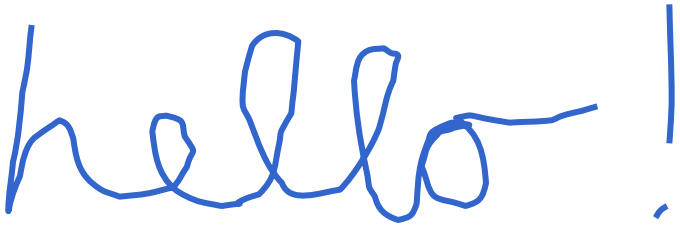
Recognition techniques

- Neural networks
  - neurally-inspired computational models
  - input: bitmap, or "vectorized" strokes
  - output: probably characters
  - best for off-line recognition
- Hidden Markov models (HMMs)
  - powerful probabilistic models
  - input: vectorized strokes
  - output: full recognition of chars, words, etc.
  - best for on-line recognition
- Consider HMM-based on-line recognition…



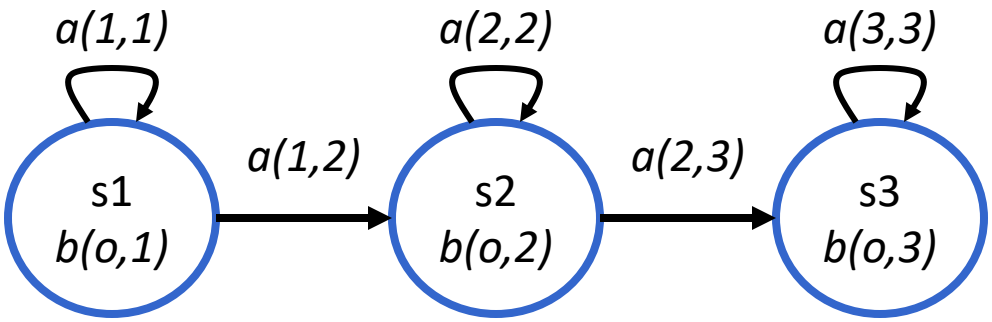
On-line feature extraction

- On-line strokes -> feature vectors
  - basic features: pen up/down, direction, velocity
  - useful features: curvature, reversal, …
  - other features: ascender/descender? …



On-line recognition

- Hidden Markov models (HMMs)
  - probabilistic models for dynamic behavior
- Set of N states with
  - a(i,j) = probability of state transition i→j
  - b(o,i) = probability of seeing o in state I
    - can be discrete or continuous prob. distributions

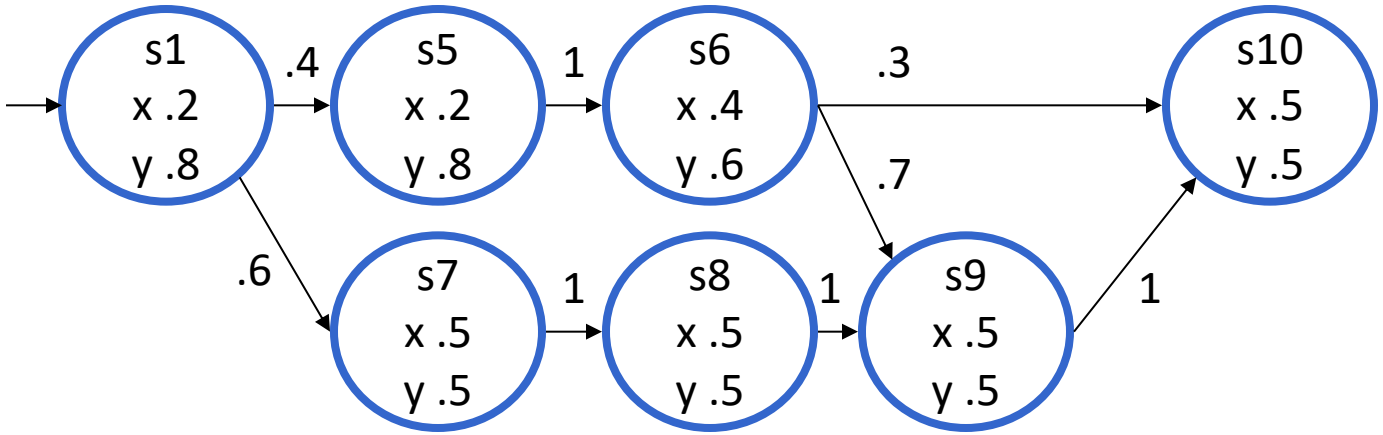


Hidden Markov models

- Let's say we have
  - M = HMM representing predicted behavior
  - O = observation vector sequence O
- Three problems
  - evaluation: find Pr(O|M)
  - decoding: find the state sequence Q that maximizes Pr(O|M,Q)
  - training: adjust parameters of M to increase Pr(O|M)

Hidden Markov models

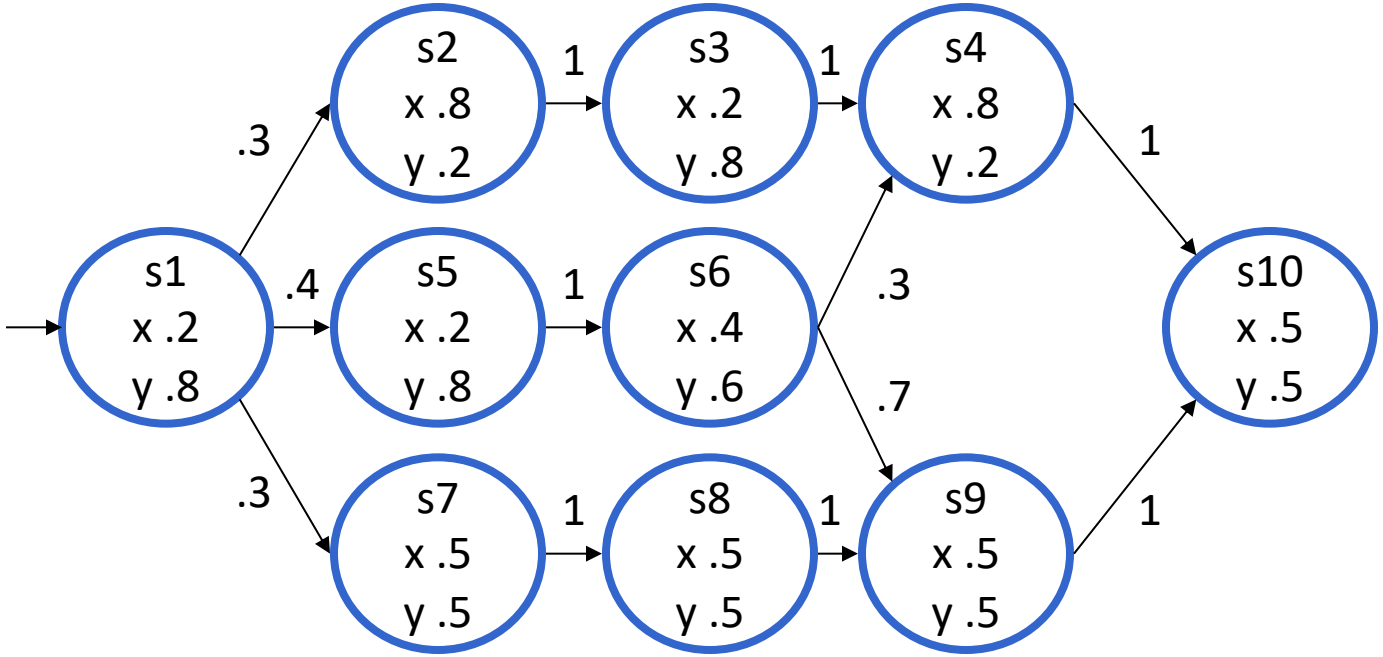
- HMM evaluation
  - find Pr(O|M)
  - evaluate O = < x x y x > …



  - can we do this efficiently?

Hidden Markov models

- HMM decoding (Viterbi algorithm)
  - find best state sequence through HMM, maximizing the probability of the sequence
  - assuming O = < x x y x y > , try to decode…



Hidden Markov models

- HMM training (Baum-Welch / EM algorithm)
  - re-adjust a(i,j), b(o,i) to increase Pr(O|M)
  - iterative procedure
  - allows for fine-tuning of HMM parameters for particular observation sets
    - what's in the set? everything? a single person? a specific group of people? (e.g., R/L-handed for writing)
  - susceptible to local minima!
    - and this is often a problem

Hidden Markov models

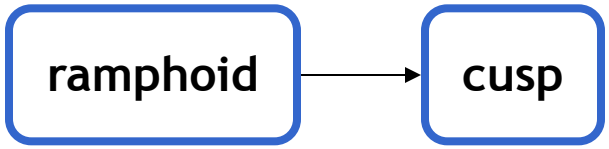
- Composing HMMs
  - we can add "sub-HMMs" into larger HMMs, creating a model hierarchy at different levels
- For instance, we can create three levels
  - strokes
  - letters
  - words

On-line recognition

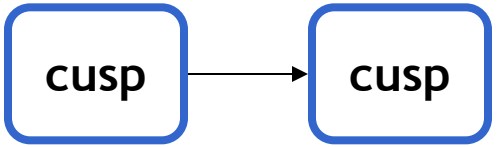
- Stroke HMMs with states
  - up-down loop
    - s1: up, + curvature, hi velocity
    - s2: down, + curvature, hi velocity
  - up-down cusp
    - s1: up, + curvature, , hi velocity
    - s2: 0 velocity
    - s3: down, + curvature, hi velocity
  - up-down ramphoid
    - s1: up, – curvature, hi velocity
    - s2: 0 velocity
    - s3: down, + curvature, hi velocity

On-line recognition

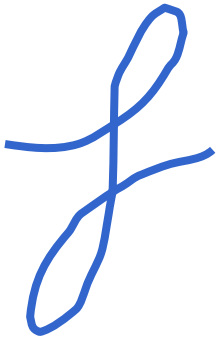
- Letter HMMs based on stroke HMMs



('o' ?)



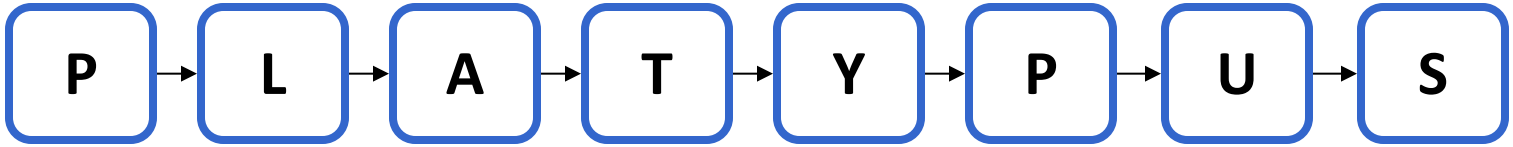
('w' ?)



('g' ?)

On-line recognition

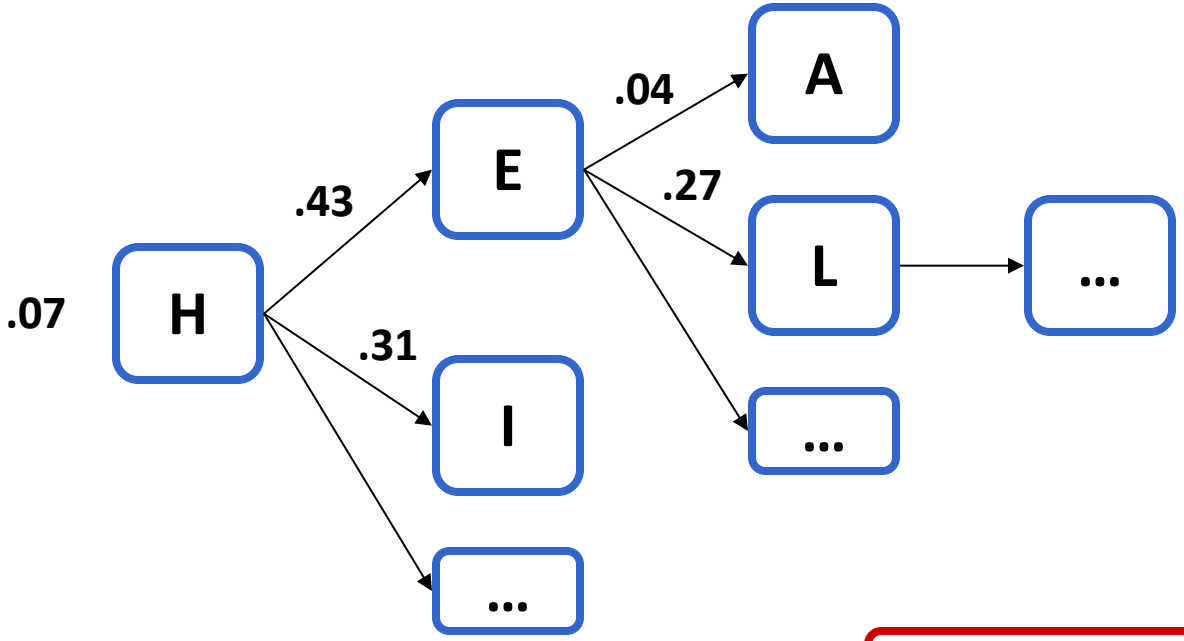
- Word HMMs based on letter HMMs



  - basic idea is straightforward
  - but it's deceptively tricky — why?

On-line recognition

- Putting it all together
  - compacting states
  - taking word frequencies into account
  - where do frequencies come from?



Note: Probabilities aren't real…

Handwriting issues

- Vocabulary size
- Individual variability
  - writer dependent, adaptive, independent
- Signal segmentation
  - isolated words, continuous
- Printed vs. handwritten?
  - often some combination of the two!
- Speed-accuracy tradeoff
- Dotting i's, crossing t's, … for on-line recognition
- Mixing language with graphics & gestures

Speech input

- Another useful input modality, at times
- A few benefits of speech...
  - like handwriting, very natural
  - hands-free
    - you can alleviate physical issues (like carpal tunnel)
- ... and a few drawbacks:
  - some people hate talking
  - some people hate other people talking
  - and again, the recognition problem

Speech recognition

- Limited speech recognition
  - only allow small sets of words/phrases
    - e.g., "one" – "nine"
- Full speech recognition
  - again, general, well-developed skill
  - full standard vocabulary (1000-10000+ words)
  - specialized vocabularies (research, medical, …)
  - "editing" vocabularies (back, delete, …)
- Same basic ideas for recognition
  - convert to recognizable signal (transforms)
  - recognize using hybrid methods along with hierarchy of phonemes, words, etc.

Speech issues

- Speech has many of the same issues as pen and handwriting input
  - vocabulary size
  - individual variability
    - speaker dependent, adaptive, independent
  - combining phonemes (co-articulation)
  - signal segmentation
    - isolated words, continuous

Multimodality

- Today's desktops mostly use mouse & keys
- We've already started expanding the horizon
  - touch input is very common
  - speech input is common
  - pen tablets around, though not hugely popular
- Off the desktop, it's even more important
  - virtual worlds
    - allow for pointing, grabbing, etc.
    - "direct manipulation" becomes even more direct
  - in-vehicle devices
    - visual & manual channels are busy
    - can interface exploit other channels?
    - what are the cognitive implications?

Other input modalities

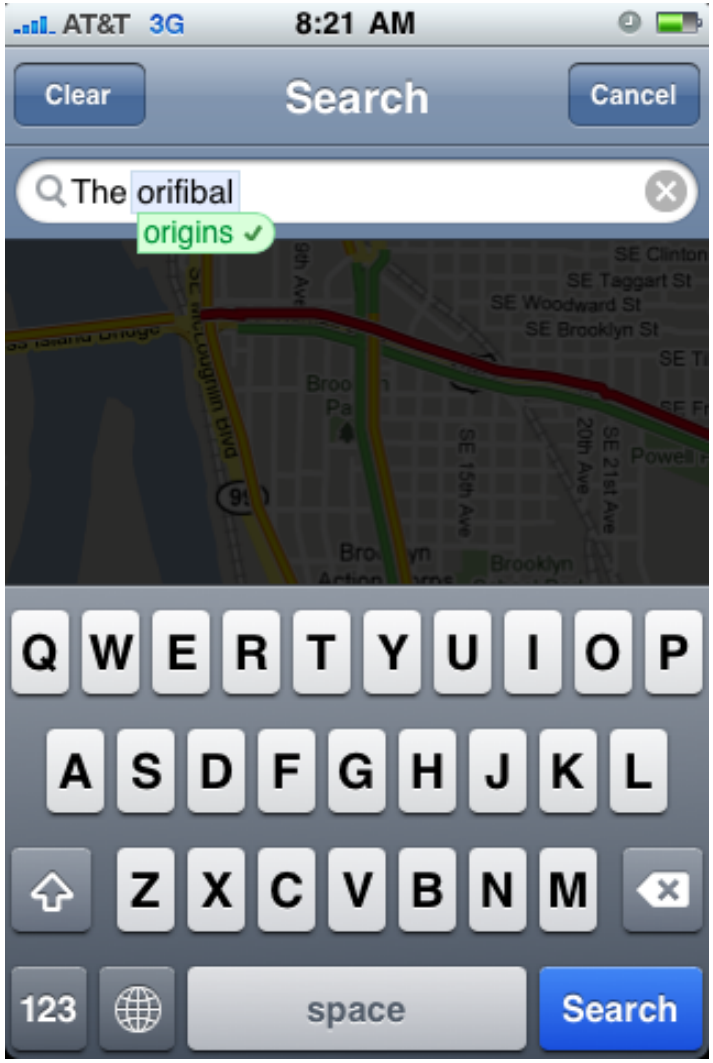
- Touch input

Other input modalities

- Touch input
  - Auto-correct
    far from perfect
  - (byssinosis =
    "brown lung
    disease")

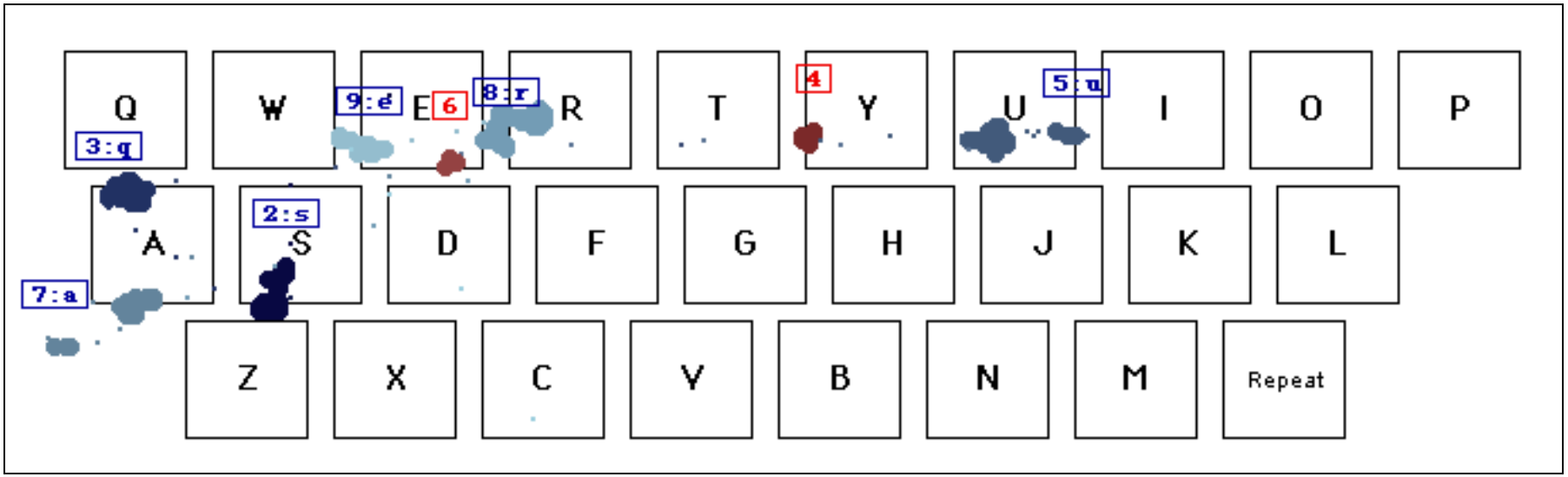
Other input modalities

- Touch input
  - Suggest in terms of map info?
  - (This image is faked, I think)



Other input modalities

- Eye-movement input



Redundancy

- Redundancy is a great thing
- Use many input devices together
- Redundancy is a great thing
- Allow the user to choose between them
- Redundancy is a great thing
- And not choose once, but continuously
- Redundancy is a great thing
- With freedom comes speed & usability
- Redundancy is a great thing

Redundancy

- Command examples of redundancy
  - mouse & keyboard
    - e.g., menu commands & keyboard shortcuts
  - commands in different places
    - e.g., menu commands & toolbar items
  - mouse & eye?