Handwriting and Gesture Recognition

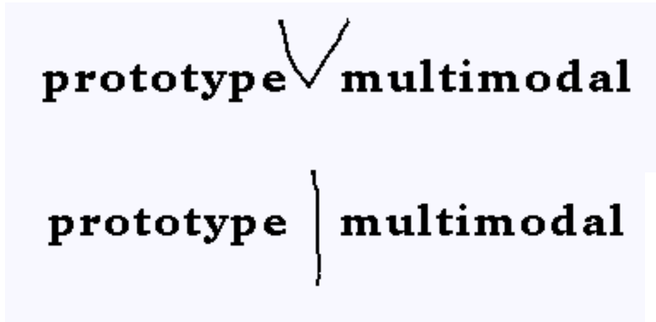
Interactive Systems Laboratories

Speech, Handwriting, Text

Gestures

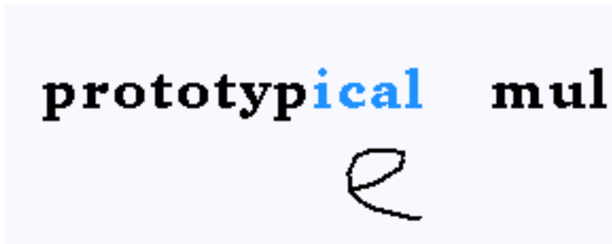
Delete Words and Characters:

Indicate Cursor Position:



Select Characters:

Partial Word Correction:



Writing and Language

- About 6000 living languages exist
  - India: > 1600
  - South America: > 1000
  - Africa: > 1000
  - Europe: < 70

- 90% of world population speak one of the 100 widely used languages

Writing and Language

- Only for 13% of the living languages, a written language exists

- Chinese, English, Spanish, Russian, Hindi, German cover ~50% of the world population

- Number of written languages ever used : ~660

Writing and Language

- First cave drawings:
  more than 30.000 years ago

- First writing systems:
  ~ 5000 b.C.   Sumerer in Mesopotamien
  ~ 7000 b.C.   South Eastern Europe
- Latin Alphabet:
  600 b.C.

Writing and Language

- Hundreds of written languages use the latin alphabet
- The latin alphabet is based on the sound of words (a significant departure)

Writing and Language

Image
Symbol
Word

Characters linked to Meaning, Pictographs (China)

Syllable
Characters
Phonetic Smbols

Phonographs

Reduction of the Number of Symbols by way Phonetic Sound-based Systems (since ~ 1500 BC)

Language Input by Speech

- Fast Input of Long Texts (dictation, essays,…)
- Data Input under Devided Attention (Driving,
  Operating Machinery, ..)
- Hands, Eye Busy Situations
  (Surgery, Construction, Human Postal Sorting...)

Why Handwriting ??

Problems:

- It is Slower than Speech and Typing
- Recognition comes with Errors

Handwriting

Data Input without Keyboard and Mouse

- Cell Phones
- Personal Digital Assistants
- Palmtops
- WebPads
- Outdoor-Activities
- ...

Handwriting

Input of brief messages:

- Keyboard substitute (?!)
- Mouse substitute

Handwriting

Communication in noisy environments:

- Factories
- Conventions
- Discos
- ....

Handwriting

Silent Communication:

- Meetings
- Presentations
- Military operations
- ...

Handwriting

Communication of Confidential Data:

- Personal Data
- Credit Card Numbers
- Codewords
- ...

Handwriting

Under Conditions, were verbal communication is not possible:

- Under water
- Handicapped people
- ...

Handwriting

Input of spatial data:

- Forms
- Mathematic Formulas
- Crossword puzzles
- ...

Handwriting

Symbolic Data

- Graphs, Tables
- Symbolic gestures
- ...

Handwriting

Input of Biometric Data
Person Verification and Identification

- Signatures
- Writing Style

Handwriting

Error Correction and Comments:

- Annotation and Modification of Documents
- Correcting Voice Recognition Errors
- ...

Handwriting

A Parallel, Alternate Input Modality !!

Redundance
Naturalness
Robustness
Flexibility

Handwriting Recognition

A Handwriting Recognizer transforms handwritten input in a computer readable format (e.g. ASCII)

Handwriting Recognition

Applications

Postal Sorting

Mail:

World-wide: ~ 1 billion mails a day

US: ~ 43 Million mails a day (Germany ?)

of which 30 million can be machine-processed (3 x Mount Everest)

Most of it „Junk Mail“ which (luckily) has textual labels

Postal Sorting

Throughput:

Human: 3800 - 5000 Mails per hour

Machine: 30 000 Mails per hour

Handwriting Recognition

- Other Applications:
  - Processing Forms (UPS, FedEx, …)
  - PDA’s, Palmtops
  - Graphic Tablets

On-Line vs. Off-line Recognition

Two types of Applications and Systems:

- Off-Line:
  - Computer input by scanning
  - Handwriting is stored as binary greyscale image
  - Writer doesn‘t need special hardware (paper and pencil are enough)
  - Data capturing can be done any time
- On-Line:
  - Need Tablet and Pen
  - Collect x,y Coordinates as a function of time
  - Use Temporal Information
  - Better Performance

Off-line Handwriting Recognition

- Possible applications include
  - check reading
  - postal address reading
  - document analysis, ...
- Input consists of scanned bitmaps without any temporal information
- Eventually location of handwriting needs to be found (document analysis)
- Stroke order doesn´t influence recognition
- But: problems through overlapping or touching characters and noisy input

Document Analysis

Figure 9. Original.

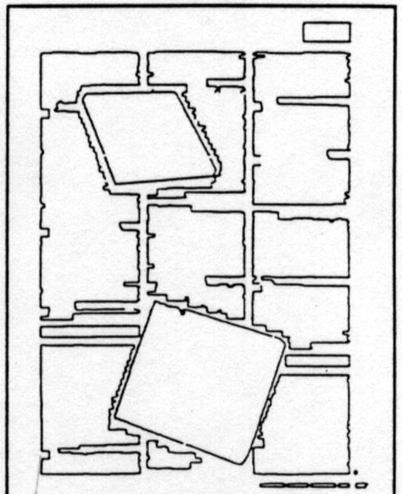


Figure 10. Result.

Figure 11. Original image with 15° skew.

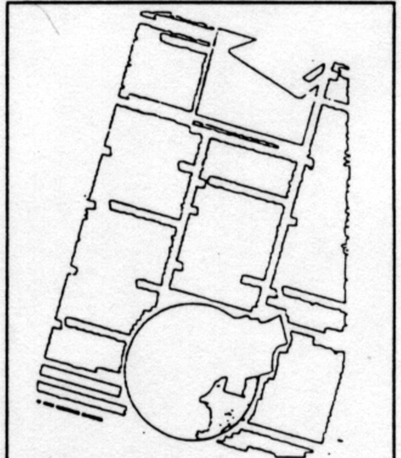
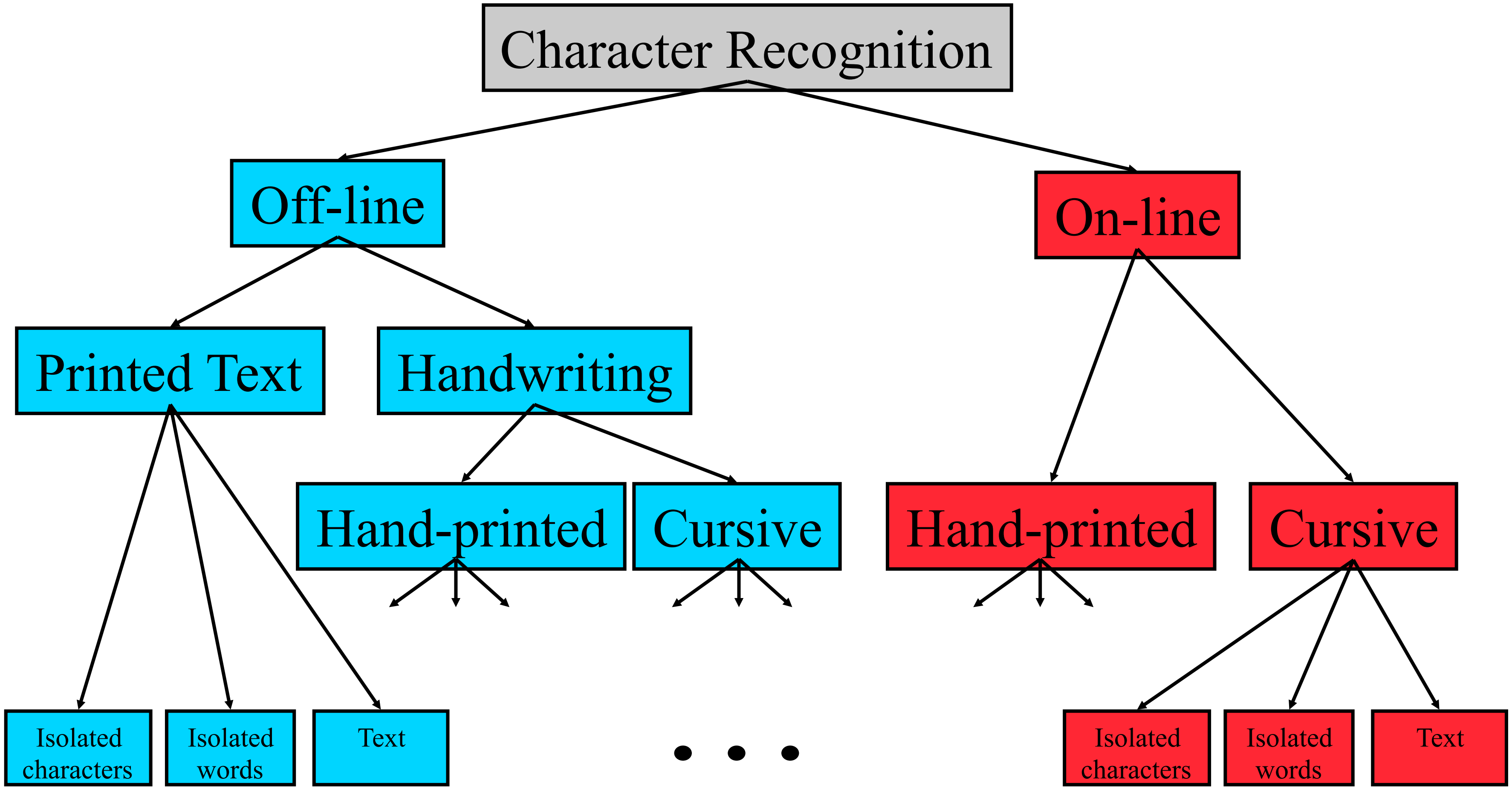


Figure 12. Segmentation result.

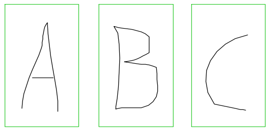
Character Recognition



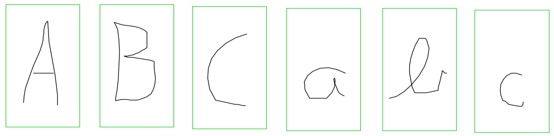
Evaluating Handwriting Recognition Systems

- Comparing different handwriting recognizers is difficult
- Performance depends on
  - recognition task (e.g. isolated characters, words, unconstrained text)
  - writing style (e.g. printed, mixed, cursive)
  - size of dictionary
  - intended end user(s)
    - single writer (allows writer dependent system)
    - multi-writer
    - Writer-independent (requires writer independent system)

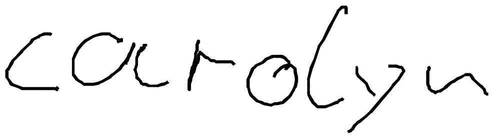
Handwriting Recognition Tasks



boxed upper case letters



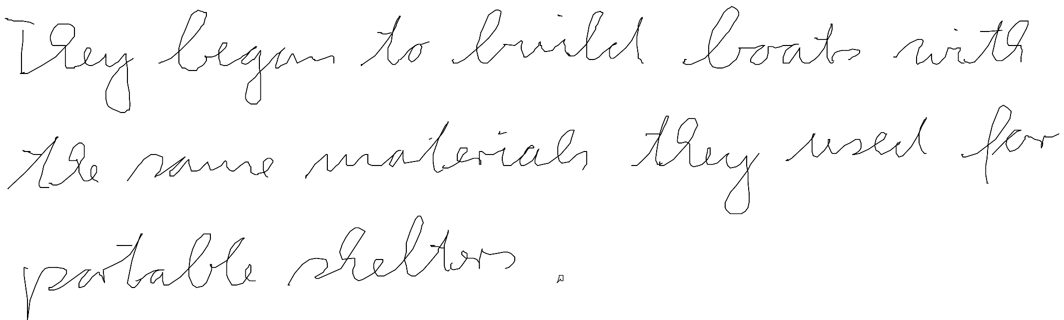
boxed letters



printed words

mixed words

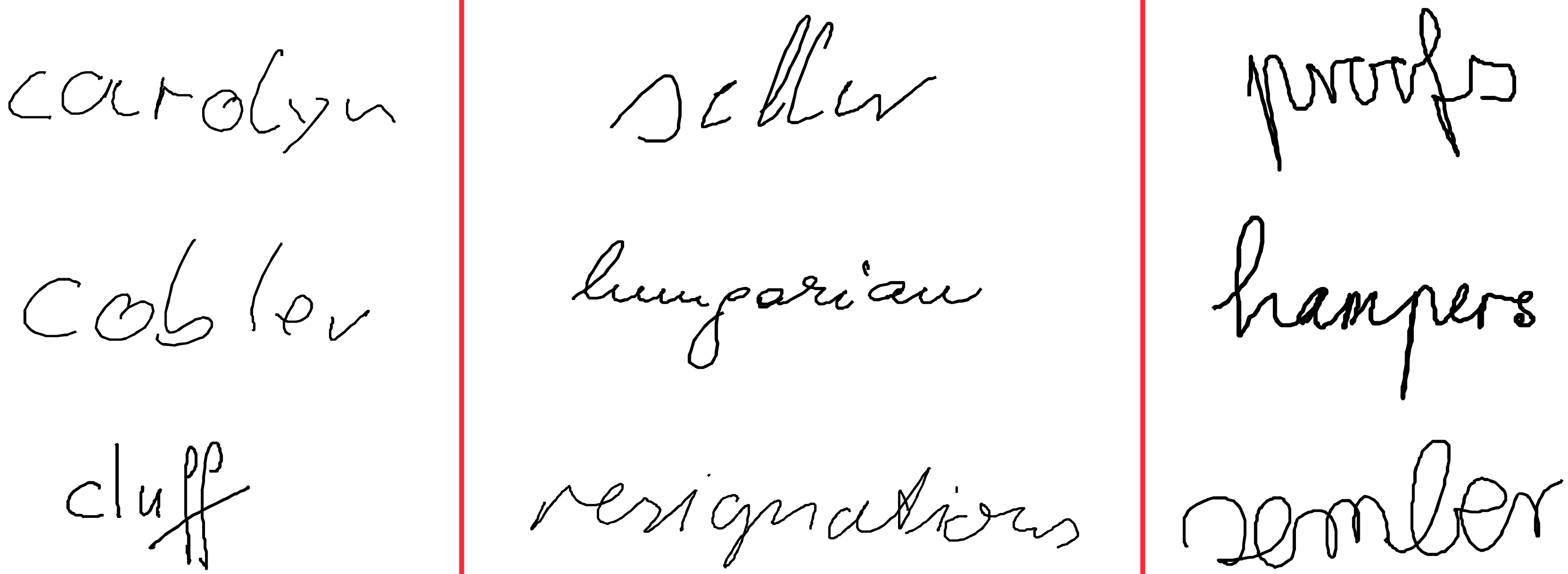
cursive words



unconstrained text

Writing Styles

Printed | Mixed | Cursive



LCD Graphics-Tablets

LCD Graphics-Tablets

LCD Graphics-Tablets

Interactive Systems Labs

Karlsruher Institut für Technologie

On-line Data Capturing

Analog Graphic tablet:
(UltraPad, ArtPad, PenPartner of Wacom

Digital-analog Graphic Tablet:
(Wacom Intuos Serie)

Tablet PC

On-line Handwriting Recognition

- Special Hardware needed (Graphic tablets)
- Interaction with Computer
- Simultaneous Writing und Capturing of Handwriting
- Handwriting stored as point sequence over time (x,y,t).

On-line Information

1. (x,y,t)

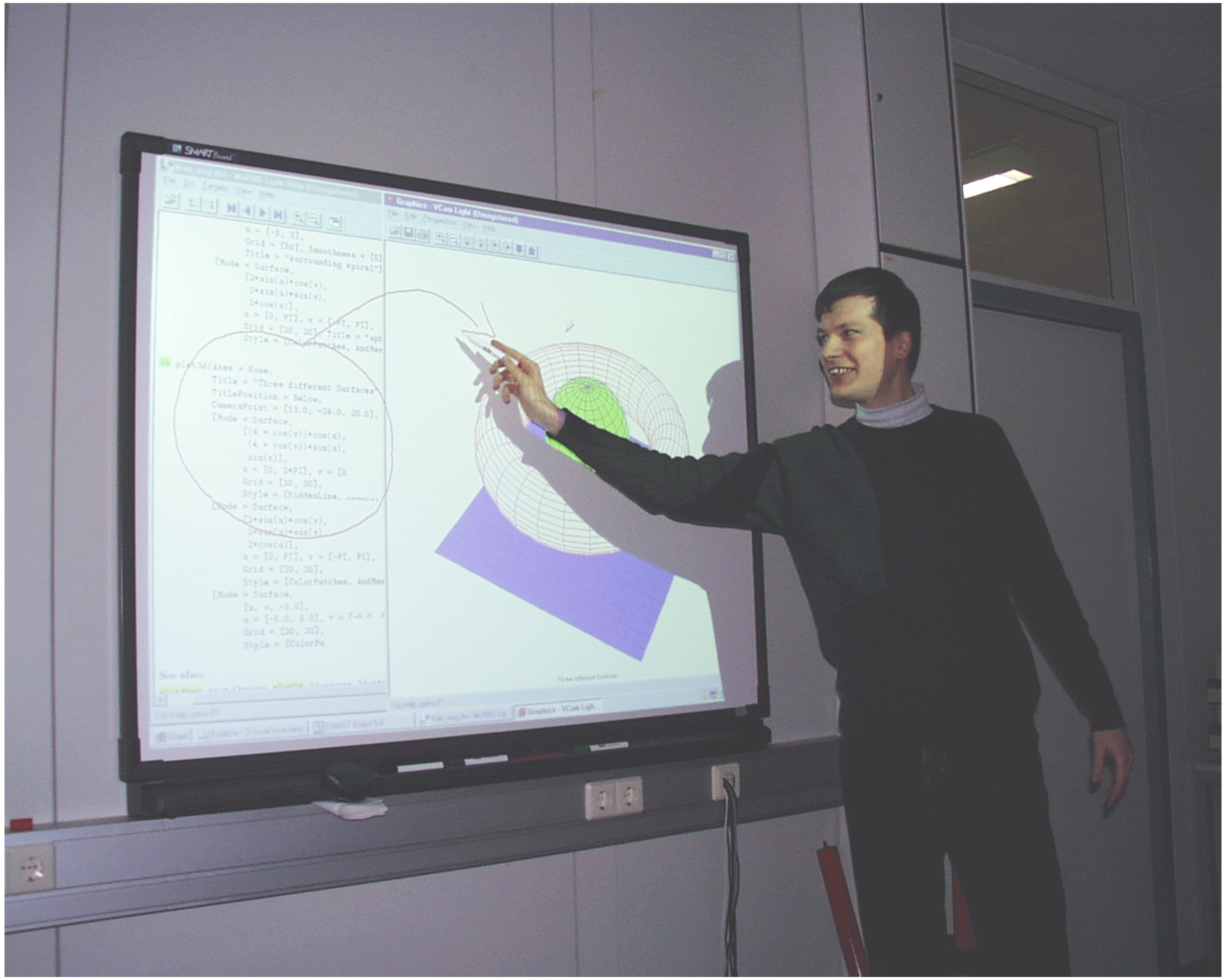
2. Pressure

3. Tilt

4. Pen_down- and Pen_Up

5. Velocity (of 1.)

Handwriting on a Smartboard



Handwriting - [erkennung]

Example: Forensic,
Identification

Forensic

- Finding signatures in a database
- Comparison of signatures

Identification (Biometrie)

Signature Verification:

- measuring ballistic movements
- mostly based on non-visible features of the signature (e.g. pressure, acceleration, ...)
- special pens exists

Other Applications

- Mimio
- CrossPad

Processing Handwritten Material

	Off-line	On-line
Handwriting-recognition	Mail automization, Form Reading (Optical Character Recognition, OCR)	PDAs Pen Computer Graphic-Tablets SmartBoards CrossPad Mimio
Forensic + Identification	Verification and Comparison of signatures (Document analysis)	Signature Verification (Biometrics)

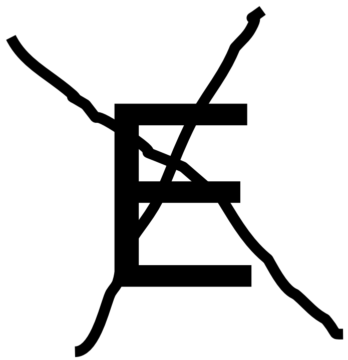
On-line & Off-line

Static Information is independent of the stroke sequence:

E

On-line & Off-line

Dynamic Information Simplifies Segmentation:



On-line & Off-line

Dynamic Information	Trivial →	Static Information
Static Information	Difficult →	Dynamic Information

Ergonomics

Mouse ⟷ Pen

Ergonomics

Ergonomic parameters:

- Wrist rotation
- Angle of Hand at wrist
- Angle between Fingers and Palm
- Distance between Fingers
- Underarm Rotation

Ergonomics

Normal position:

- No rotation of hand at wrist
- No angle of hand at wrist
- No angle between fingers and palm
- no distance between fingers
- No rotation of underarm

Ergonomics

Mouse: very big horizontal rotation at wrist

(specifically rotation in the direction of the small finger)

Ergonomics

Mouse: Vertical rotation at wrist

Moving mouse forward and backward

Ergonomic Advantages of Pens

- Less deviation from the normal position (especially for wrist rotation)
- Makes use of the fingers fine-motoric
- Complete Arm Movements
- Action at Tip of Pen (LCD Tabletts)

On-Line Handwriting Recognition

Interactive Systems Labs

KIT
Karlsruher Institut für Technologie

Design Parameters

- Handwriting Style
- Size of Dictionary
- Writer dependent / independent
- National Particuliarities (e.g., r‘s, ...)
- Left-handed / Right-Handed

History

- Online-Recognition started end of the 50s. (Off-line recognition already earlier)

- Handwriting recognition for mail sorting:
  mid 90s
  (blockletters already earlier)

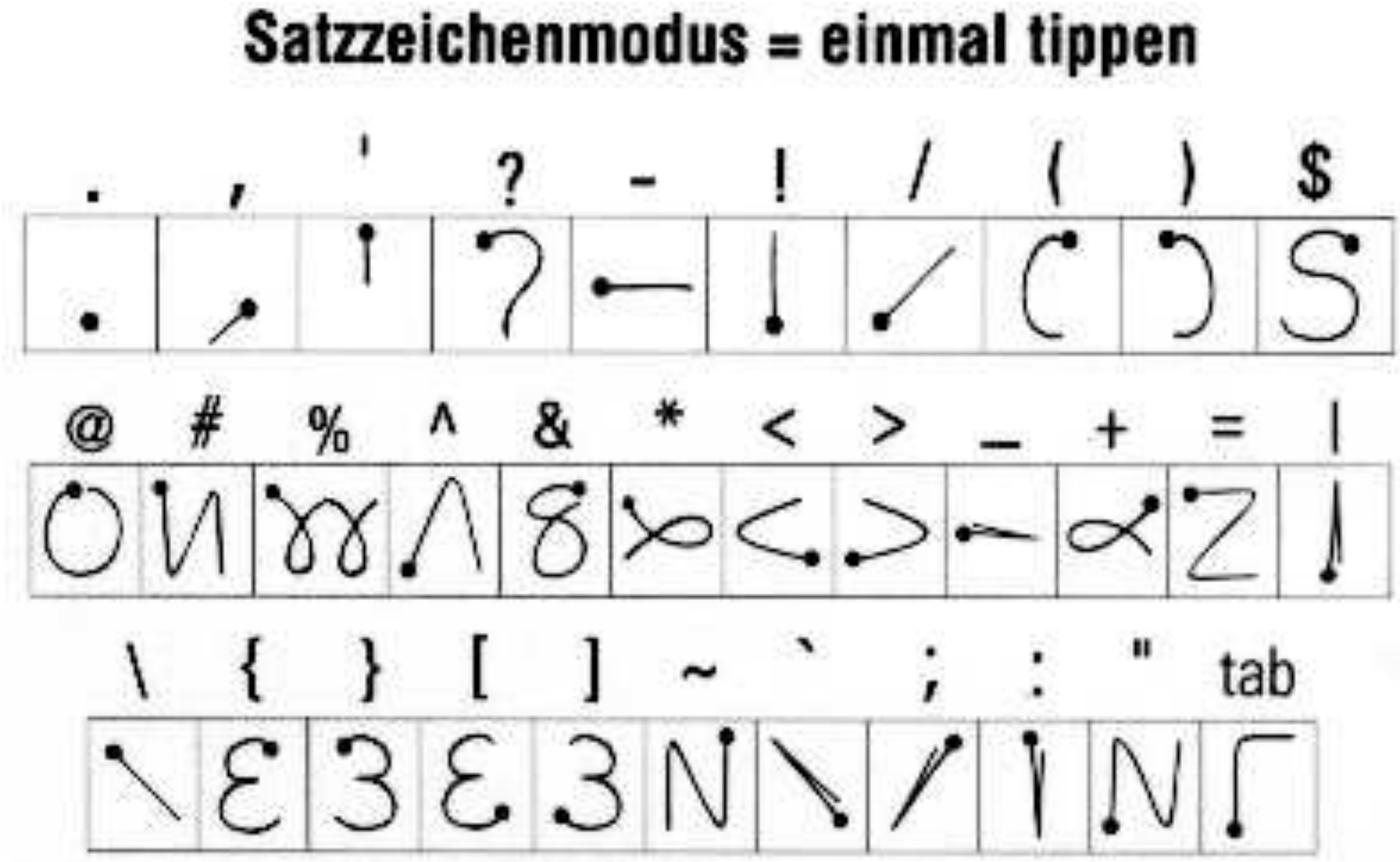
- PDAs with Handwriting Recognition:
  Beginning of 90s (Newton of Apple)

- Palm Graffiti – Speed, Accuracy... Naturalness ?

Graffiti


ß
Ä
Ö
Ü
Leert.
Rückt.
Eingabet.
Groß
Feststell
(•) Der dicke Punkt
ist der Anfangspunkt.

Graffiti



Graffiti

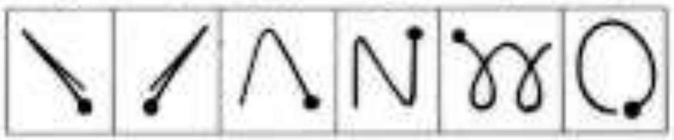
Zeichen mit Akzent

Zur Eingabe der folgenden Buchstaben mit Akzent:

à á â ã ä å è é ê ë ì í î ï
ò ó ô õ ö ù ú û ü ÿ ý ñ,

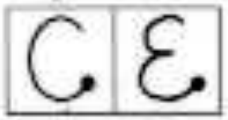
Schreiben Sie zuerst den jeweiligen Buchstaben und dann den Akzent wie folgt:

` ´ ^ ~ ¨ °



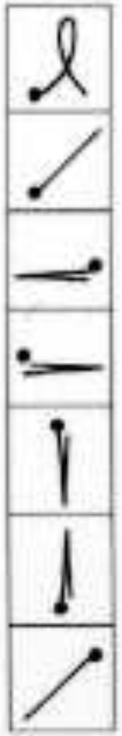
Die folgenden Zeichen können mit Graffit ohne Umschalten geschrieben werden:

ç æ



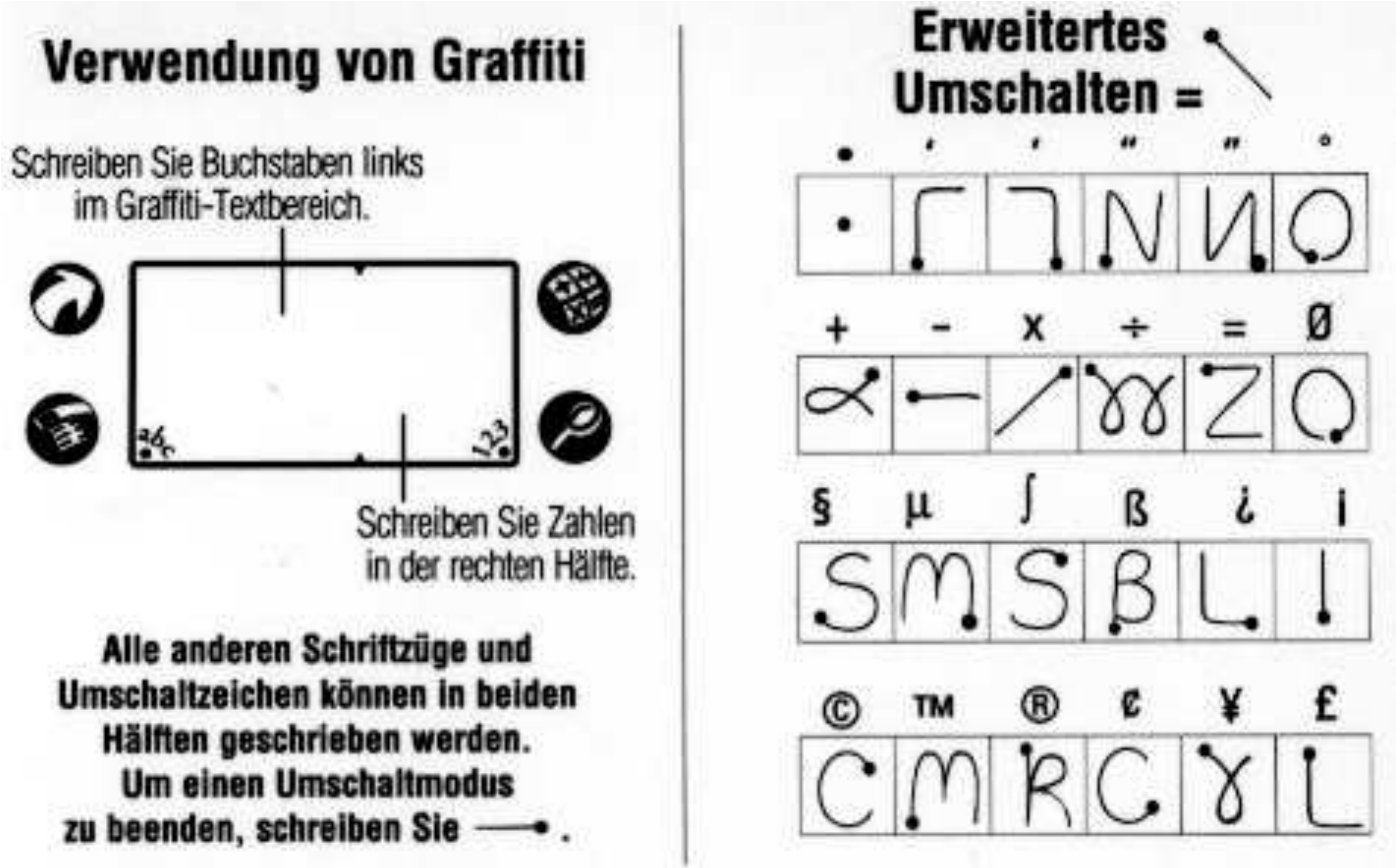
Besondere Schriftzüge

Einzelheiten siehe PalmPilot-Gebrauchsanweisung.



- ShortCuts
- Befehlszeichen
- Cursor nach links
- Cursor nach rechts
- Nächstes Feld (Adreßbearbeitungs-bildschirm)
- Vorheriges Feld (Adreßbearbeitungs-bildschirm)
- Eintrag öffnen (Adreßbearbeitungs-bildschirm)

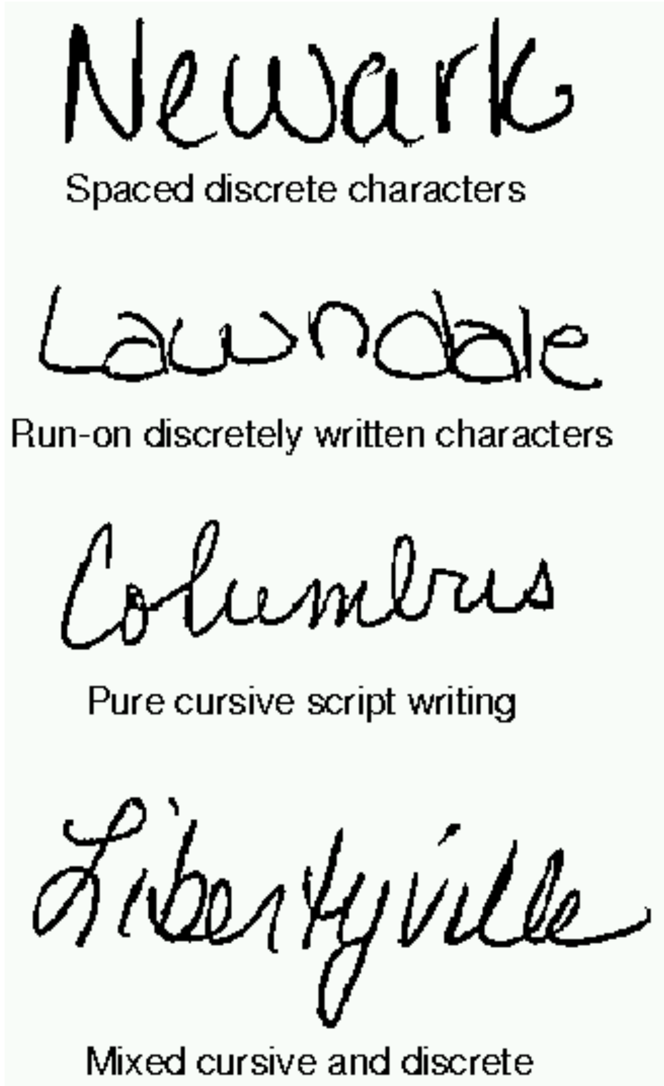
Graffiti



Verwendung von Graffiti
Schreiben Sie Buchstaben links im Graffiti-Textbereich.
Schreiben Sie Zahlen in der rechten Hälfte.
Alle anderen Schriftzüge und Umschaltzeichen können in beiden Hälften geschrieben werden. Um einen Umschaltmodus zu beenden, schreiben Sie —• .
Erweitertes Umschalten =
+ - x ÷ = Ø
§ µ ∫ ß ¿ ¡
© ™ ® ¢ ¥ £

Handwriting types

Classification of Handwriting Types According to Tappert (IBM):



Recognition Rates

Recognition of single symbols:

Npen++:	0-9	96.5%
	A-Z	92.7%
	a-z	91.1%

Recognition Rates

Recognition of words:

On-line recogniton rate is higher than off-line recognition rates:

Off-line: 95% on 500 words
On-line: 95% on 5.000 words
90% on 50.000 words

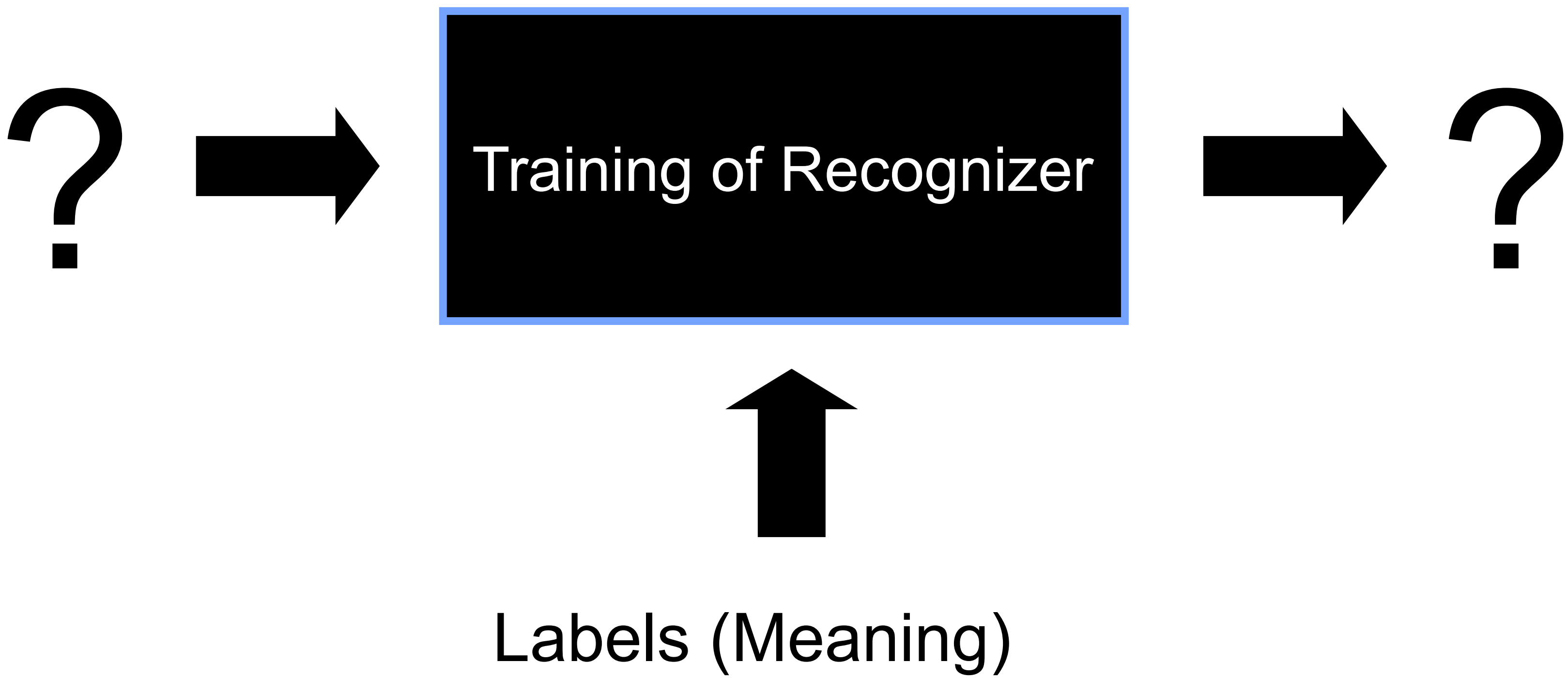
Recognition Rates

Sentence Recognition:

Robust recognition of word sequences is not yet solved
(Npen++: 86,6% with 20.000 Words).

Training and Classification

Training



Training

Manuelle Segmentation:

Training

- Strokes
- Letters
- Words
- Sentences
- Texts

Training

Increasing Dictionary, Greater Specificity

- Strokes
- Letters
- Words
- Sentences
- Texts

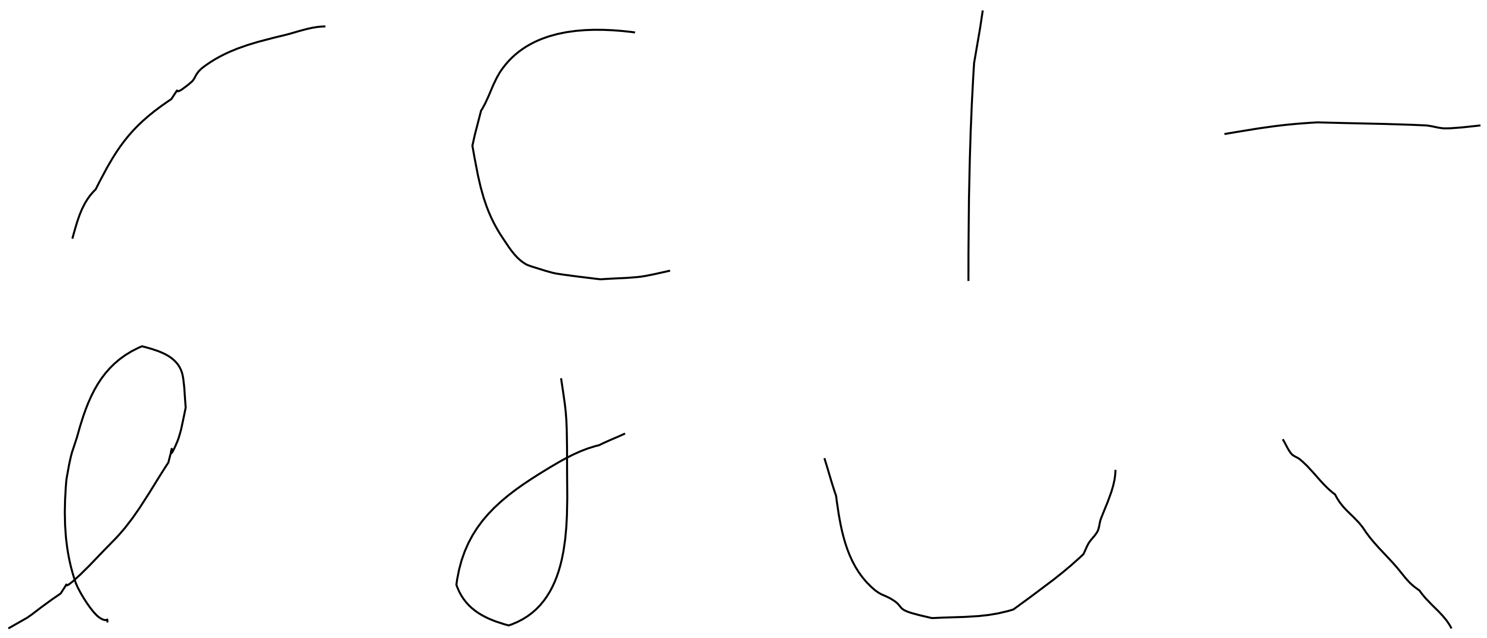
Increasing cost of Labeling, More Training Data

Strokes Level

- Analysis-by-Synthesis
- Rule-based Recognizer
- Syntactic Recognition
- Symbolic learning

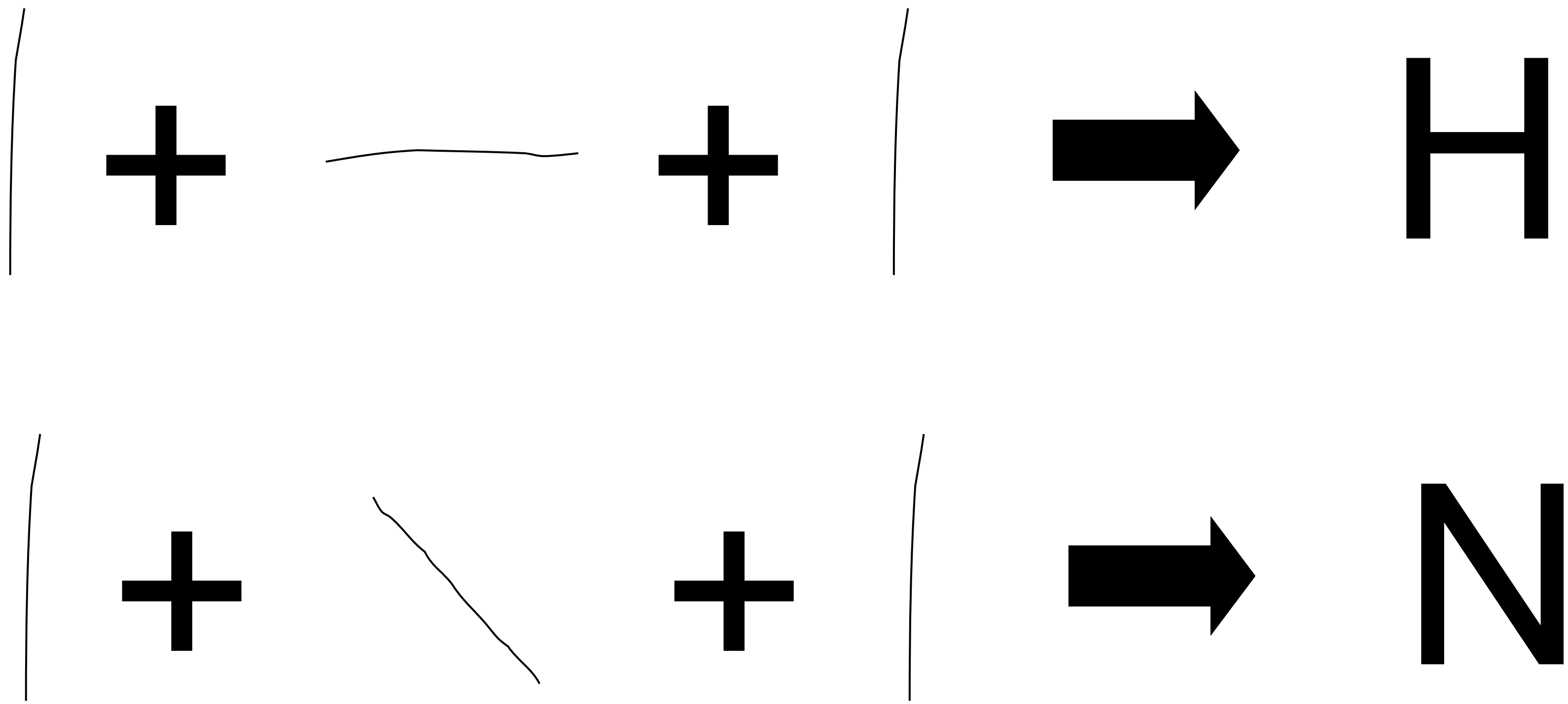
Stroke Level

Decomposition / Identification of atomic units, building blocks



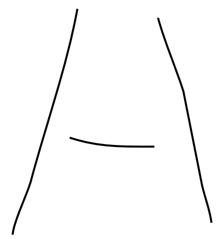
Strokes Level

Finding Rules

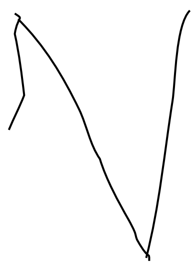


Stroke Level

Disadvantage: complex rule bases



A oder H ?



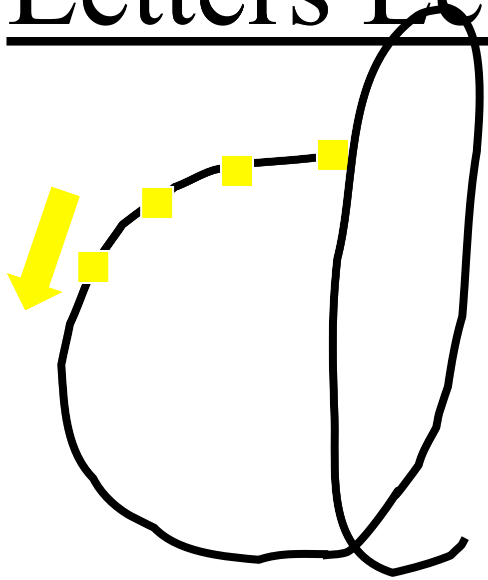
N oder V ?

Stroke Level

- Training of rules is difficult.
- Syntactic Approaches did not prove to work well

Letters Level

Training:



Sequence of (local) Feature vectors

Class „c“

Letters Level

Features: Direction r,
Slope s,
Curvature k,
...

→ (r,s,k, ...)

No methods exist to find optimal features.

Mathematical methods exist to rank features according to their relevance.

Word Level

Word Level Training (holistic approaches):

- Recognition on the whole word
- No segmentation on letter level

Advantage: No Segmentation, Detailed Modeling

Disadvantage:
- Recognition restricted to special dictionary
- Extension of dictionary requires new training
- Less training data per class

Training

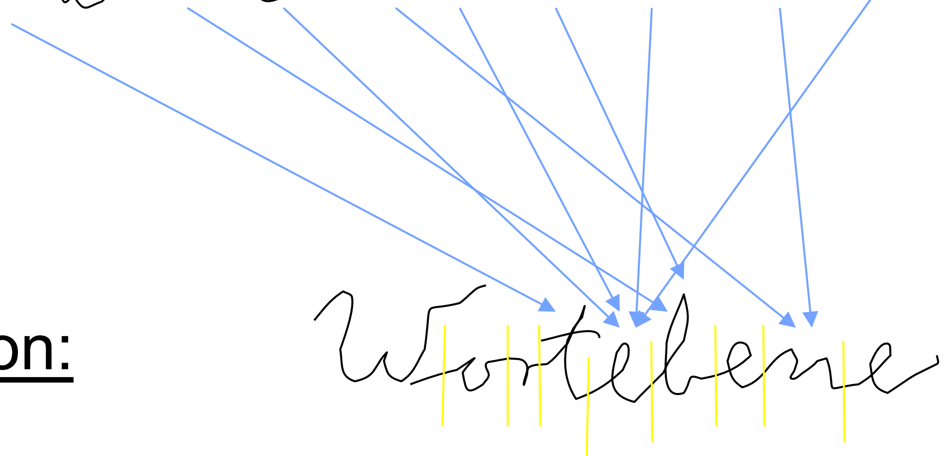
Use of unsegmented data on word and sentence :

Idea: Use pre-trained recognizer to segment higher levels

Training

Manual Segmentation:

Automatic Segmentation:



Classification

Classification

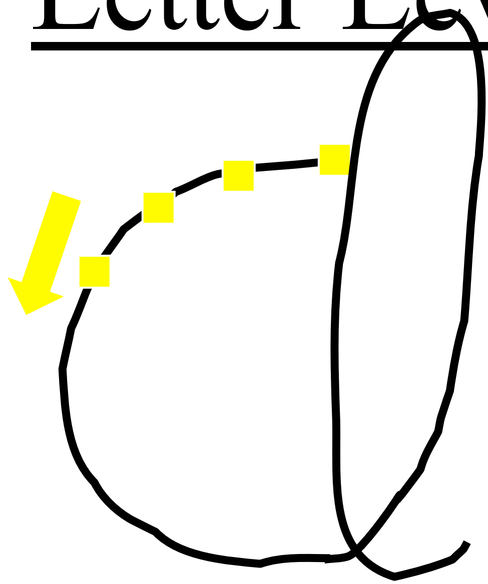
Classification

Class:

Group of feature vectors, which are typical for a part of a letter
e.g. all feature vectors, which are typical for the beginning of „d“

Letter Level

Training:



Sequence of (local) feature vectors

Class „d“

Letter Models

Letter models:
Representation of typical sequences of feature vectors for letters

Segmentation

Segmentation

Classification of Words and Texts requires Segmentation:

- explicit Segmentation
- implicit Segmentation

Explicit Segmentation

Segmentation performed prior to recognition:

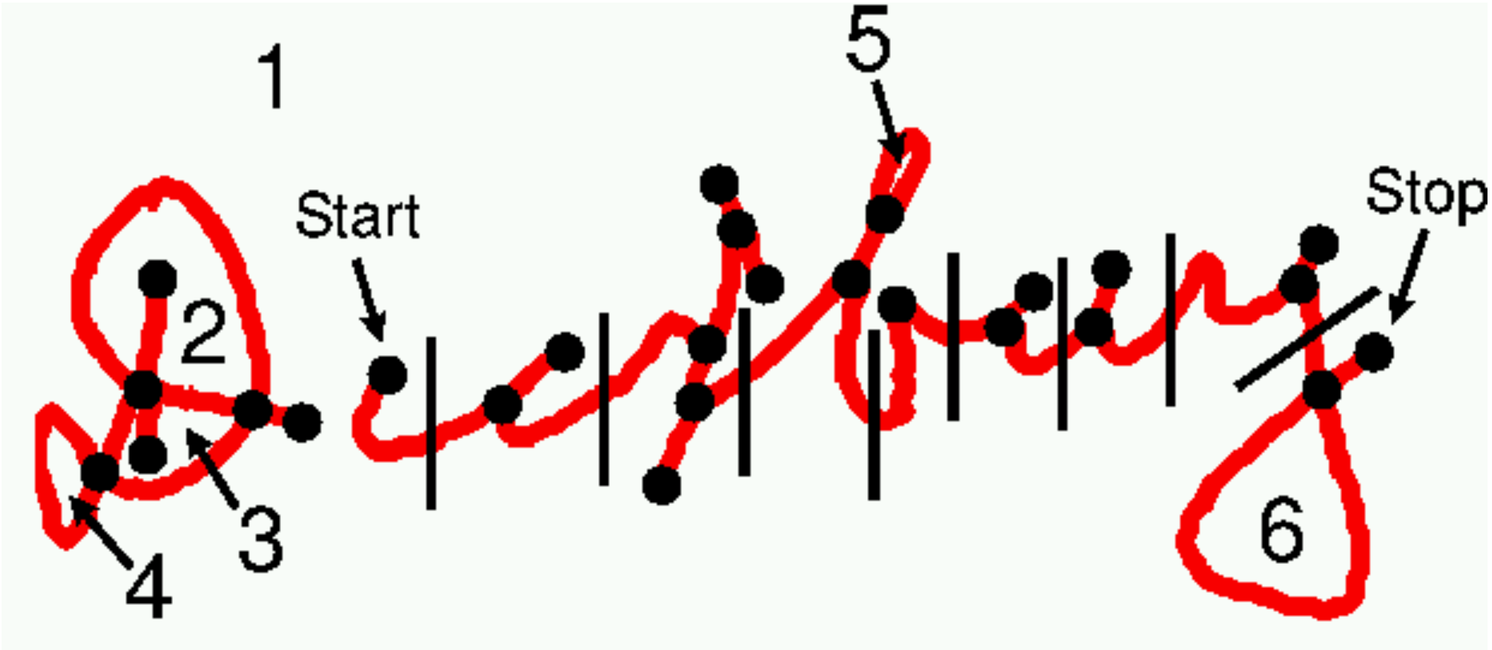
- spatial features (e.g. stroke distance)
- temporal features
  (e.g. temporal distance between strokes)

Explicit Segmentation

Disadvantages:

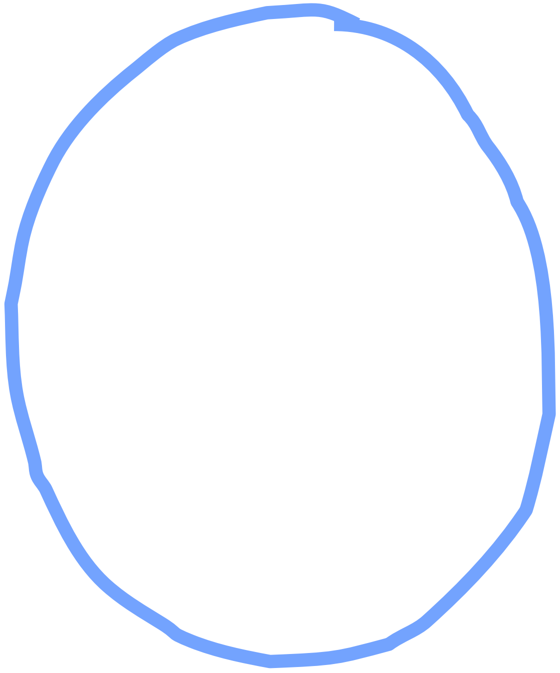
• Segmentation is difficult to find
• Segmentation errors lead to classification errors

Explicit Segmentation



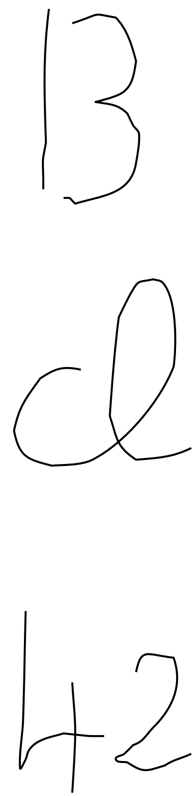
Implicit Segmentation

Context



Context

Context influences segmentation:



Recognition of Single Symbols

Recognition of single symbols

Problem:

- Deal with Deviations
- Find best Match

Dynamic Programming:

- Elastic Matching
- Editing distance
- Levenstein-Distance

Elastic Matching

INDUSTRY ⟷ distance ⟷ INTEREST

Operations:

- Deleting
- Inserting
- Replacing

Elastic Matching

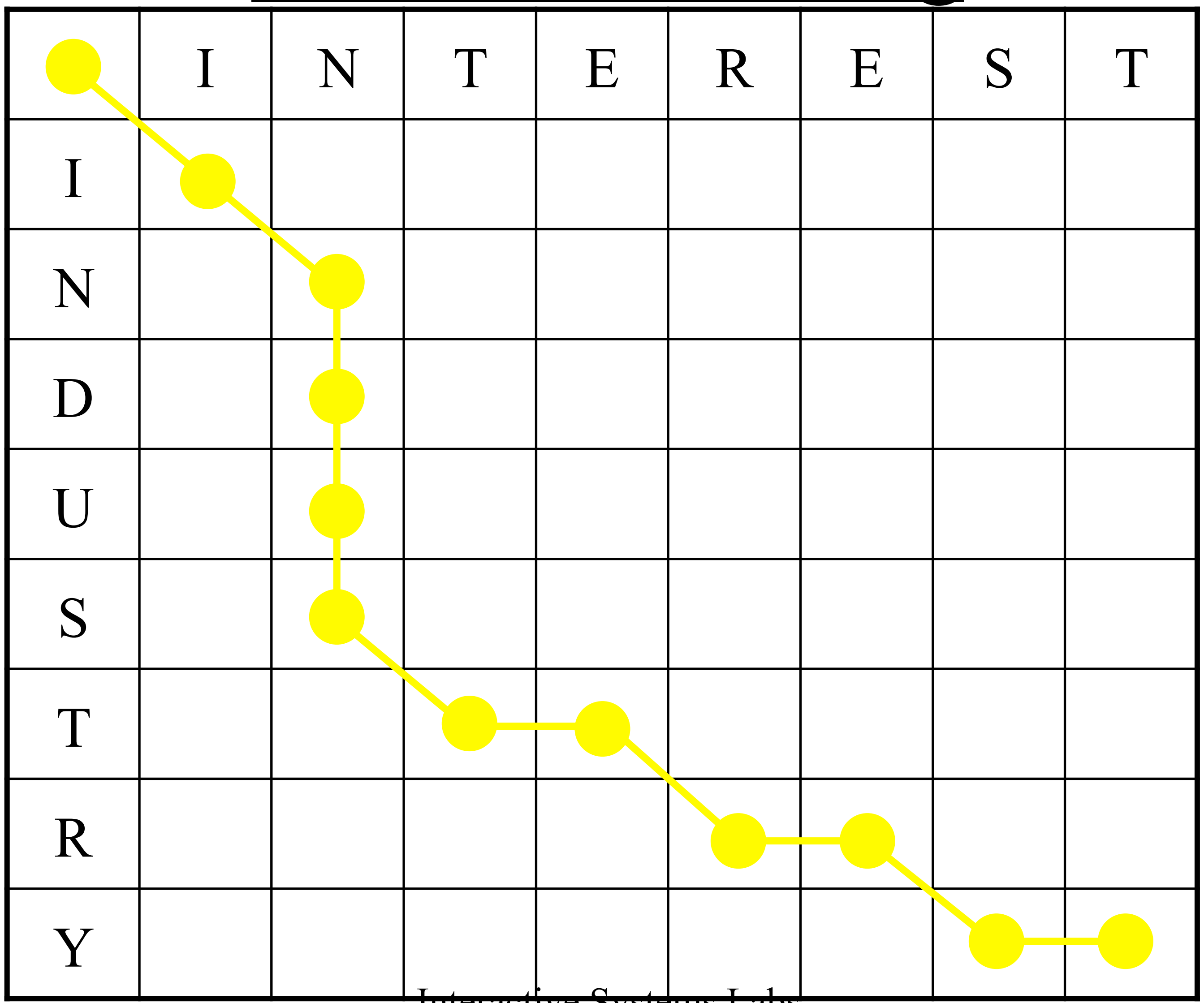
INDUSTRY	
	Deletion of D
INUSTRY	
	Deletion of U
INSTRY	
	Replace Y with S
INSTRS	
	Insertion of E
INSTERS	
	Insertion of E
INSTERES	
	Deletion of S
INTERES	
	Insertion of T
INTEREST	

Elastic Matching

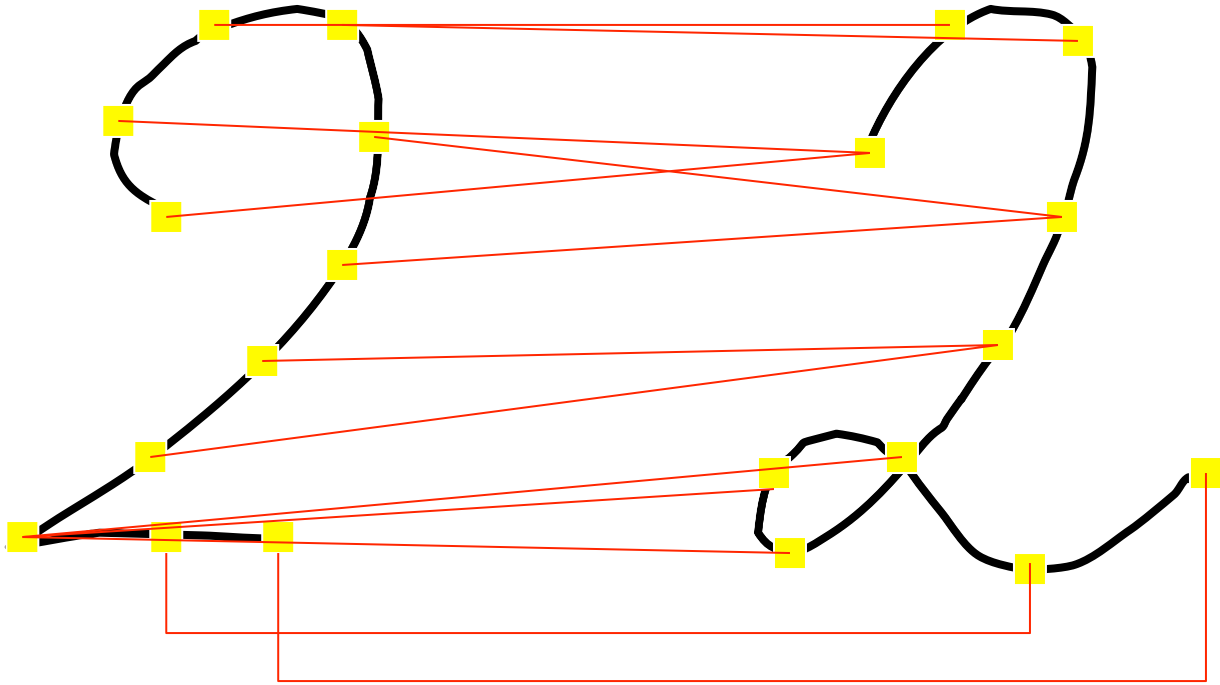
$$d(a,b) = d(a^m, b^n)$$

$$d(a^i, b^{j)} = \min \begin{array}{c} d(a^{i-1}, b^j) + w(a_i, \varnothing) \\ d(a^{i-1}, b^{j-1}) + w(a_i, b_j) \\ d(a^i, b^{j-1}) + w(\varnothing, b_j) \end{array}$$

Elastic Matching



Elastic Matching



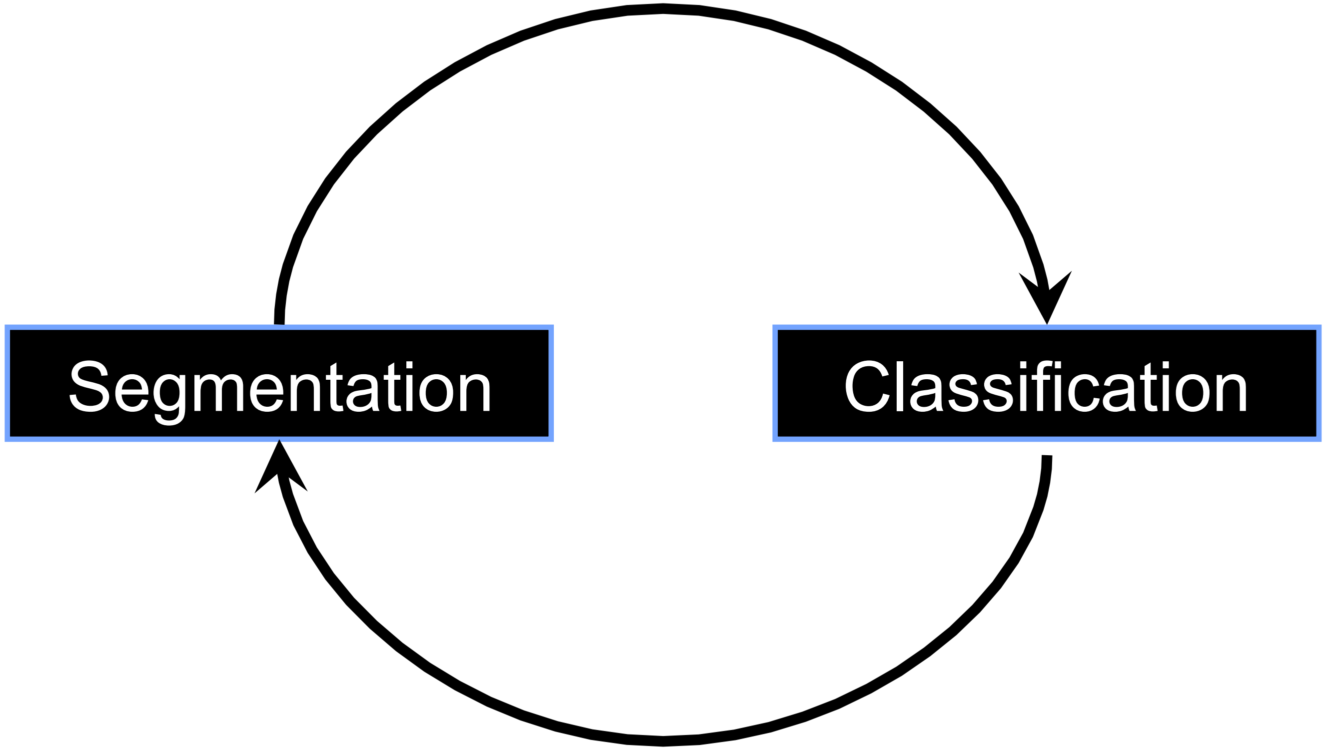
Word Recognition

- Problem
  - Word is a Sequence of Subword Units:
    A Sequence of Letters
  - Do not want to Train at the Word Level
    - Number of Units
    - Training Data Requirement
    - Problem of Adding New Words

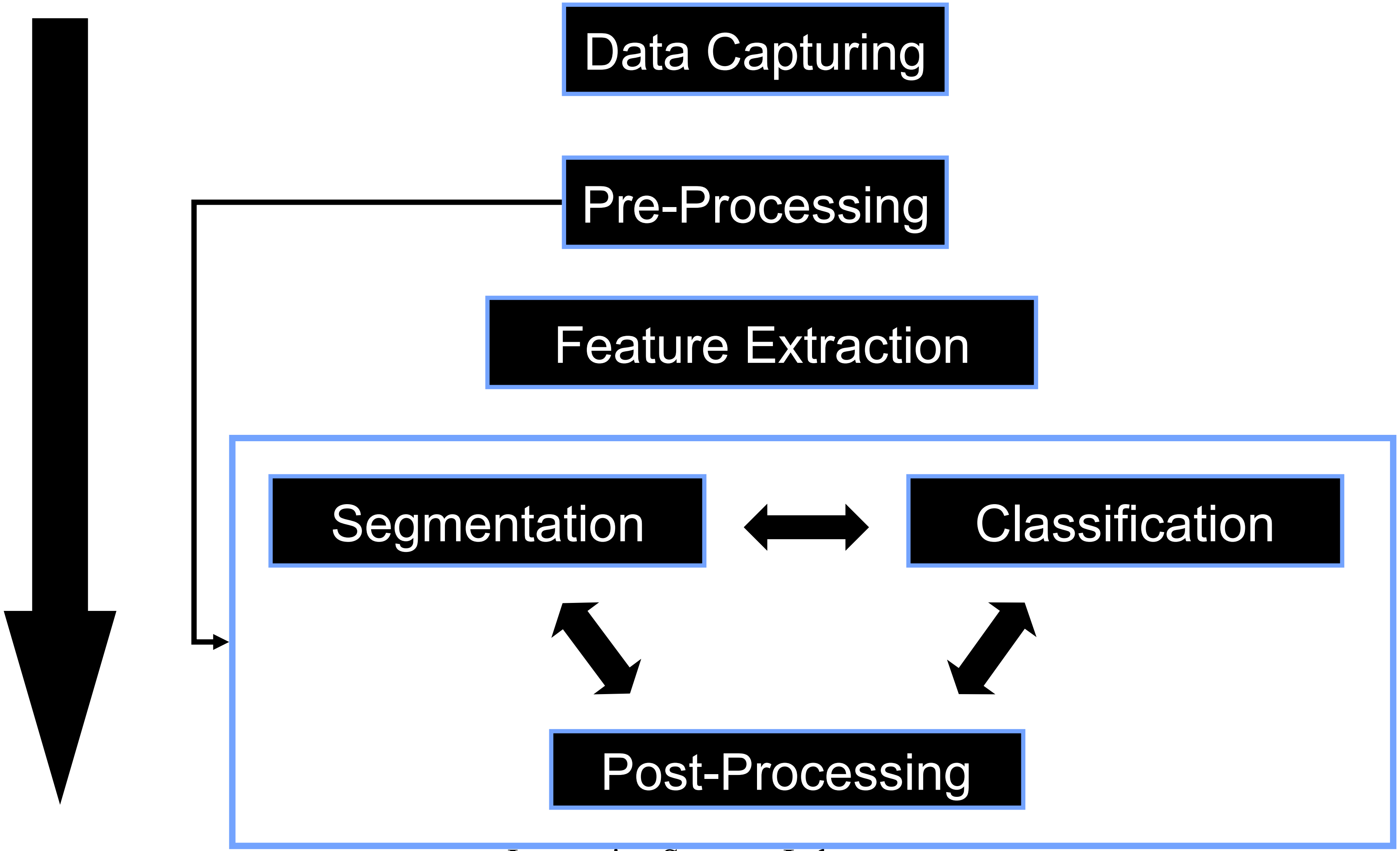
Word Recognition

- Data Capturing
- Pre-Processing
- Feature Extraction
- Segmentation
- Classification
- Post-Processing

Implicit Segmentation

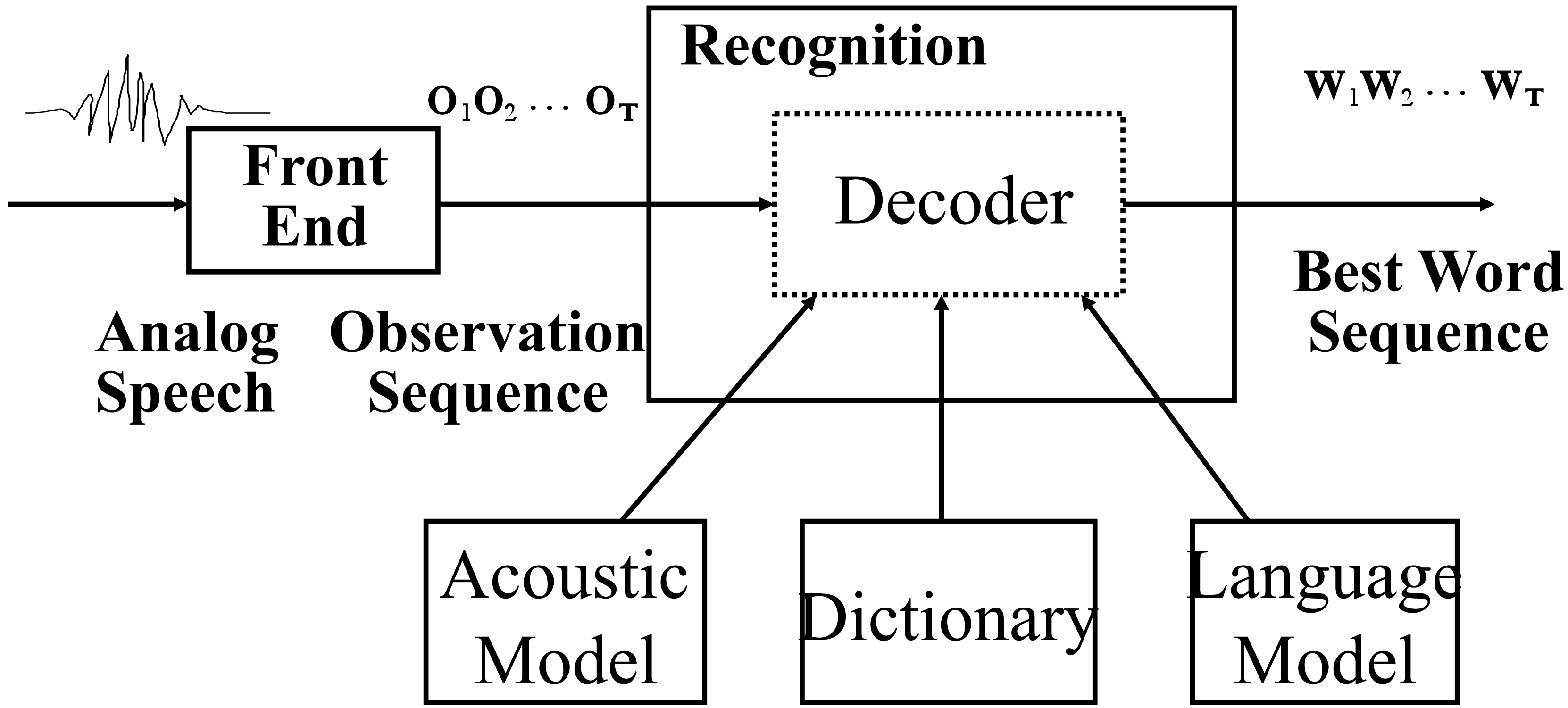


Word Recognition



Speech Recognition

- Recognizer Components:



Word and Letter Models

Techniques that use letter models:

- Hidden Markov Models (HMMs)
  - Model Training
  - Finding the best path (Segmentation)
  - Finding the probability of a word

- Multi-State Time Delay Neural Networks (MS - TDNNs)

Words and Letter Models

Techniques to help guide the search:

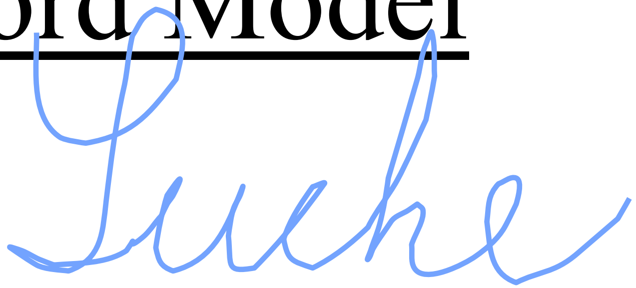
- Language Models (e.g. Bi- or Trigrams)
- Dictionaries

Word Model

Suche

Model A	-------------------------------
Model S	++++------------------------
Model U	-----++++++---------------
Model C	-----------++++-------------
Model H	--------------+++++--------
Model E	-----------+++-----++++++
Model I	-----------+++---------------
Model L	---------------+++-----------

Word Model



Model A	---------------------------------
Model S	++++-------------------------
Model U	-----+++++------------------
Model C	-----------++++-------------
Model H	---------------+++++--------
Model E	-----------+++-----++++++ Best Path
Model I	-----------+++---------------
Model L	---------------+++-----------

Dictionaries

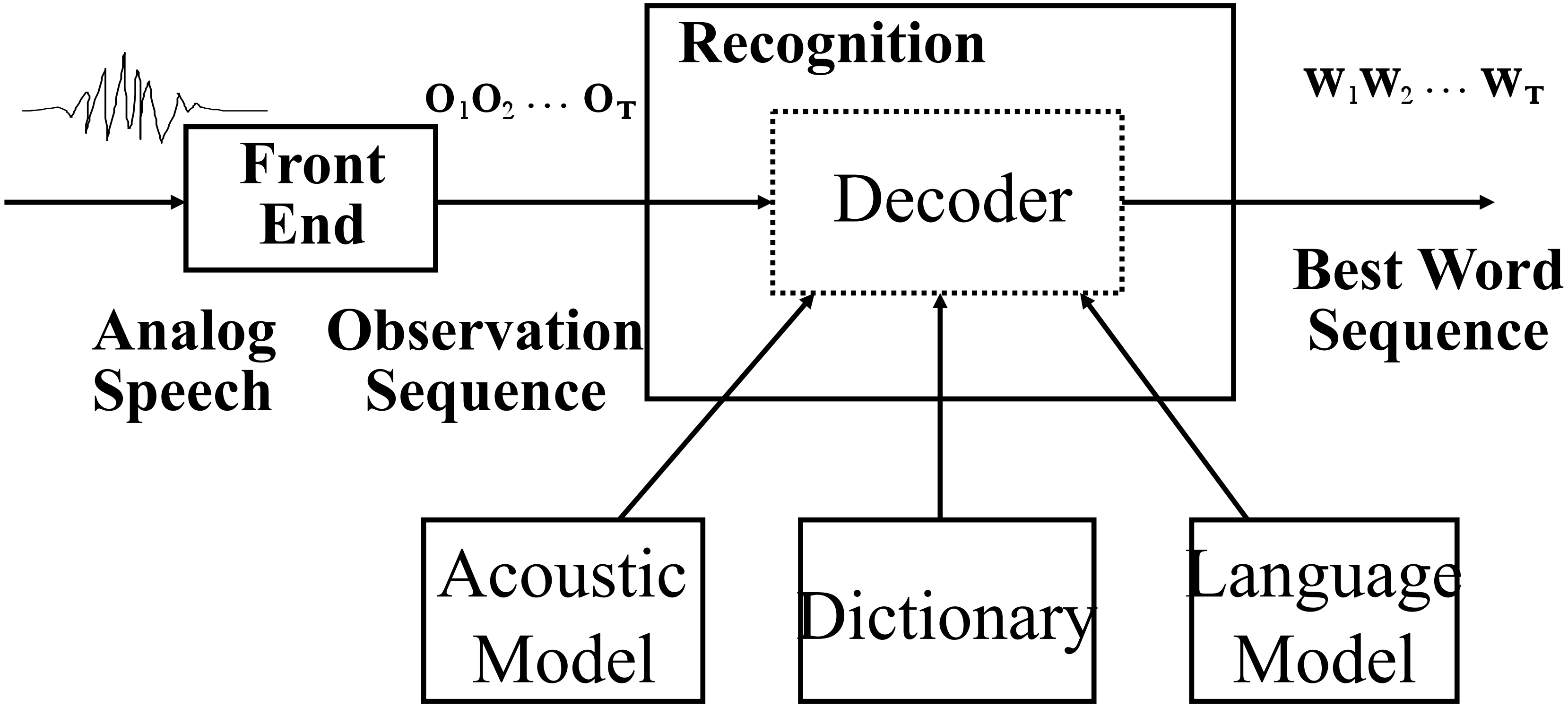
Model S | Model U | Model C | Model H | Model E

Computation of word probability through concatenation of letter models

Disadvantage: classificator only recognizes words from the dictionary

Speech Recognition

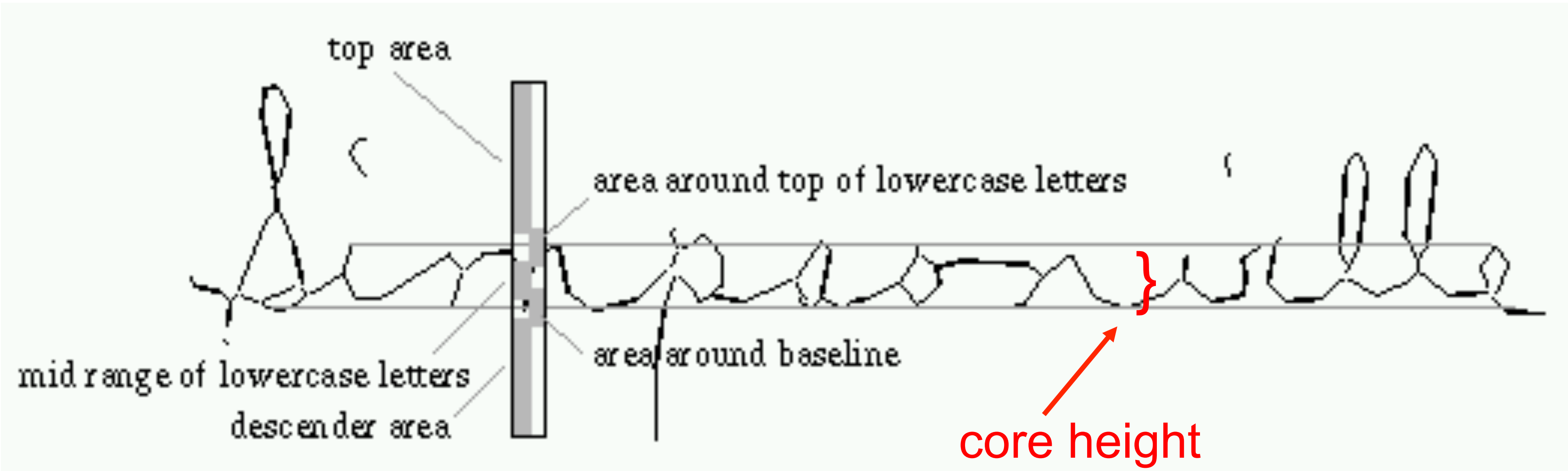
- Recognizer Components:



Pre-Processing

Size Normalization

Base-line computation:



Pre-Processing

$$b_0, \ldots, b_n \quad b_i \in IR^d$$

$$b(t) := \sum_{i=0}^{n} b_i B_i^n(t), \quad t \in [0,1]$$

$$B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i}, \quad i = 0, \ldots, n.$$

Pre-Processing

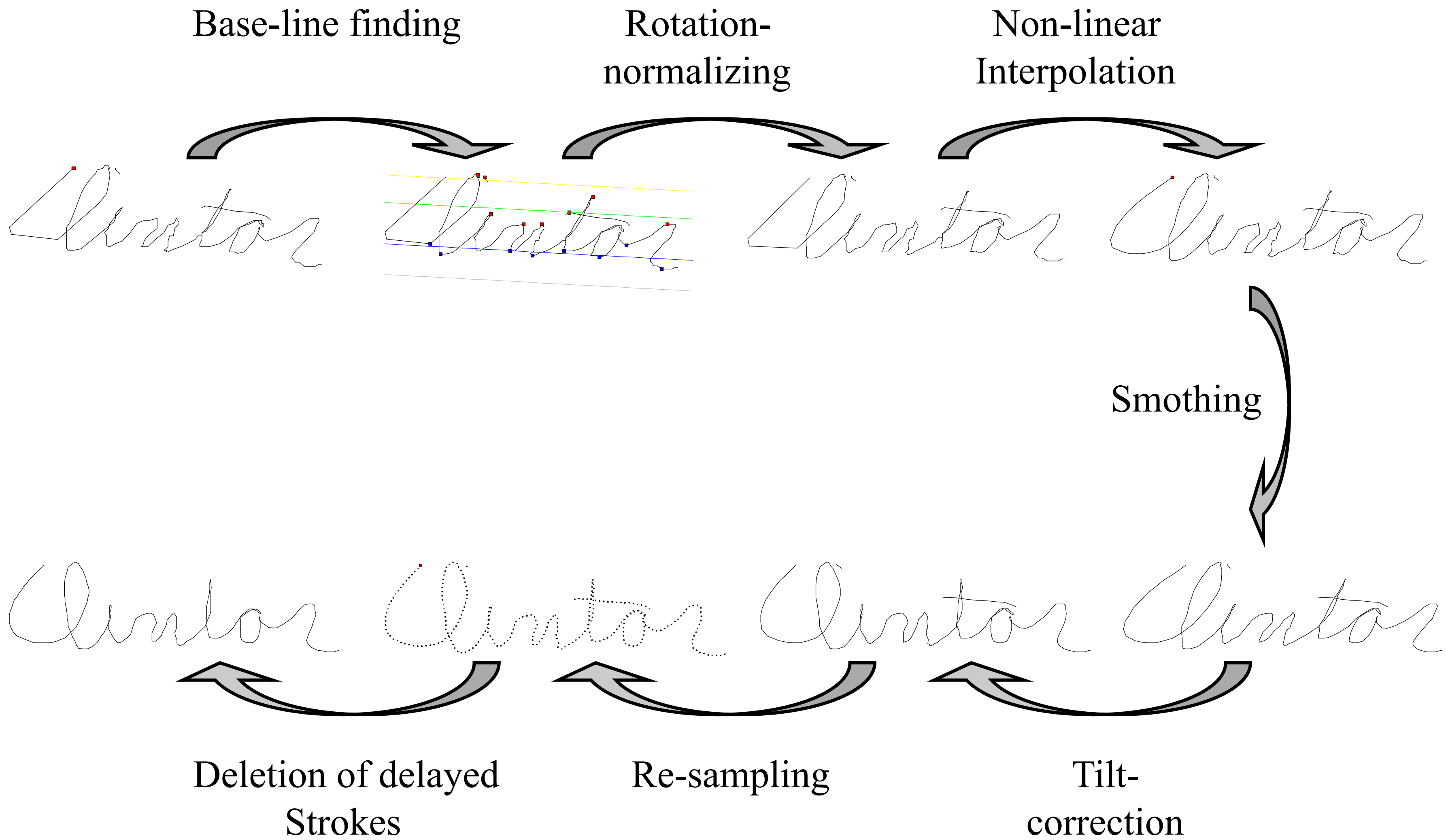
Re-sampling:

temporally equidistant points

↓

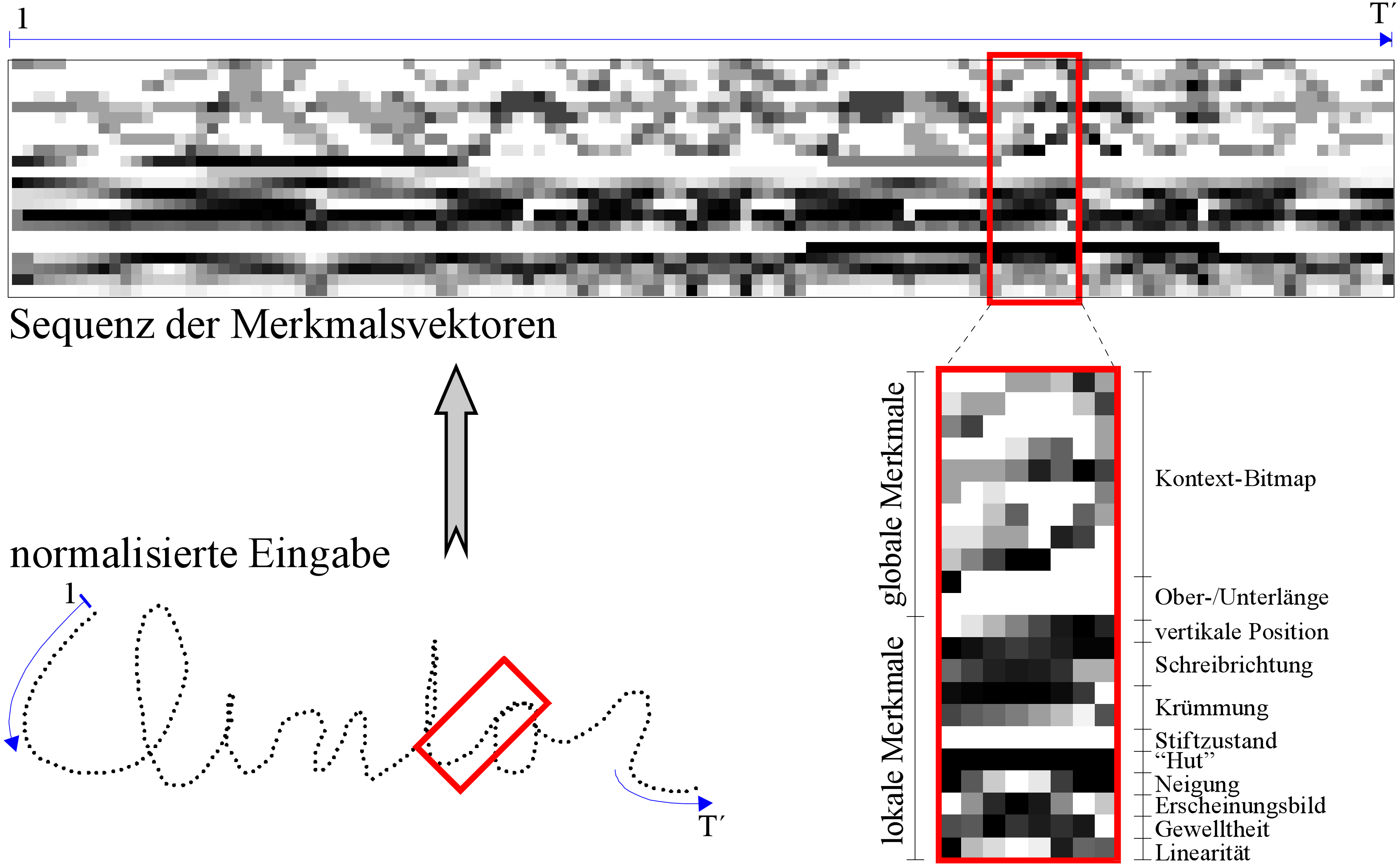
spatially equidistant points

Pre-Processing



MS-TDNN Architecture

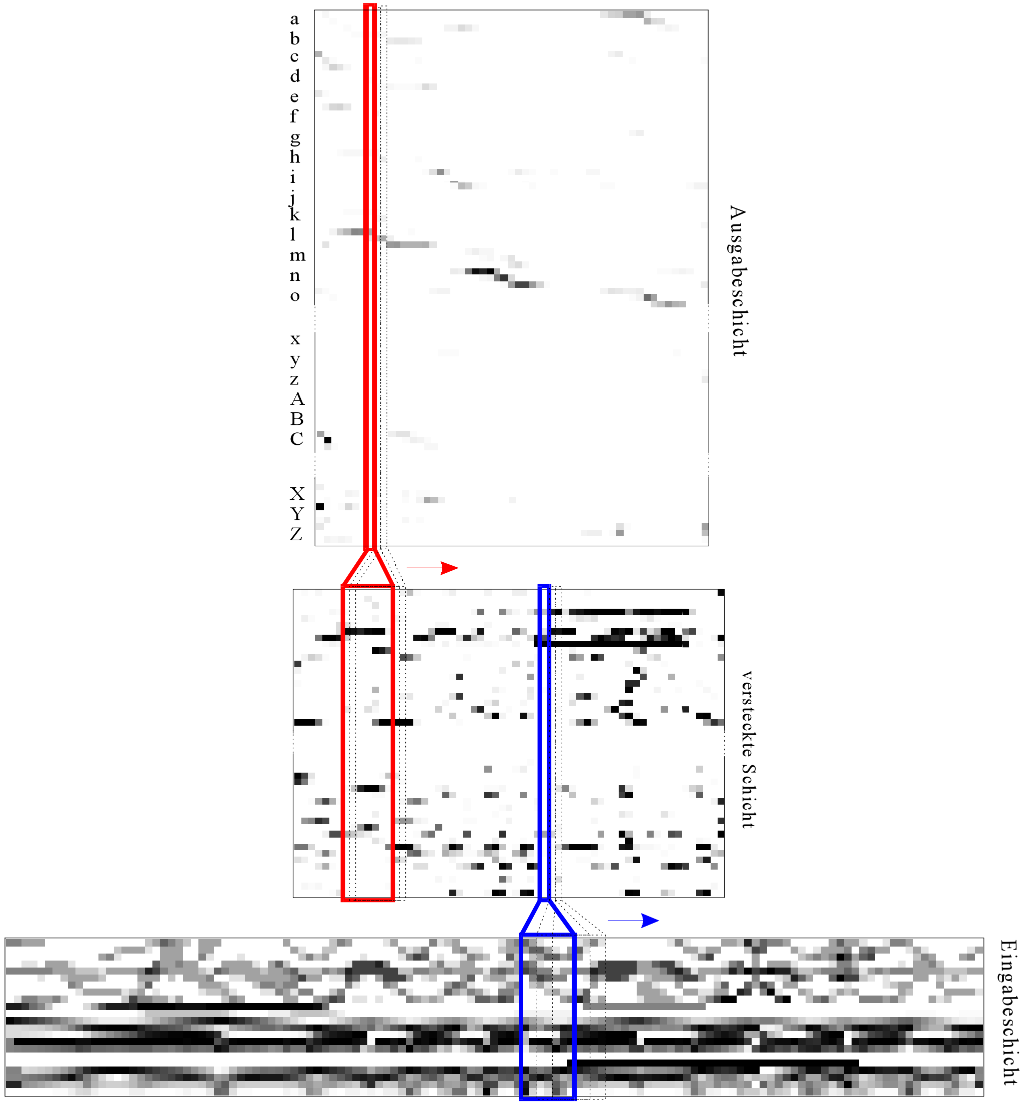
MS-TDNN Architecture



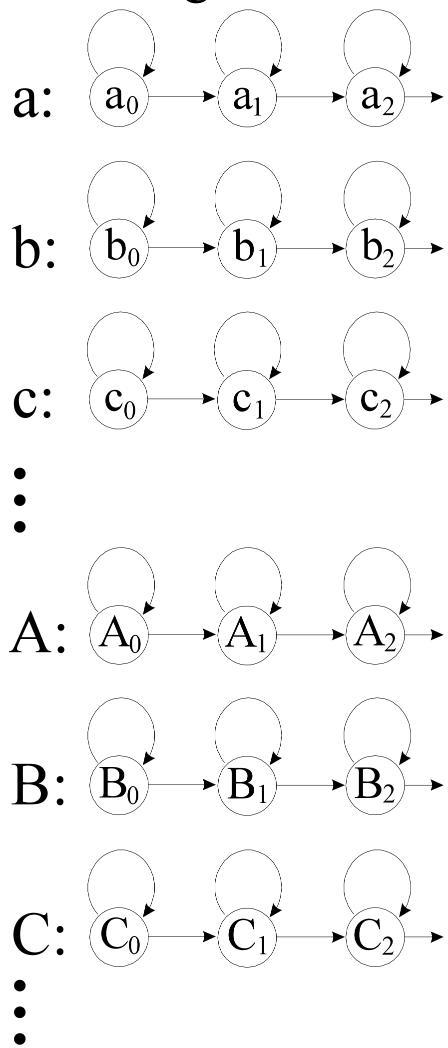
Interactive Systems Labs

Karlsruher Institut für Technologie

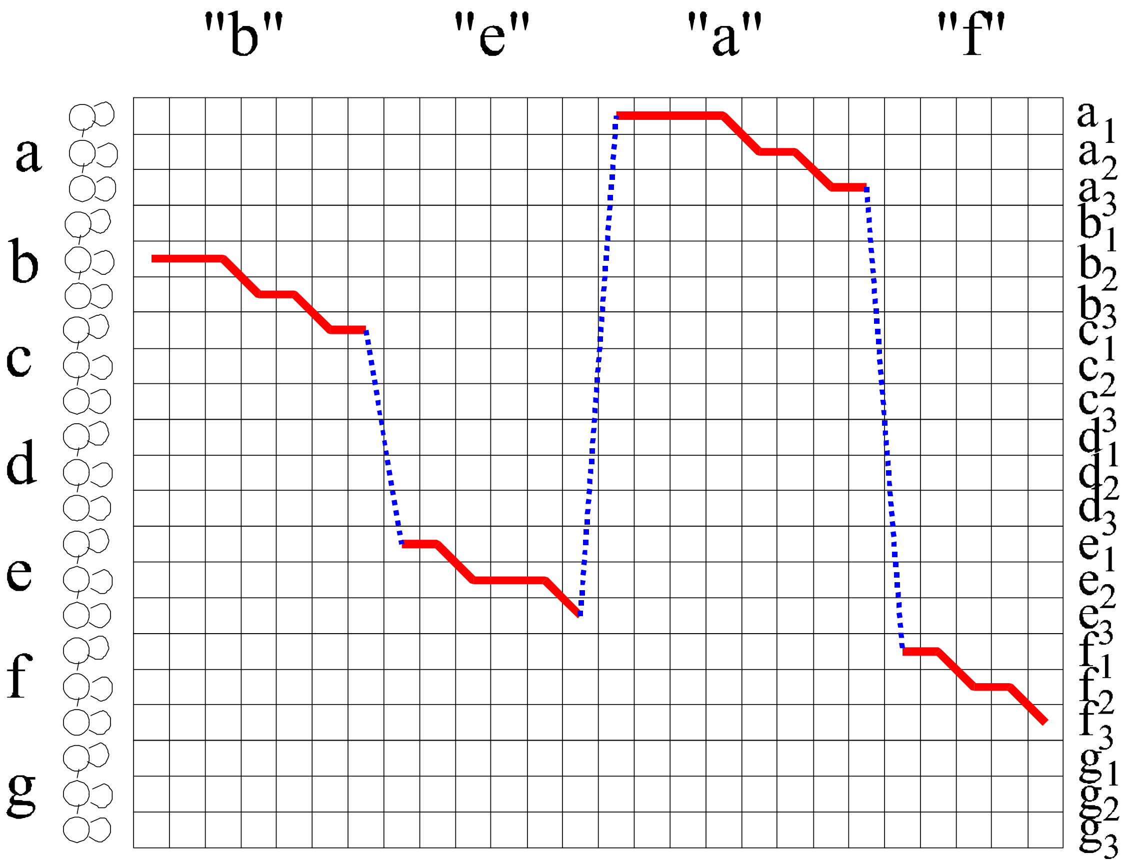
MS-TDNN Architecture

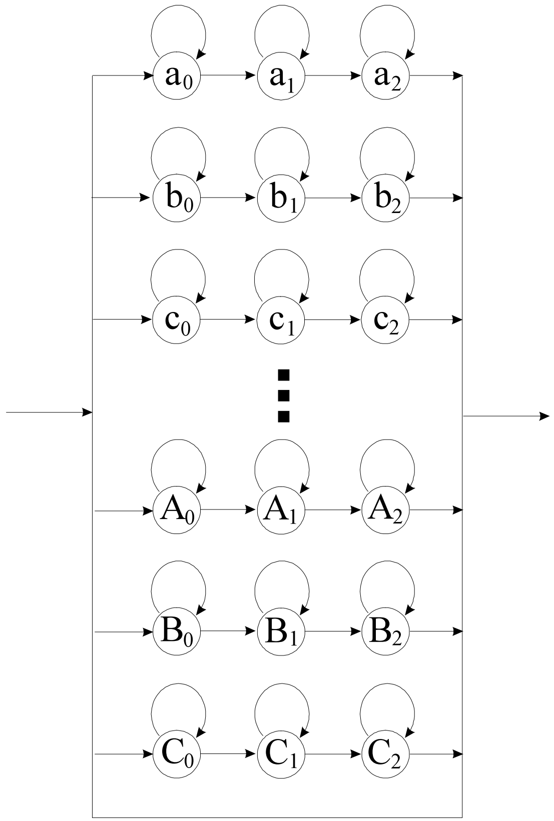


Modeling:



MS-TDNN Architecture

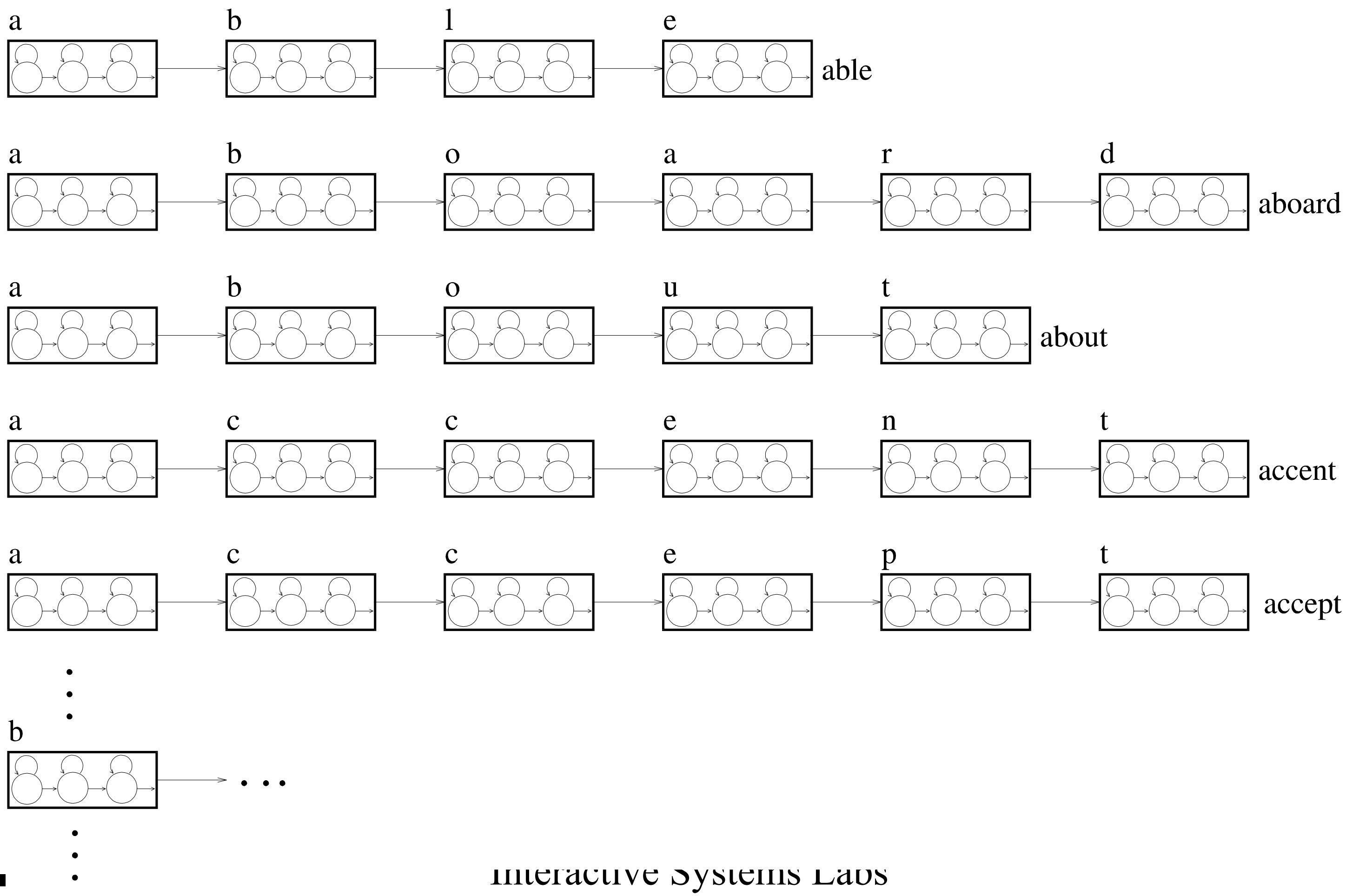




Use of Language Models and Dictionaries

MS-TDNN Architecture

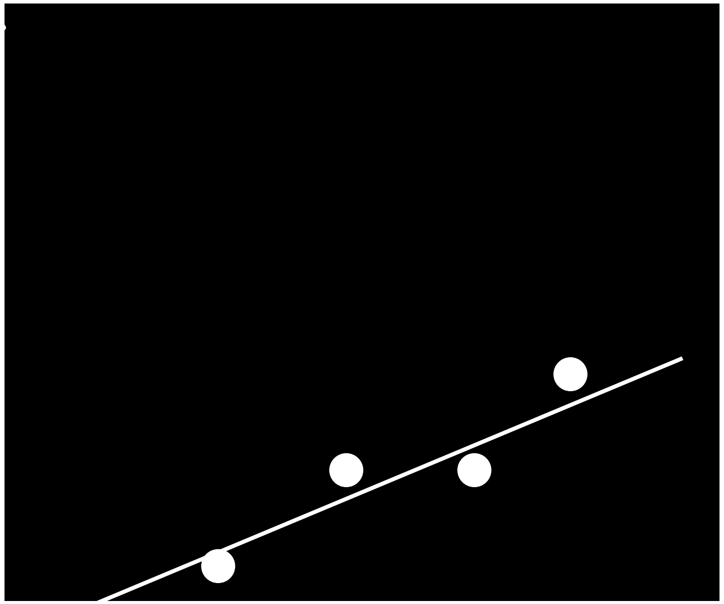
Each word is represented by concatenating its letter models.

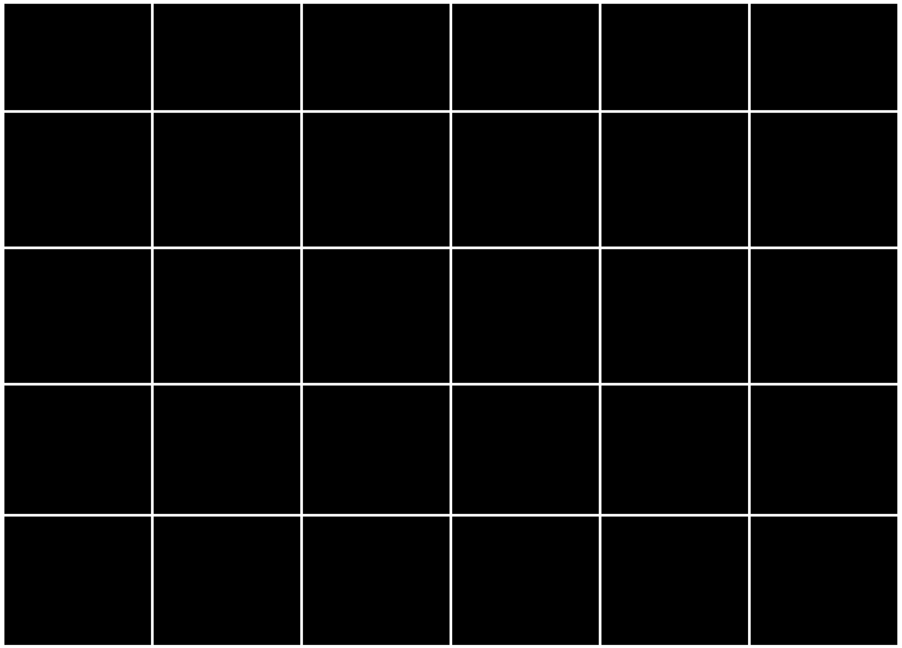


On-line & Off-line

Connection of two Pixels with a straight line

Idea: Use the horizontally right or diagonally situated pixel, which is closest to the exact





On-line & Off-line

$$p_1 = (x_1, y_1), p_2 = (x_2, y_2)$$

$$0 \leq y_2 - y_1 \leq x_2 - x_1 \quad x_i, y_i \in IN$$

p_1 p_2

Online & Offline

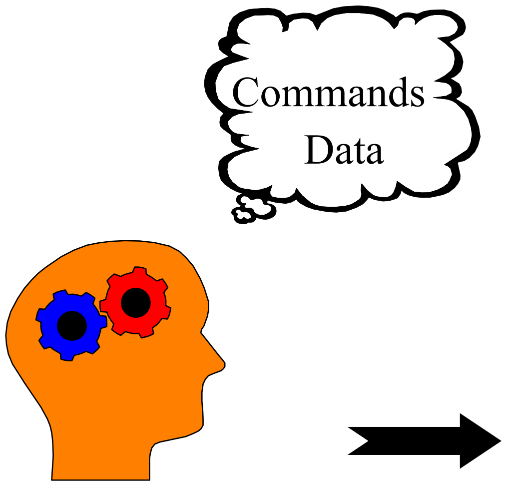
$$\Delta_x := x_2 - x_1;\quad \Delta_y := y_2 - y_1;$$
$$x := x_1;\quad y := y_1;$$
$$c_y := 2 * \Delta_y;\quad \Delta := c_y - \Delta_x;\quad c_x := \Delta - \Delta_x;$$

$$(x, y)$$
$$x := x + 1;$$
$$\Delta < 0 \qquad \Delta := \Delta + c_y$$

$$y := y + 1;$$
$$\Delta := \Delta + c_x;$$

$$x > x_2$$

Human-Computer Interaction



Human

Mouse

Keyboard

Face Tracking

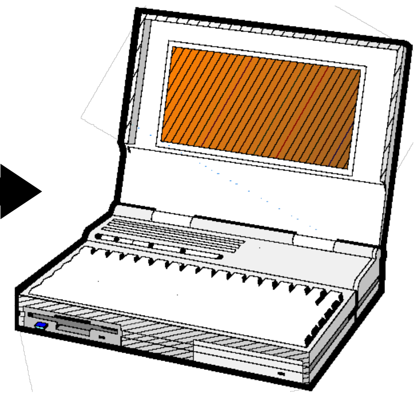
Speech Recognition

Handwriting Recognition

Gesture Recognition

Eye Tracking

Lipreading



Computer

Multi-modal Interface

Karlsruher Institut für Technologie

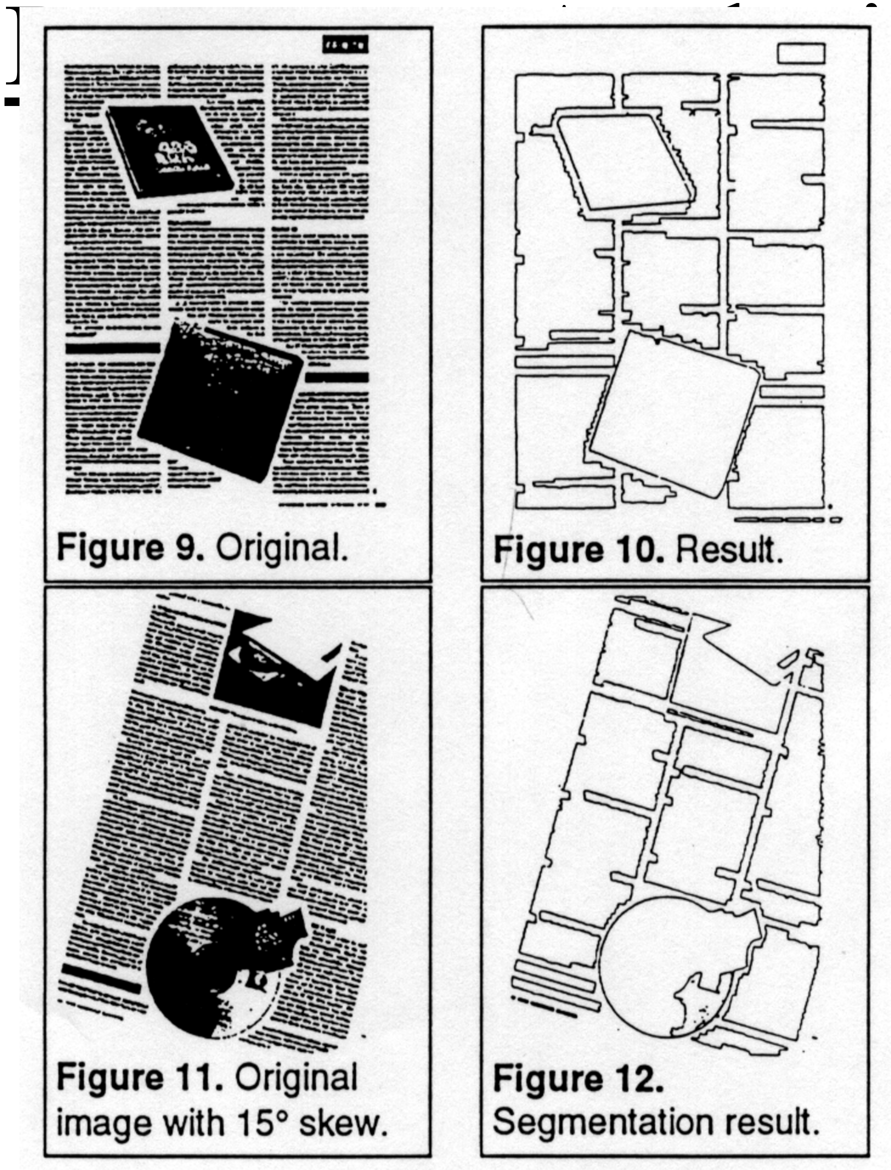


Figure 9. Original.

Figure 10. Result.

Figure 11. Original image with 15° skew.

Figure 12. Segmentation result.

Handwriting Recognition

Part II

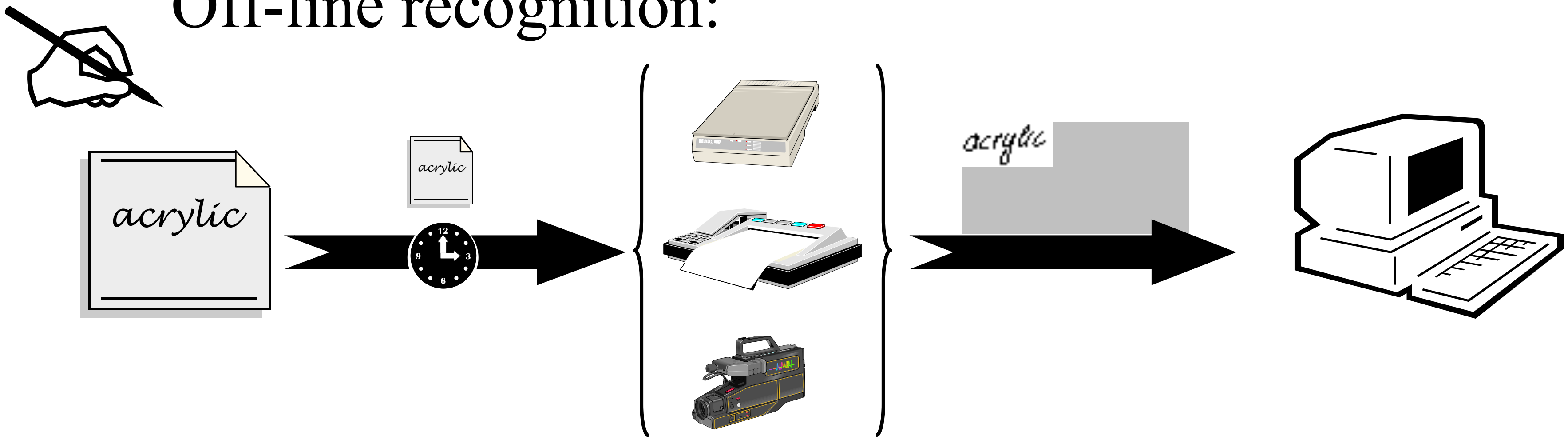
Algorithms and Systems

Alex Waibel
Carnegie Mellon University
University of Karlsruhe

Karlsruher Institut für Technologie

Off-line and On-line Handwriting Recognition

Off-line recognition:



On-line recognition:

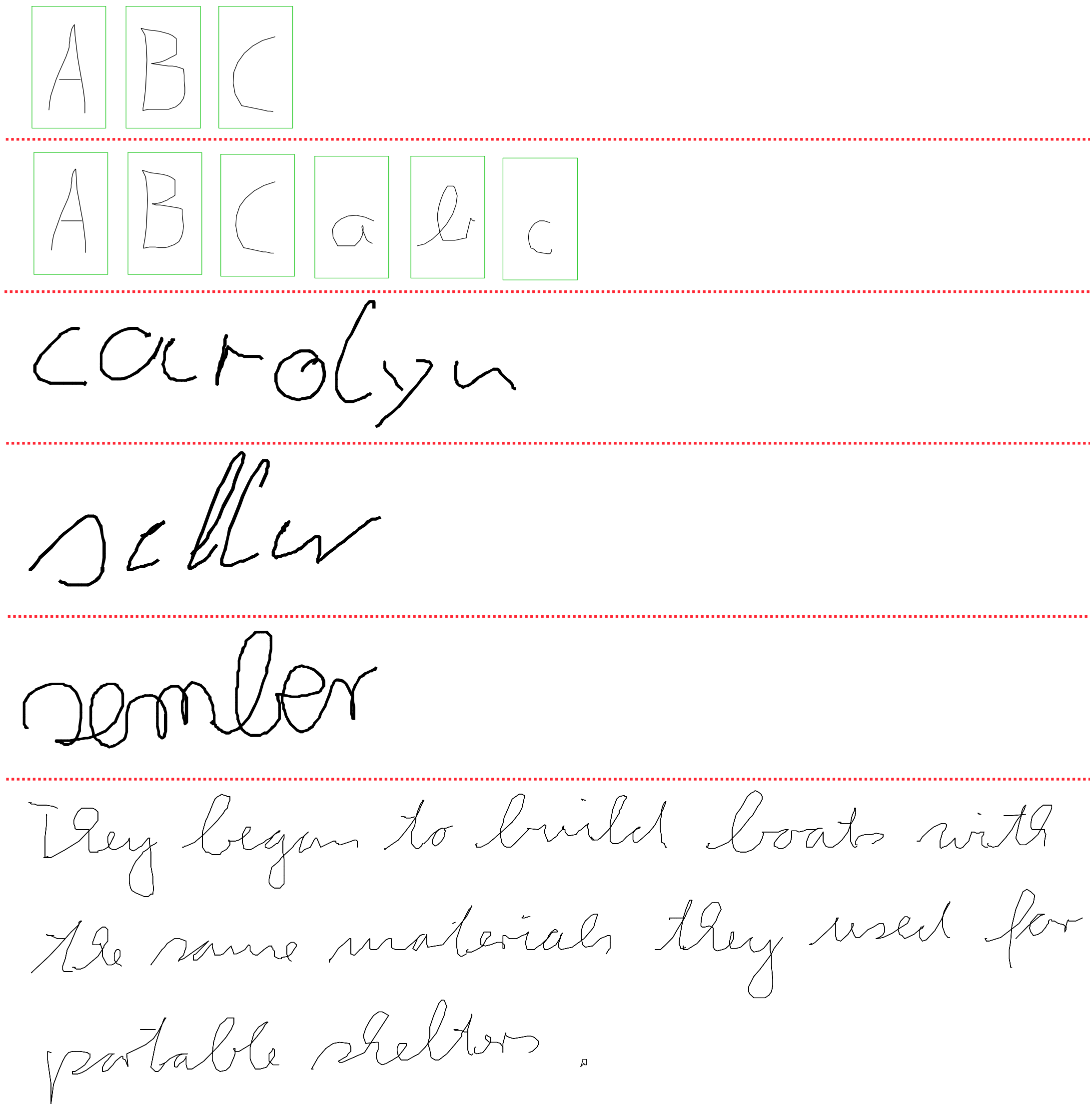
On-line Handwriting Recognition

- Provides pen-based input for human-computer interaction
- Pen-based input can be used
  - alone or
  - as part of a multi-modal interface
- Possible applications include
  - form filling
  - editing existing text
  - short notes
  - calendars
- Input consists of dynamic writing information (e.g. temporal sequence of data points)
- But: stroke order (within characters or words) influences recognition

Evaluating Handwriting Recognition Systems

- Comparing different handwriting recognizers is difficult
- Performance depends on
  - recognition task (e.g. isolated characters, words, unconstrained text)
  - writing style (e.g. printed, mixed, cursive)
  - size of dictionary
  - intended end user(s)
    - single writer (allows writer dependent system)
    - multi-writer
    - omni-writer (requires writer independent system)

Handwriting Recognition Tasks



boxed upper case letters

boxed letters

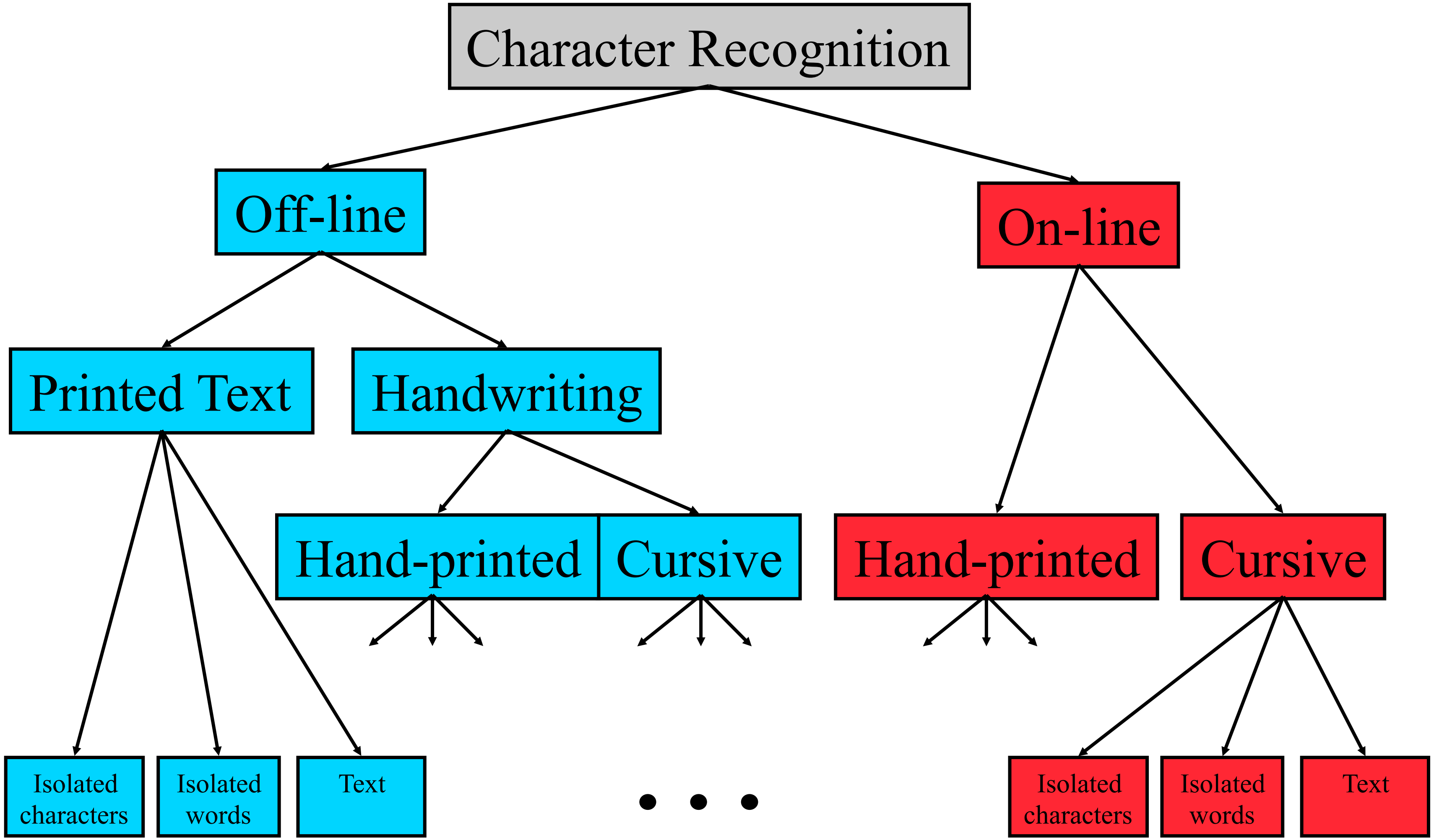
printed words

mixed words

cursive words

unconstrained text

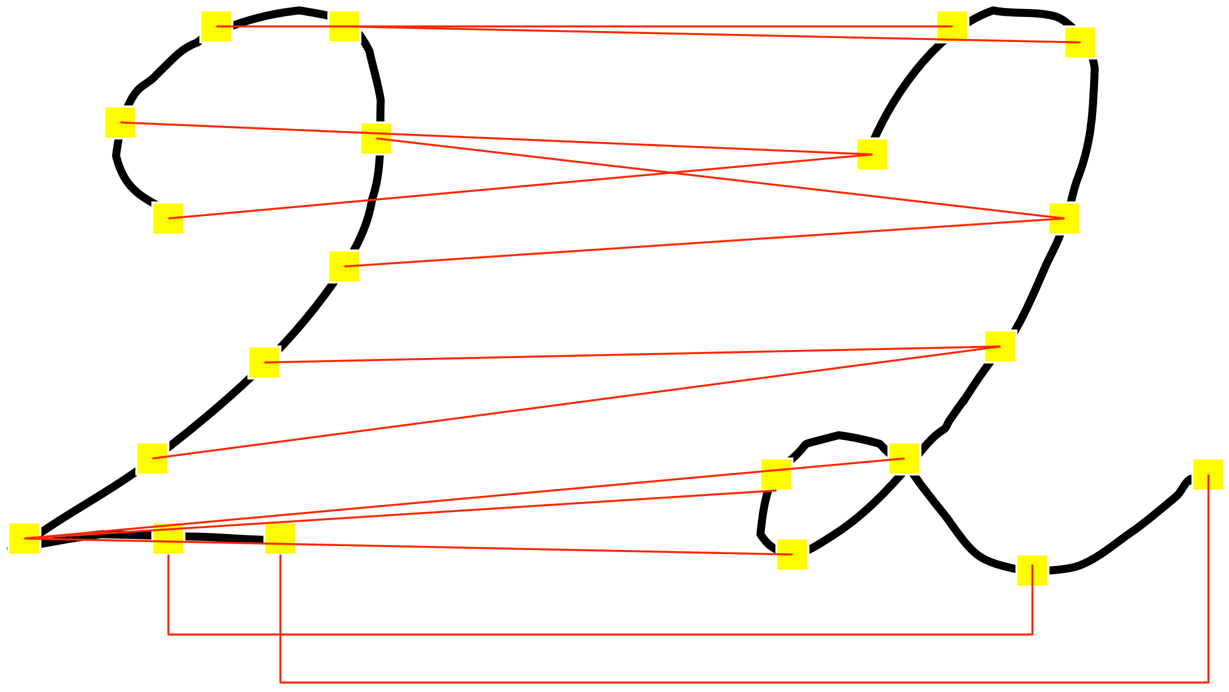
Character Recognition



On-Line Recognition

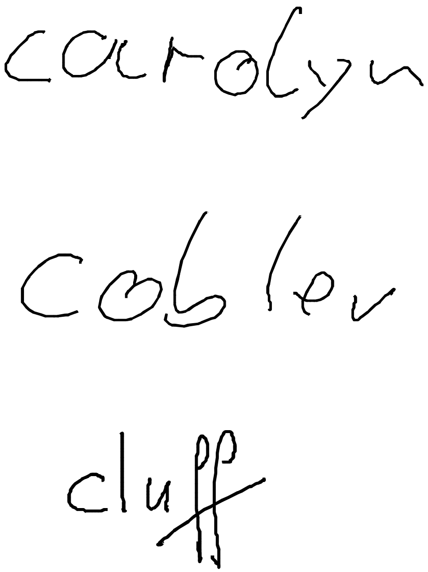
- Simplest Case:
  - Single Characters (English, German)
  - Upper Case
  - Position Known
  - On at a time

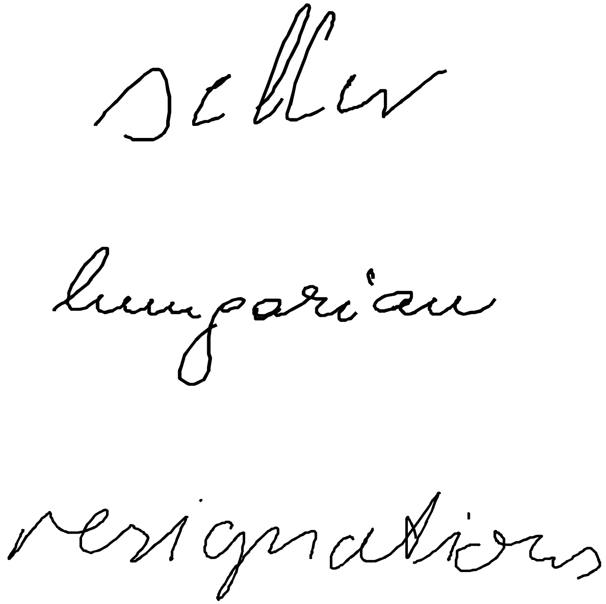
Elastic Matching

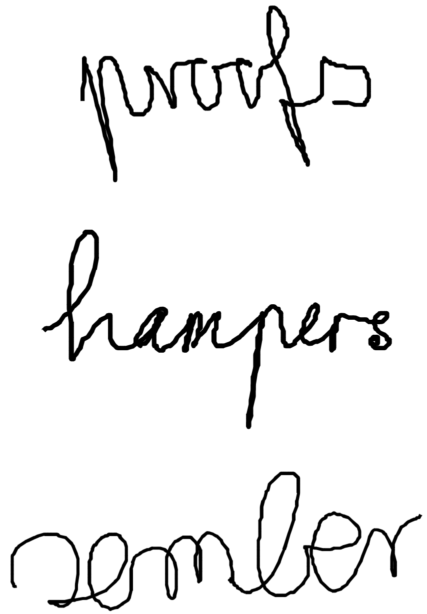


Writing Styles

Printed	Mixed	Cursive







The Unipen Project

- Initiated by the Technical Committee 11 of the IAPR to
  - constitute a sizable, quality database
  - evaluate the state of the art in on-line handwriting recognition by testing recognizers in the same conditions on several tasks of various difficulty
  - bring together researchers and developers in on-line handwriting recognition from Universities and Industry.
- About 40 participants

  AT&T, Apple Computer Inc., Bolt Beranek and Newman Inc., IBM (NY), Lexicus Corp (CA), NICI (Netherlands), Aachen Technical University (Germany), University of Karlsruhe (Germany), Hewlett-Packard Labs (UK), Philips Research Lab (Netherlands), ...
- About 4 million characters (approx. 500K words) of data donated by participants

Recognition Strategies

- Wholistic approach
  - recognition is performed globally on the whole word
  - no attempt is made to identify characters individually

- Analytical approach
  - recognition is performed at an intermediate level
  - words are considered as sequences of smaller units (e.g. letters)

Wholistic Strategies

Wholistic methods usually follow a two-step scheme:

❶ feature extraction

❶ global recognition by comparing the representation of the unknown word with those of references stored in the dictionary

Practical consequences:

- as letter segmentation is avoided and recognition performed in a global way it is tolerant to deformations that affect unconstrained cursive handwriting
- the recognition is constrained to a specific dictionary of words
- if training on word samples is required, the dictionary cannot be updated automatically from letter information and thus a training step is mandatory to expand or modify the dictionary

Suitable for applications

- with small dictionaries
- where the dictionary is statically defined and not likely to change

Analytical Strategies

- Analytical strategies deal with several levels of representation
  - Feature level
  - One or more intermediate levels dealing with subparts of words
  - Integration at word level
- Different kinds of subparts: letters, graphemes, states, strokes ...
- Units of intermediate level usually are related to letters
- Letter-based recognition is independent from specific dictionary
- Dictionary can be replaced or modified without any retraining
- Analytical approaches fall into two main categories:

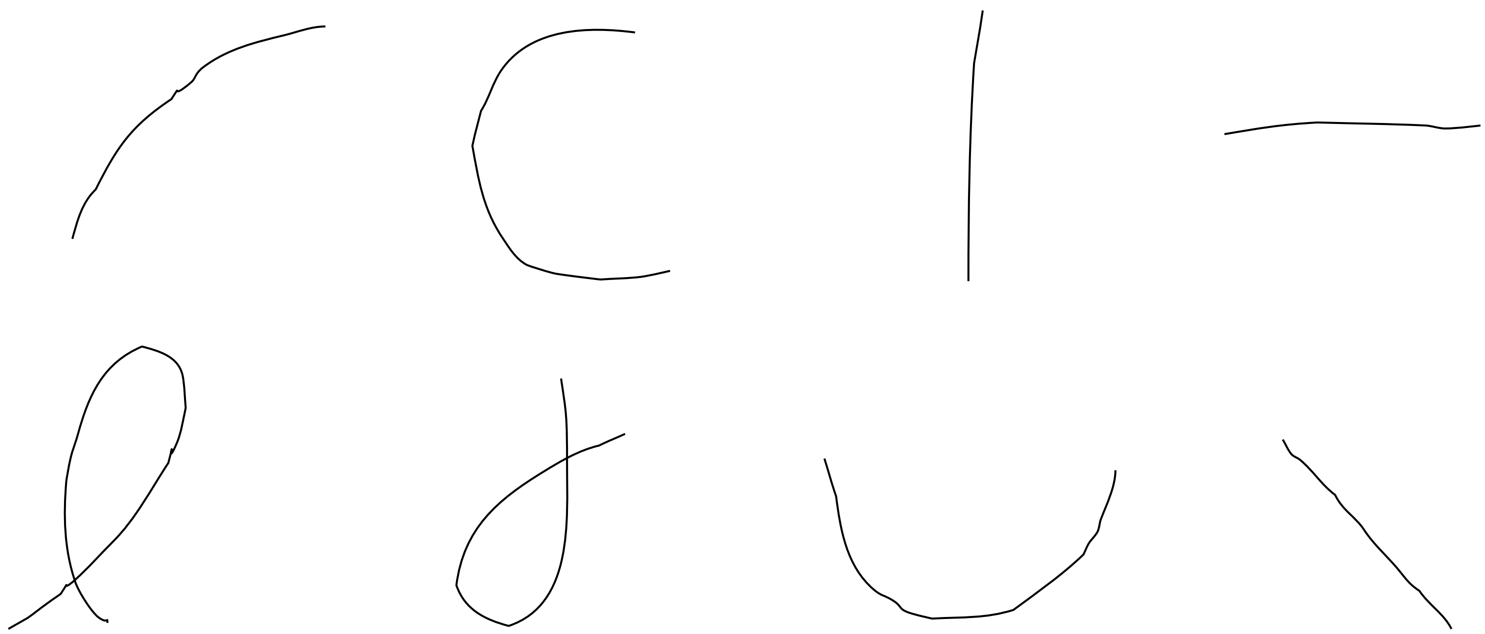
  ❶ with explicit segmentation (input segmentation)

  ❷ with implicit segmentation (output segmentation)

  which differ in the way (letter) segmentation is performed

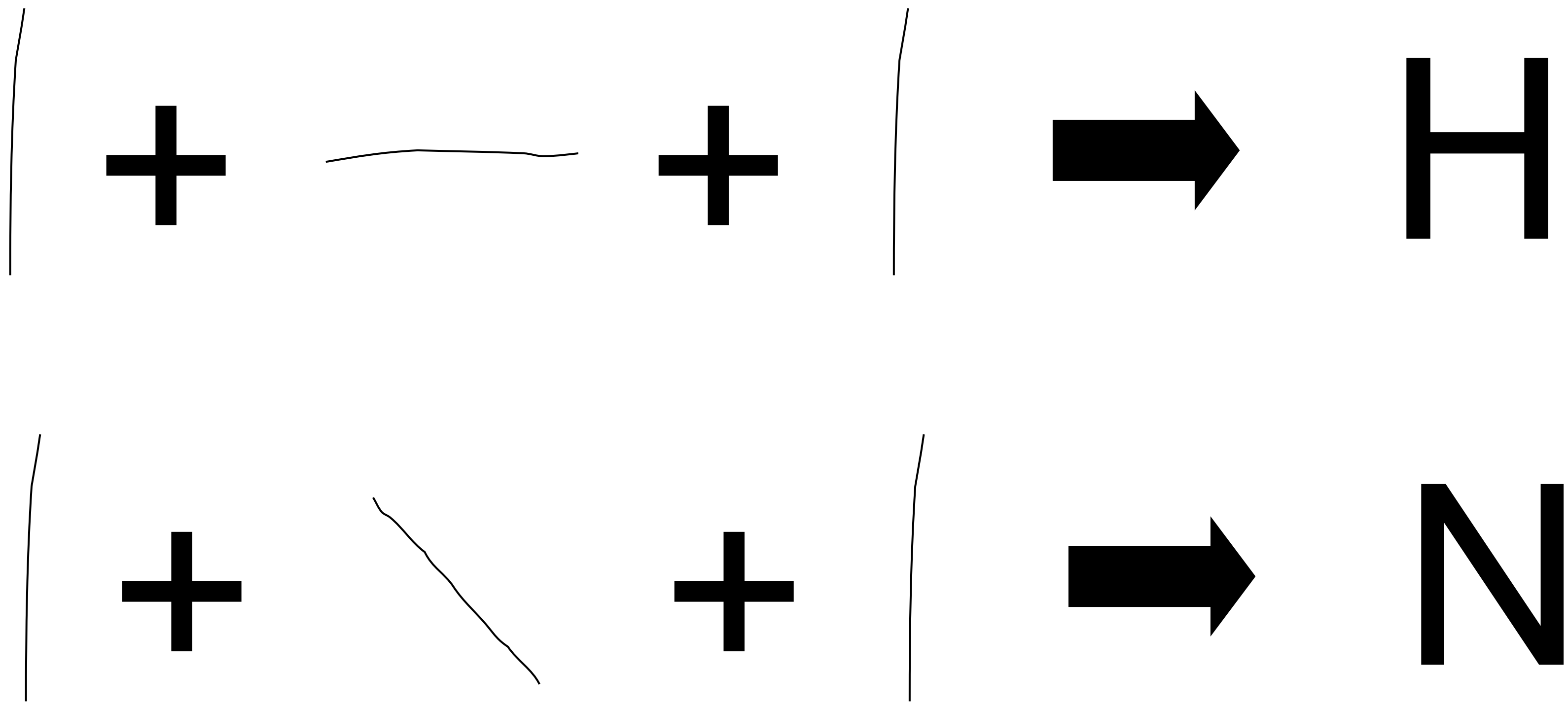
Stroke Level

Decomposition / Identification of atomic units, building blocks



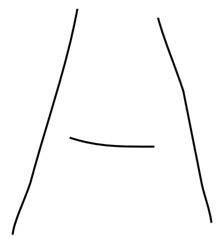
Strokes Level

Finding Rules

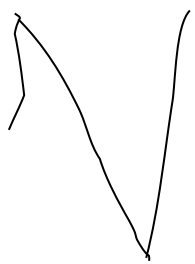


Stroke Level

Disadvantage: complex rule bases

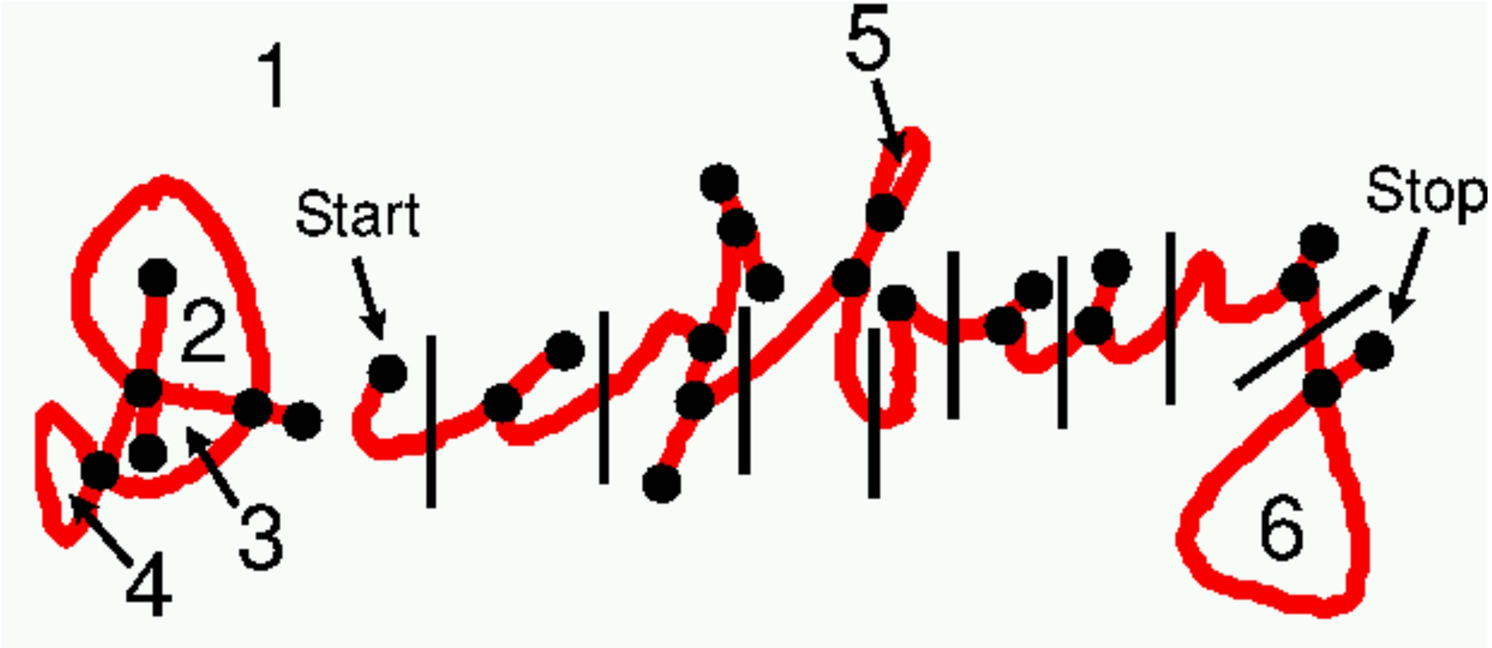


A oder H ?

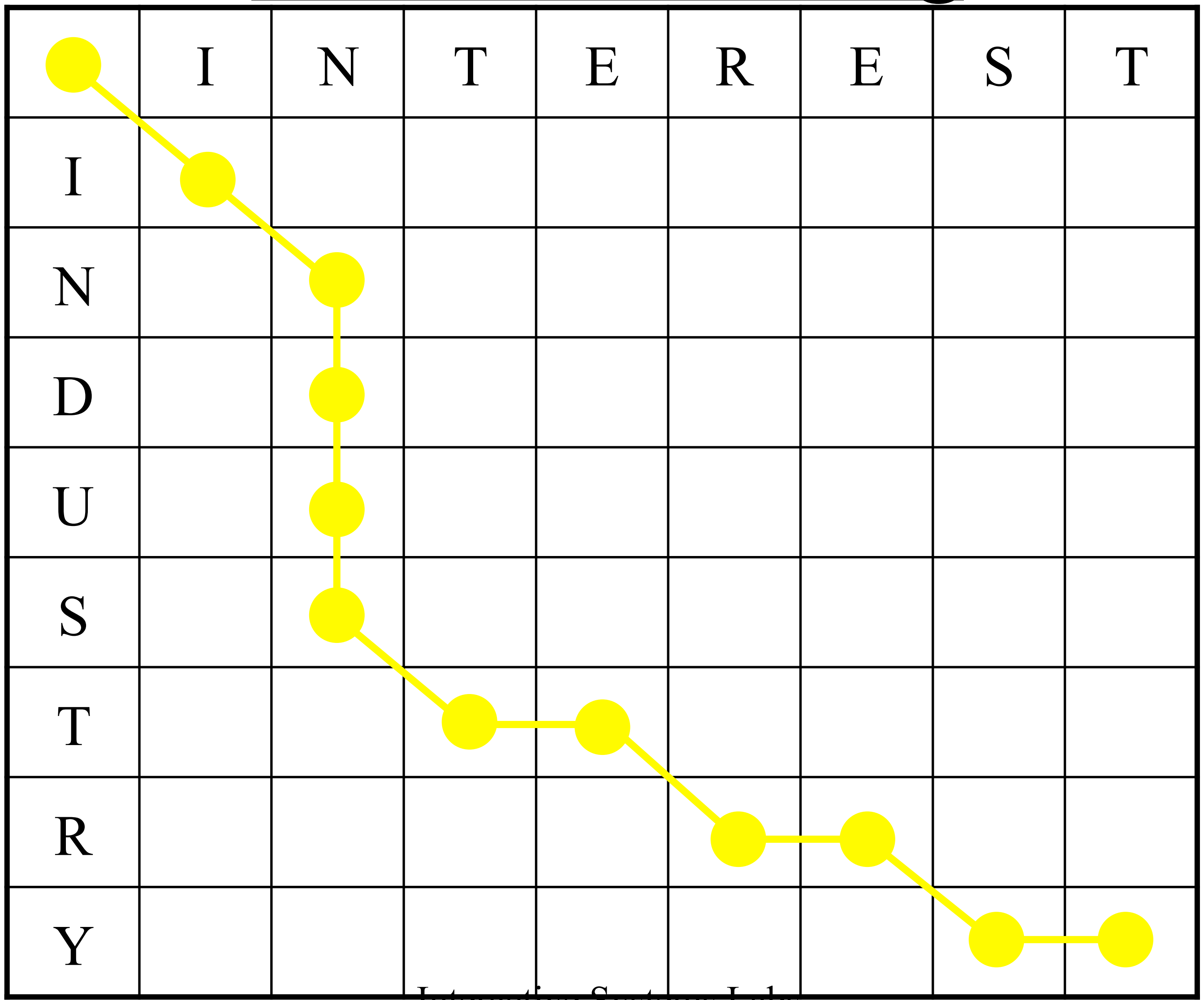


N oder V ?

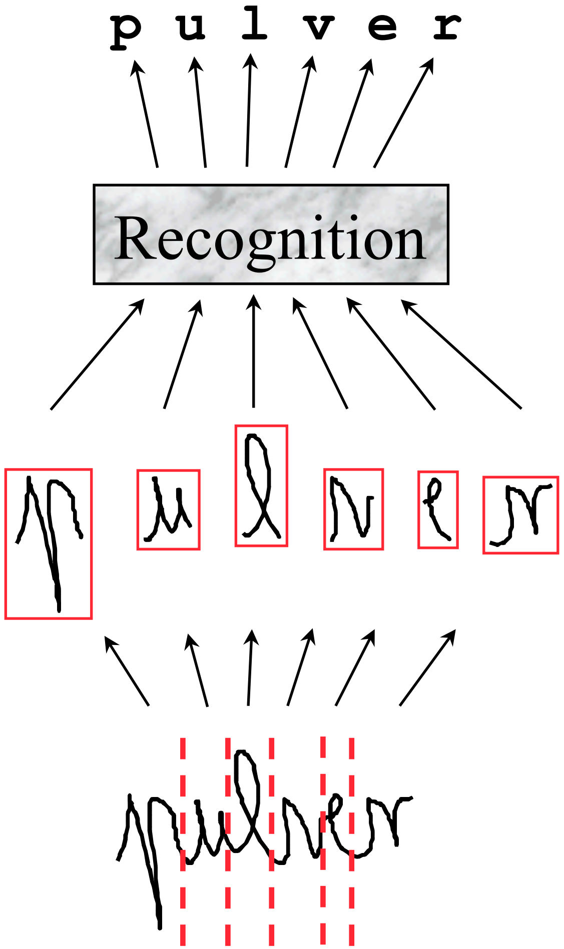
Explicit Segmentation



Elastic Matching

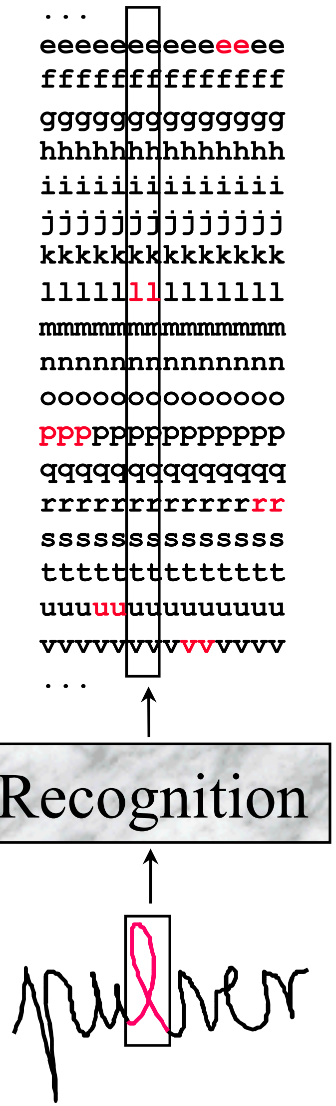


Input and Output Segmentation



Input Segmentation

Output Segmentation



Input Segmentation

- Following three steps are performed:
  ❶ external segmentation of the word into smaller units (e.g. letters)
  ❷ individual recognition of these units
  ❸ contextual post-processing using lexical, syntactic or semantic knowledge
- Major drawbacks of input segmentation:
  - segment boundaries are often difficult or impossible to find
  - erroneous segmentation may lead to incorrect recognition
- Contextual post-processing can be performed by
  - orthographic correction techniques using statistics of the dictionary (e.g. based on n-gram frequencies)
  - direct comparison with a dictionary (e.g. based on Edit Distance)
- Classical remark:

  „*it is necessary to segment to recognize, but it is also necessary to recognize to segment*“

Output Segmentation

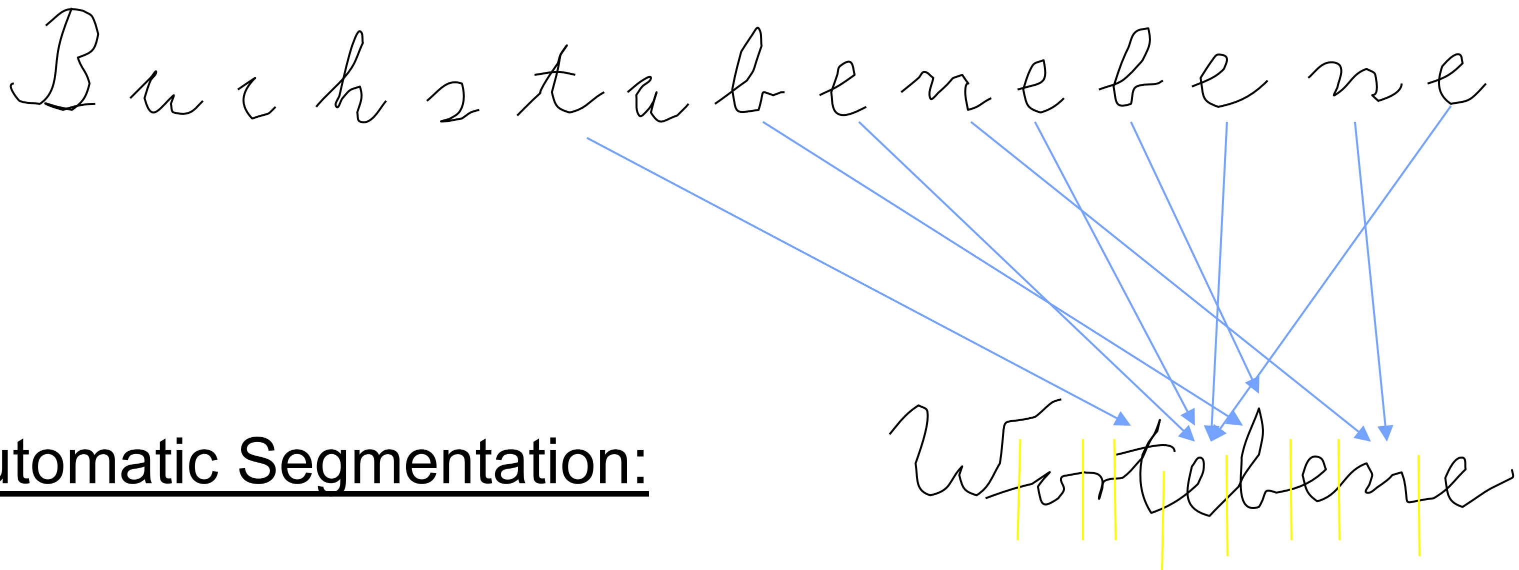
- A recognition-based segmentation is performed

- The decision about letter boundaries is delayed to the end of the recognition process
  - this avoids the problem of misrecognitions through early segmentation errors

- Contextual knowledge can be introduced
  - in a statistical way (e.g. letter n-gram)
  - by using dictionaries

Training

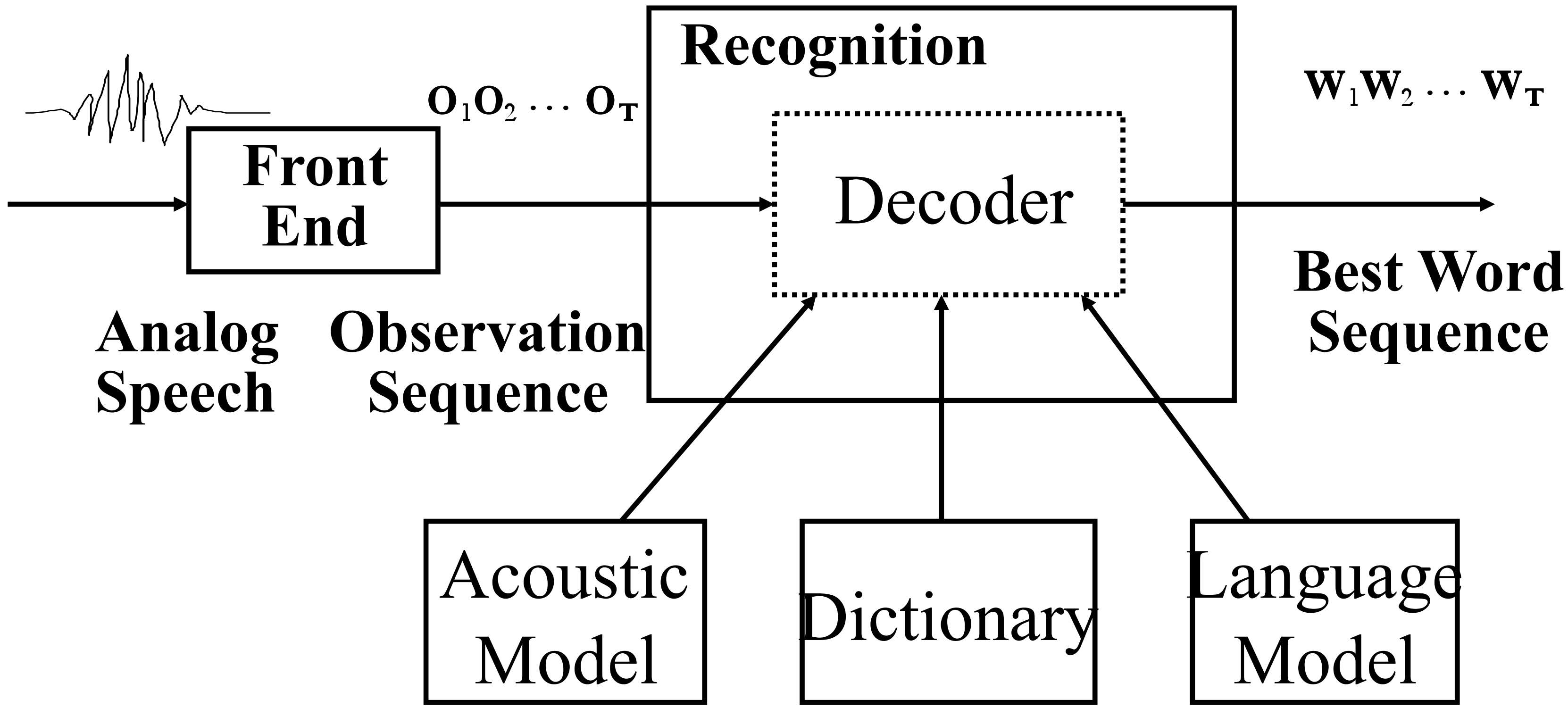
Manual Segmentation:



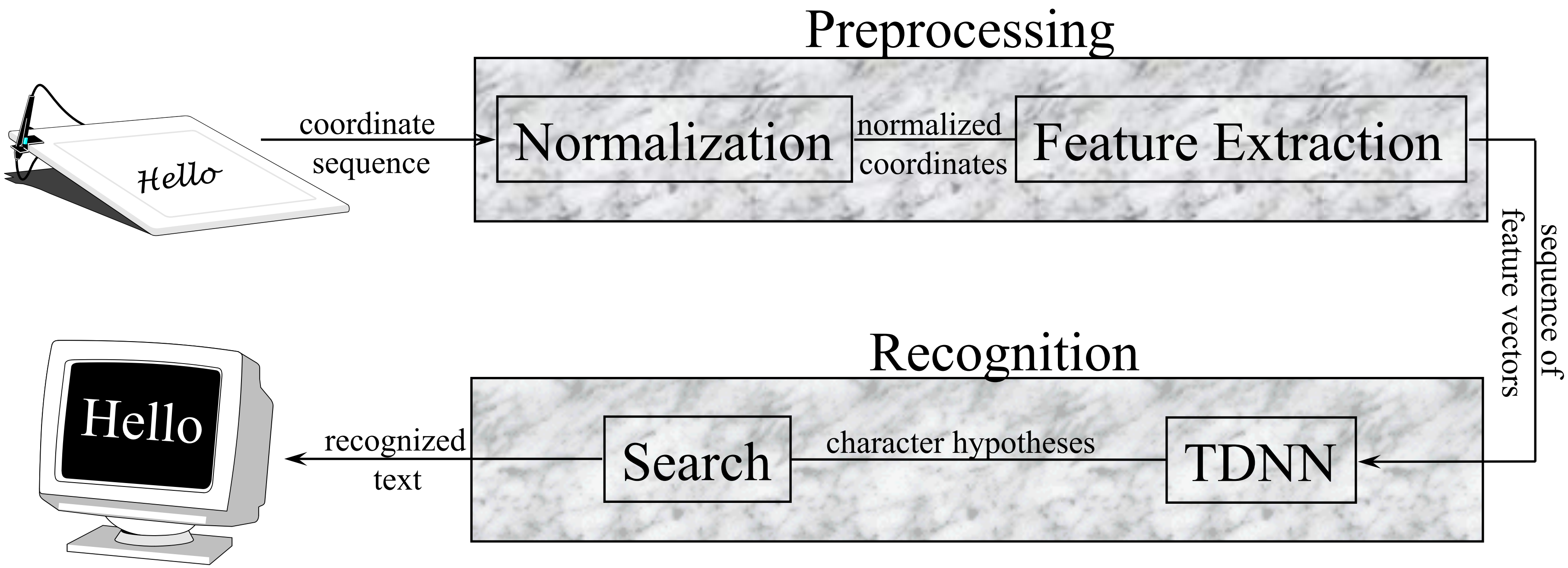
Automatic Segmentation:

Speech Recognition

- Recognizer Components:



NPen^{++} - Cursive Handwriting Recognition



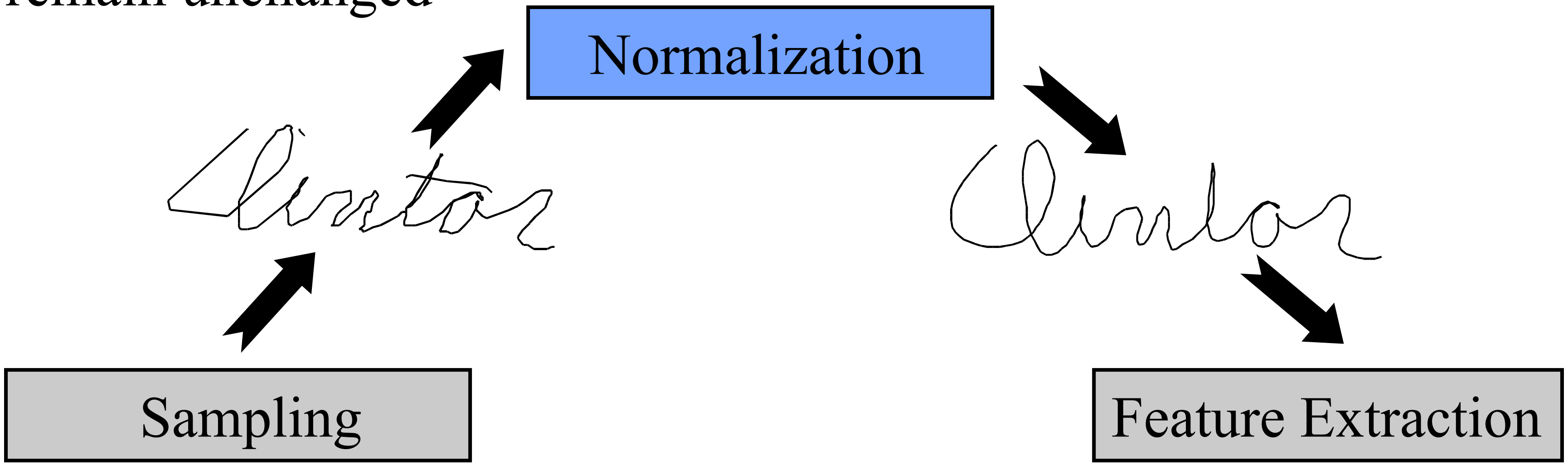
The system was designed ...

- ... to be writer independent
- ... to work with large vocabularies
- ... to be fast enough for real-world applications
- ... to make use of the dynamic writing information

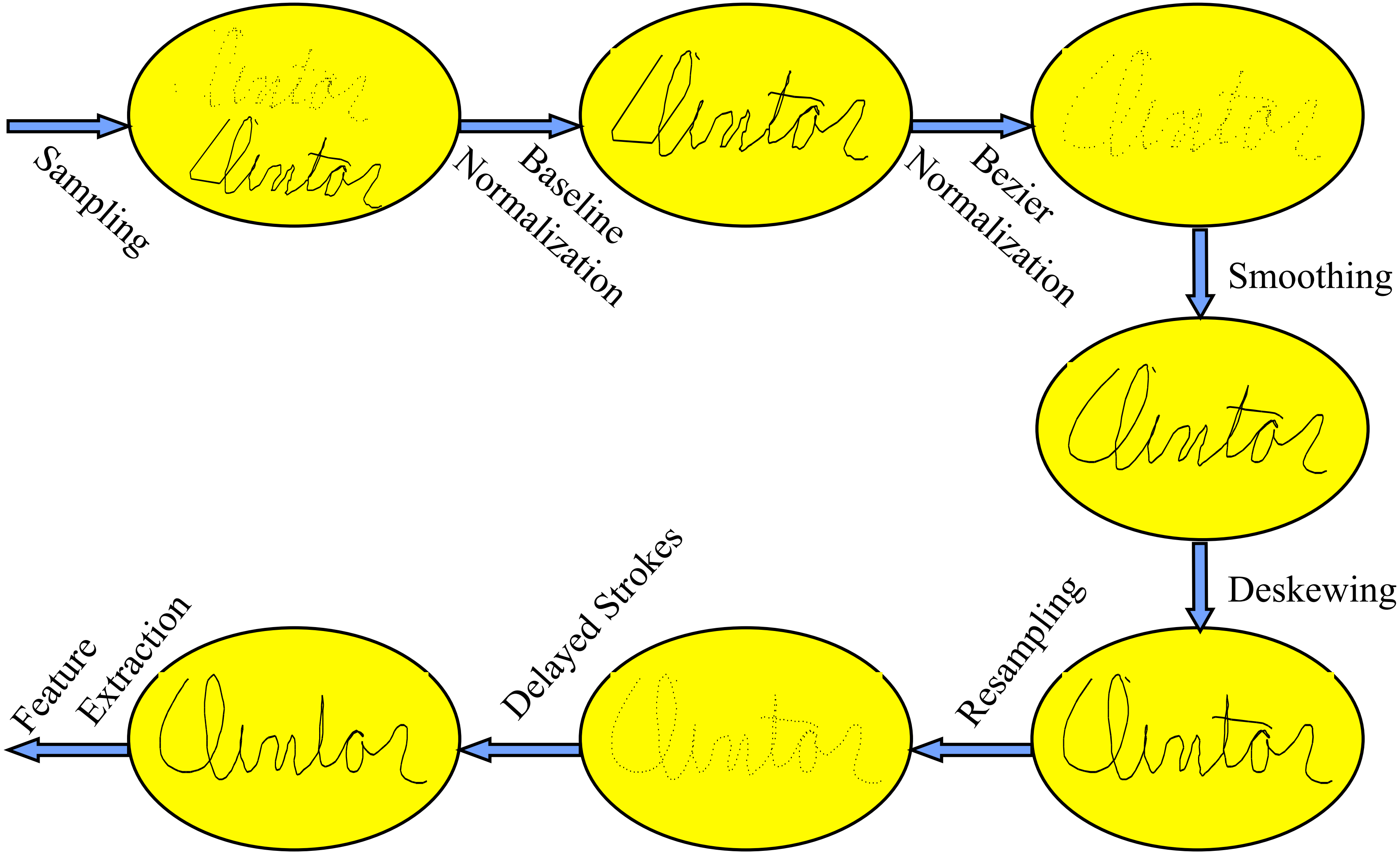
Karlsruher Institut für Technologie

Normalization

- Removing undesired variability from the original pen trajectory
  - baseline normalization, deskewing
  - bezier normalization, smoothing
  - size normalization
  - resampling from temporal to spatial equidistance
  - removing delayed strokes
- The original dimension and temporal ordering of the input signal remain unchanged

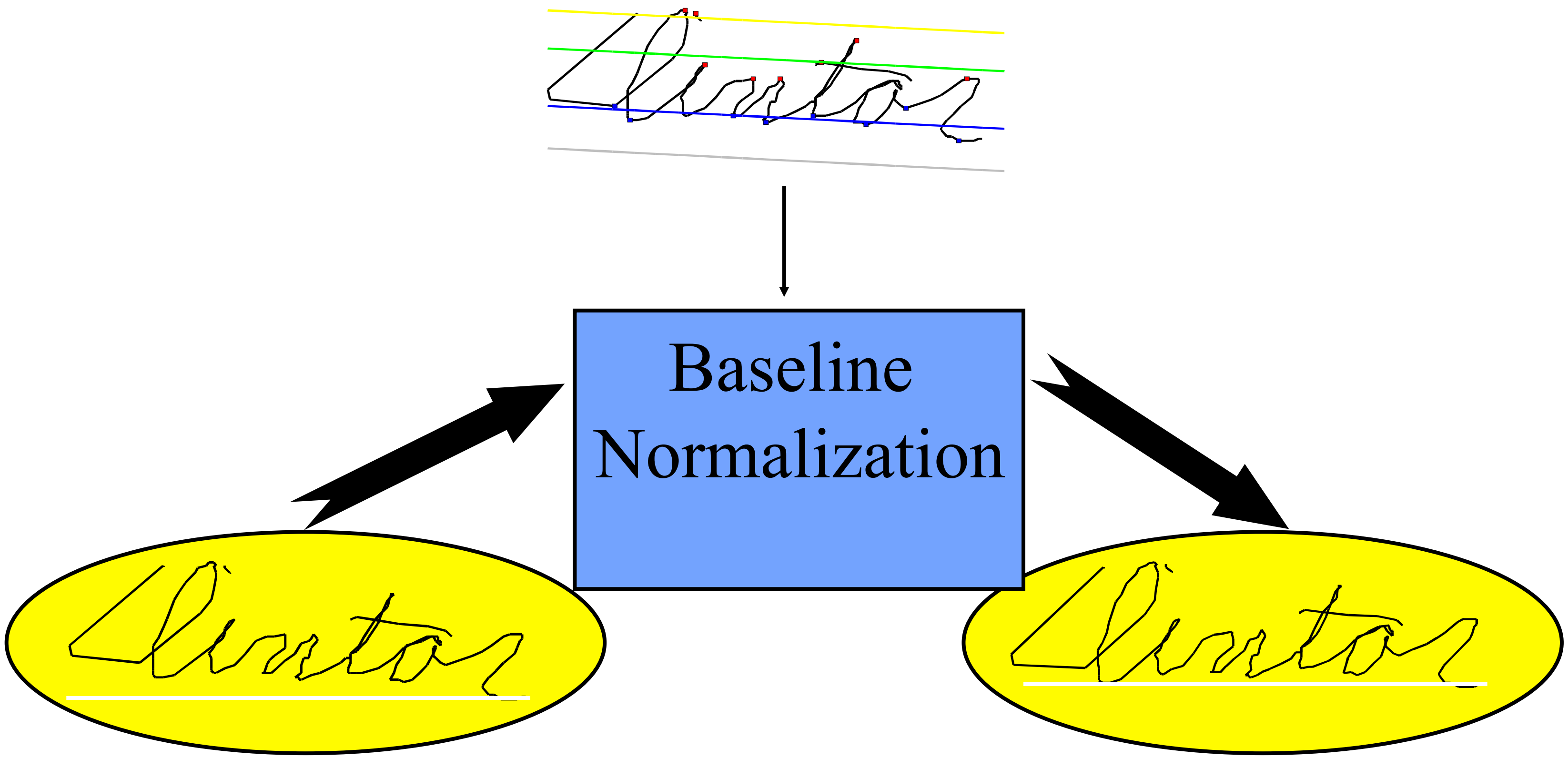


Normalization Overview



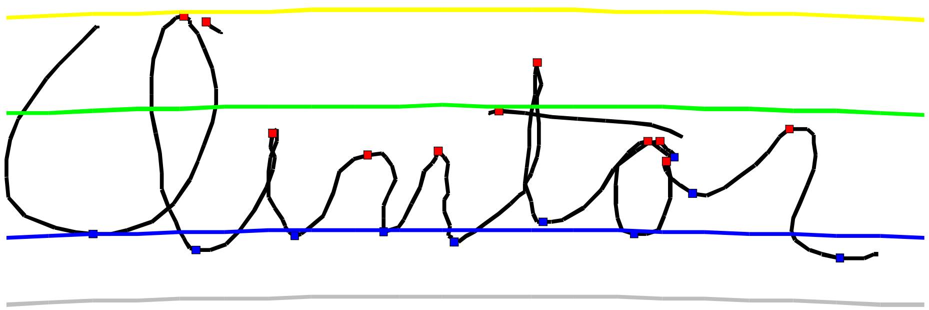
Baseline Normalization

- According to the computed baselines the input pattern is rotated to a nearly horizontal orientation:



Baseline Detection

- Using an EM (Expectation Maximization) algorithm the baseline, centerline, descenderline and ascenderline of the pattern are calculated simultaneously from the local minima and maxima:



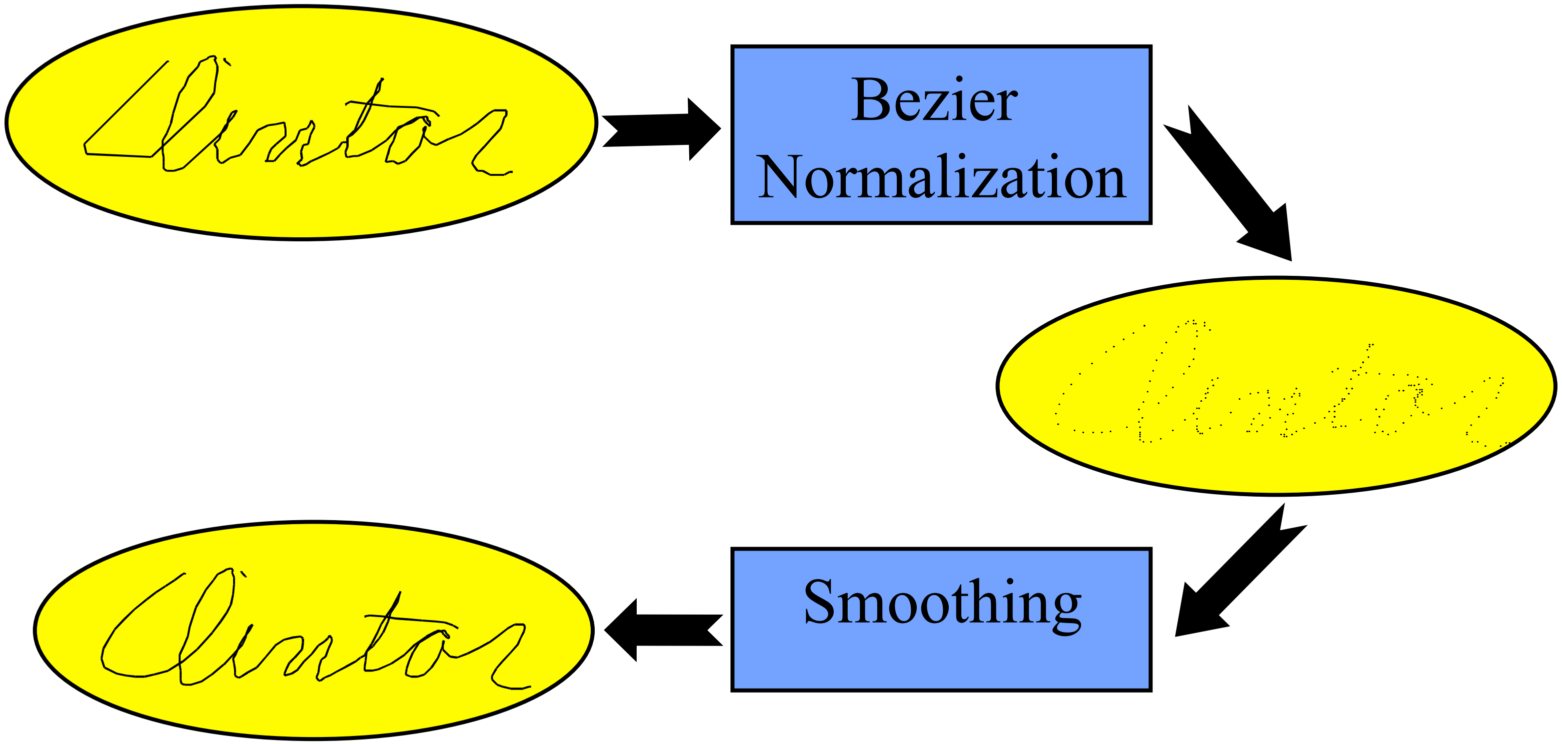
- The second degree polynomials

$$f_i(x) = k(x - \tilde{x})^2 + s(x - \tilde{x}) + y_i \qquad (i = 0,\ldots,3)$$

are used to approximate these lines, where the parameters k (curvature), s (slant) and x (horizontal displacement) are shared among all four curves. The vertical displacements y_i are given by $y_0 = b - d, y_1 = b, y_2 = b + c, y_3 = b + c + a$

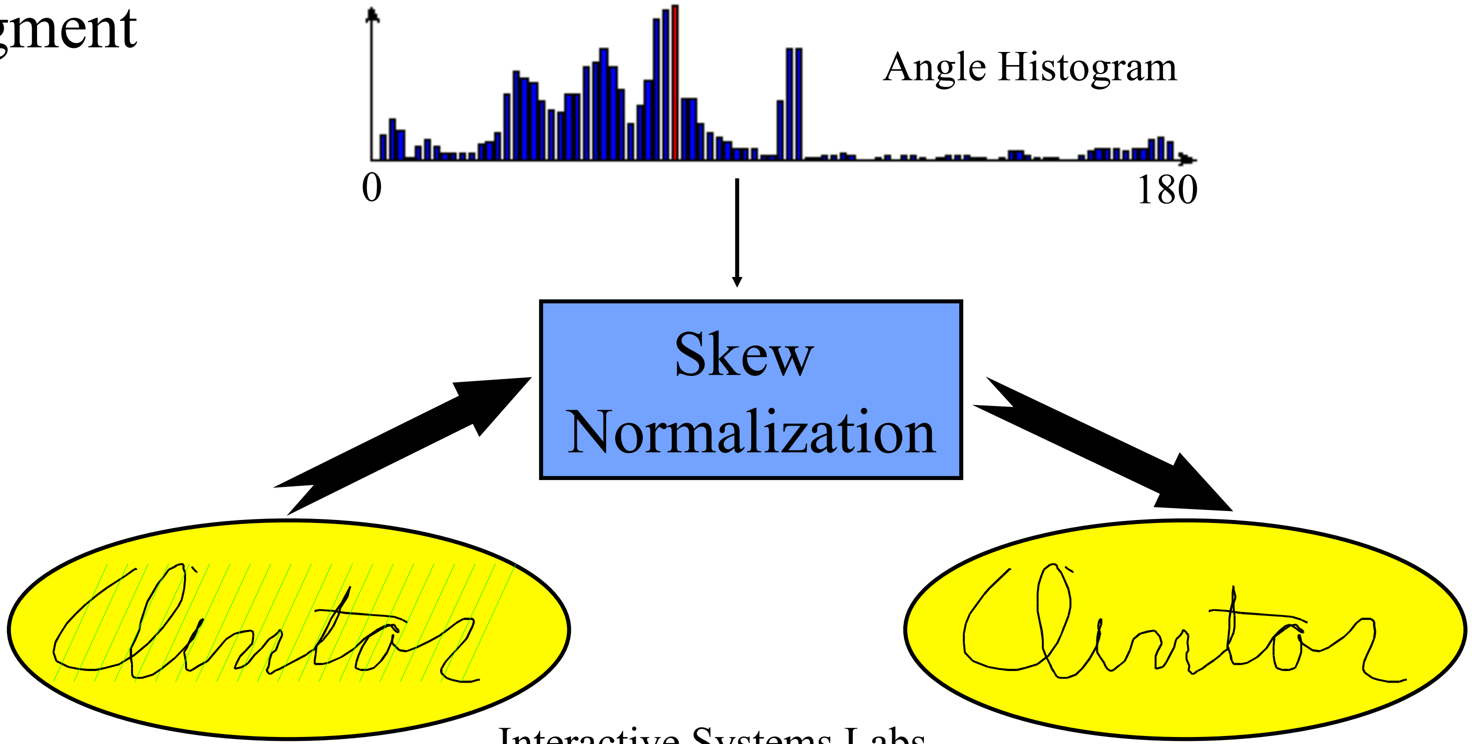
Bezier Normalization and Smoothing

- A Bezier algorithm, which approximates missing data points, is used to compensate for sampling errors
- A moving average window is used for smoothing to remove sampling noise



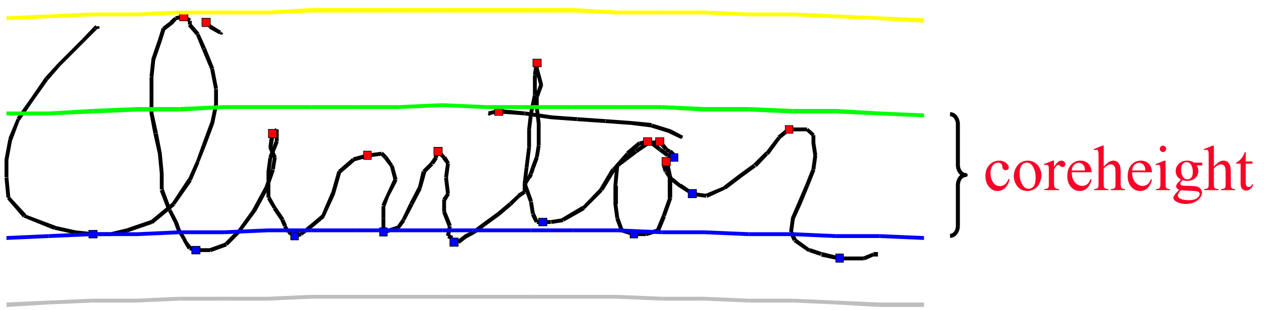
Skew Normalization

- To ensure a nearly vertical orientation of all characters, the input pattern is normalized according to the skew angle
- The skew is computed from a histogram of all angles between a line segment and the x-axis multiplied with the length of this segment



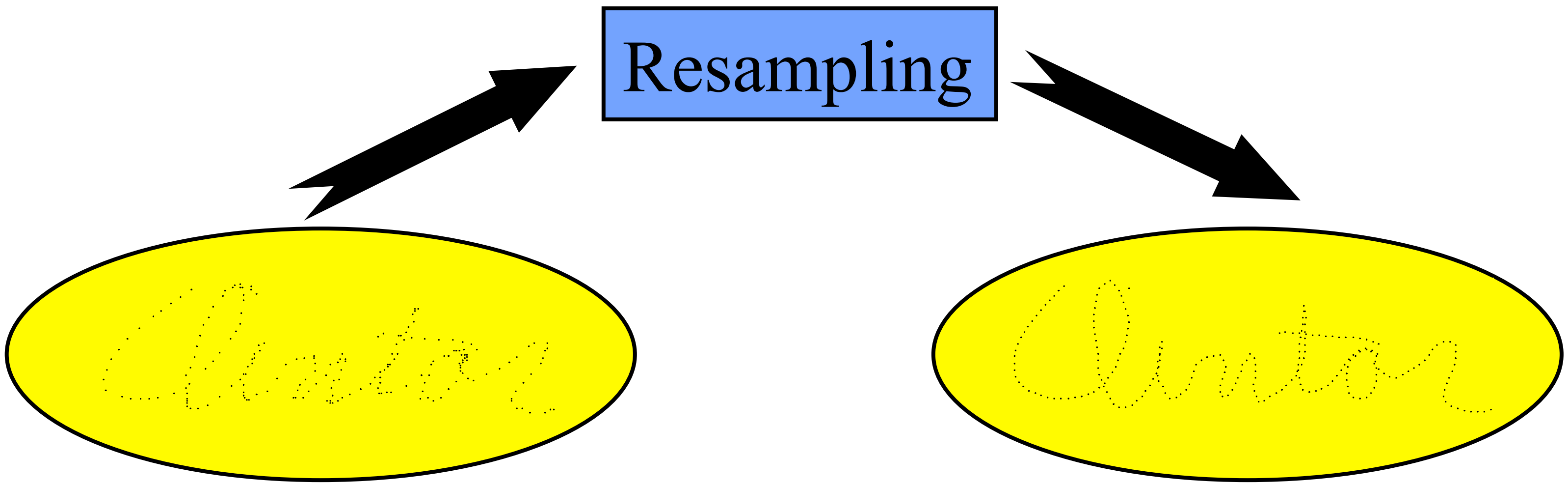
Size Normalization

- The pattern is rescaled with respect to its current coreheight, which is the distance between the baseline and centerline
- This ensures that all words have (nearly) the same character size



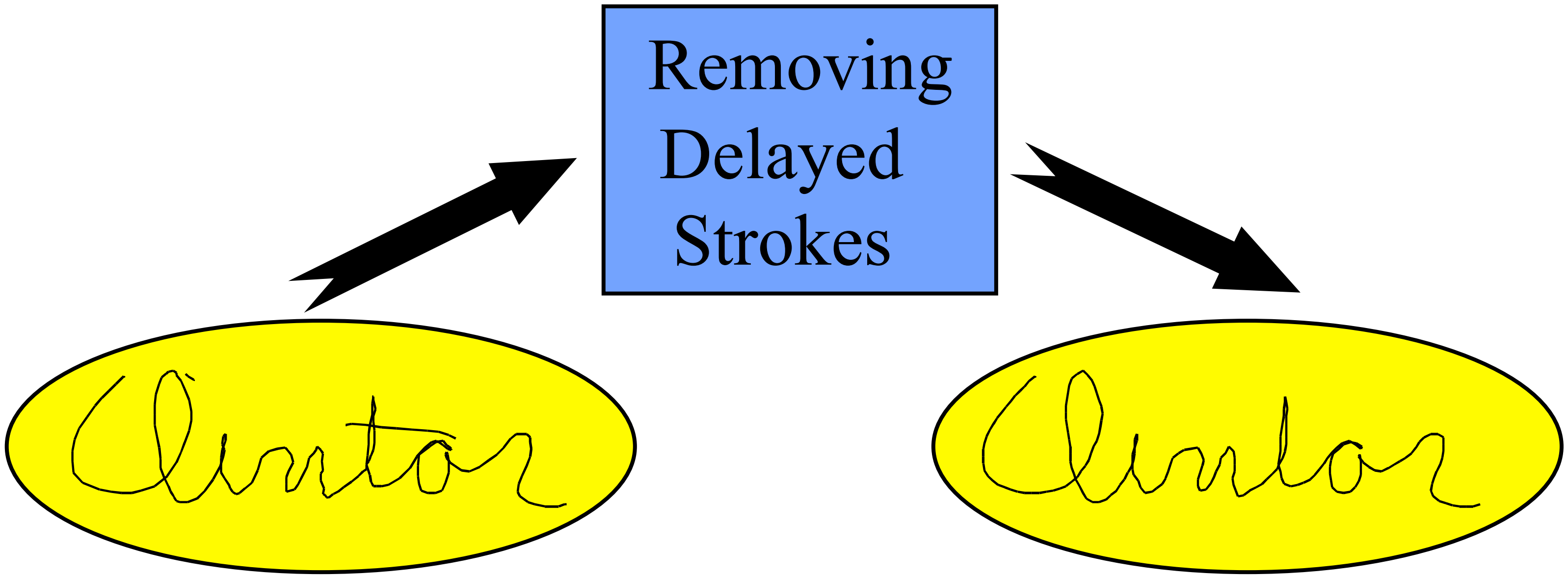
Resampling

- The spatial distance between two successive data points depends on
  - the general sampling rate of the used hardware
  - sampling errors
  - the current writing speed
- Therefore the sequence of data points is resampled from temporal to spatial equidistance

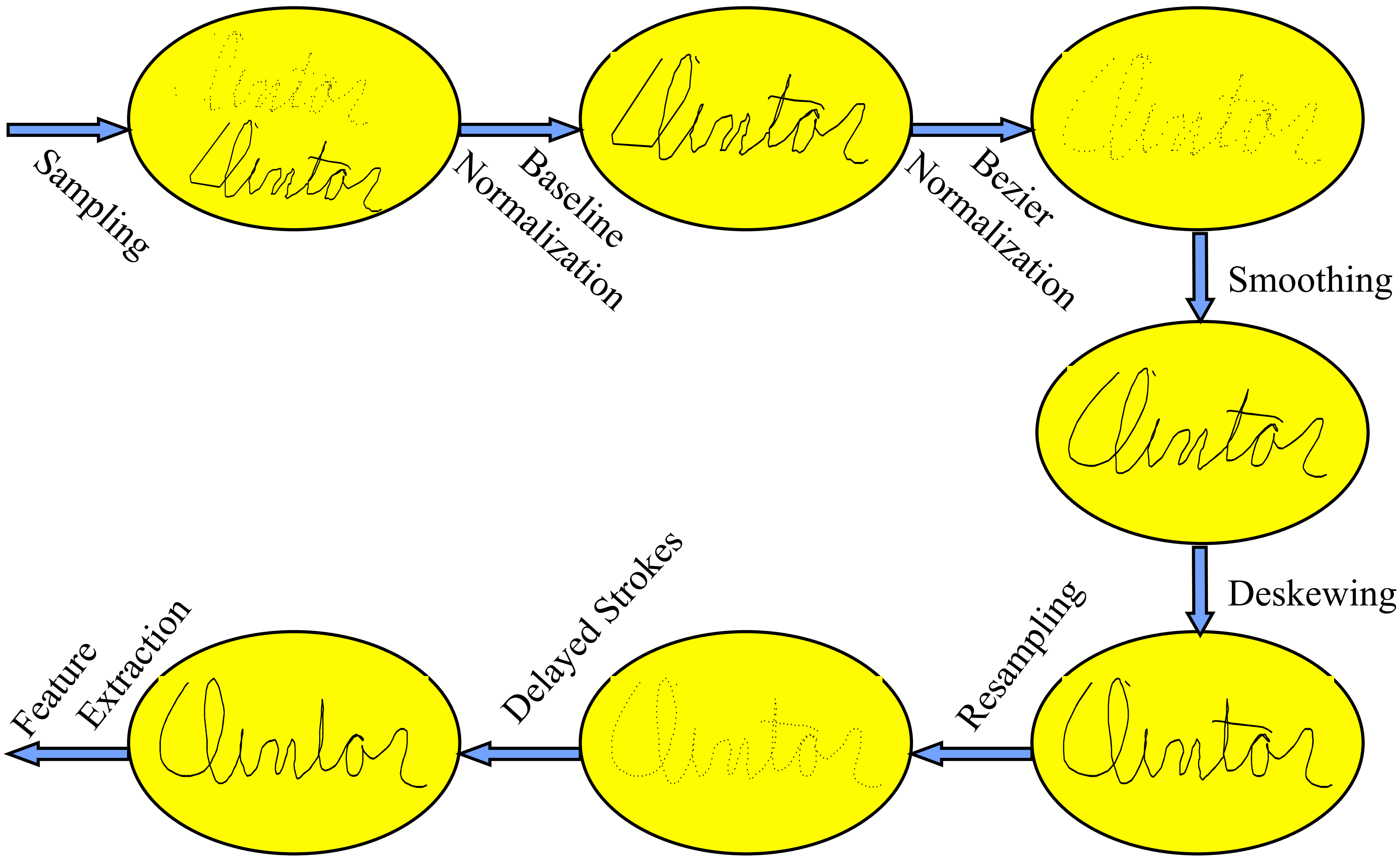


Removing Delayed Strokes

- Delayed strokes like i-dots and t-strokes are removed from the sequence of data points, if they do not occur directly after the corresponding character in the temporal sequence

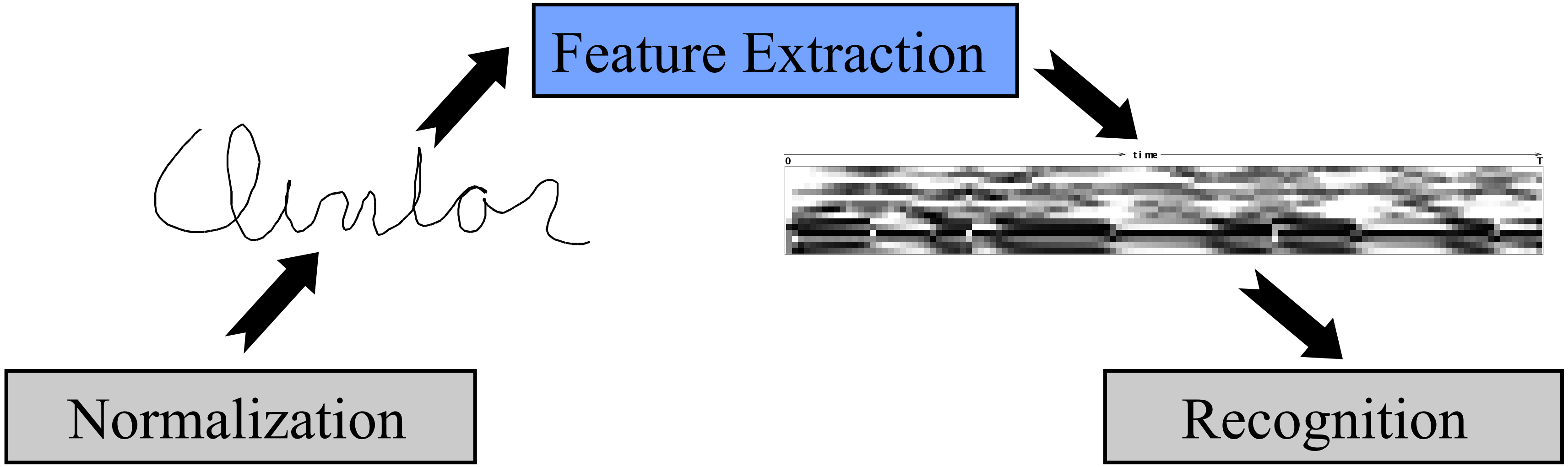


Normalization Overview

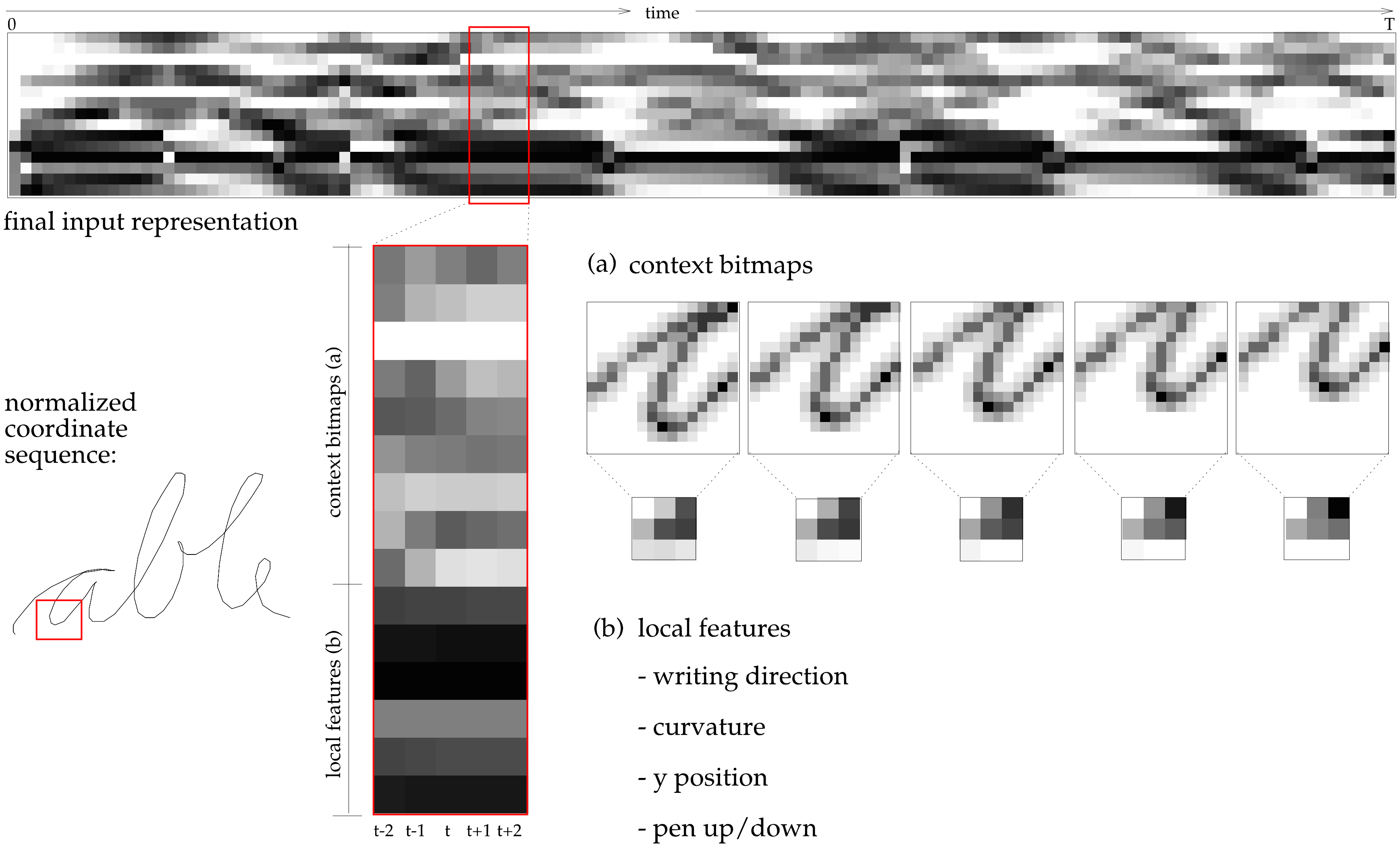


Feature Extraction

- Extraction of features along the normalized pen trajectory, yielding a temporal sequence of n-dimensional feature vectors
- Each feature vector consists of:
  - Local features: writing direction, curvature, position, pen up/down, lineness, aspect, curliness, slope, ...
  - Global features: context bitmaps



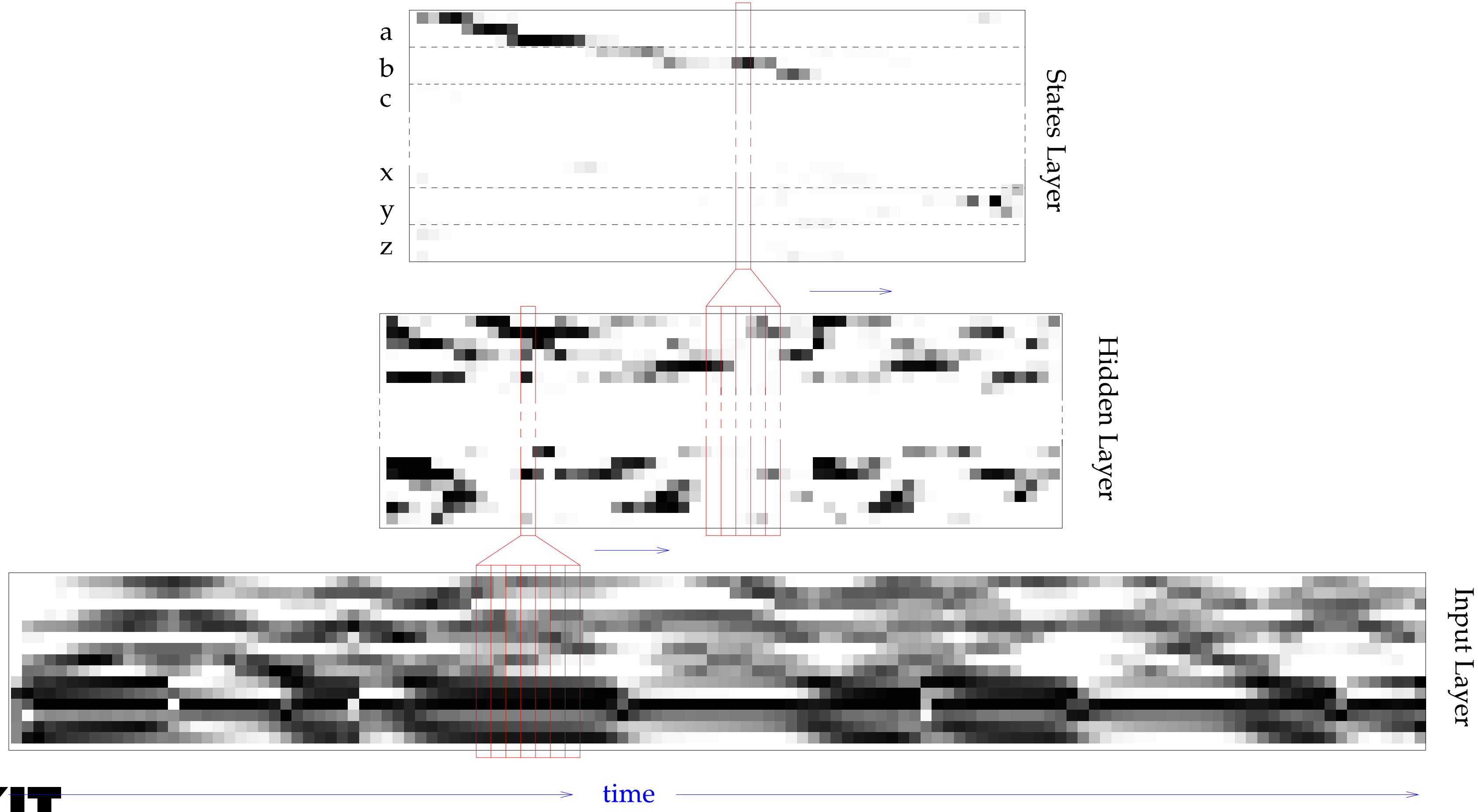
Feature Extraction



Karlsruher Institut für Technologie

Computing State Hypotheses

Time Delay Neural Network (TDNN) to compute state hypotheses over time given a feature vector sequence:



Word Modeling

- Each word w_i of the dictionary $W = \{w_1,...w_k\}$ is represented as its character sequence

$$w_i = c_{i1}c_{i2}...c_{ik}$$

- Each character c_j itself is modelled by a three state hidden markov model

$$c_j = q_j^0\ q_j^1\ q_j^2$$

- The states q_j^0, q_j^1, q_j^2 model the initial, middle and final section of the character's coordinate sequence
- I.e. the final modelling of word w_i is

$$w_i = q_{i0}\ q_{i1}\ ...\ q_{j3k}$$

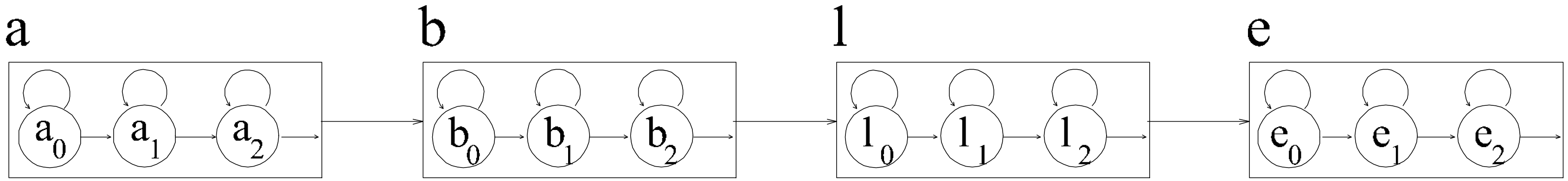
(e.g. able = $a_0a_1a_2b_0b_1b_2l_0l_1l_2e_0e_1e_2$)

Karlsruher Institut für Technologie

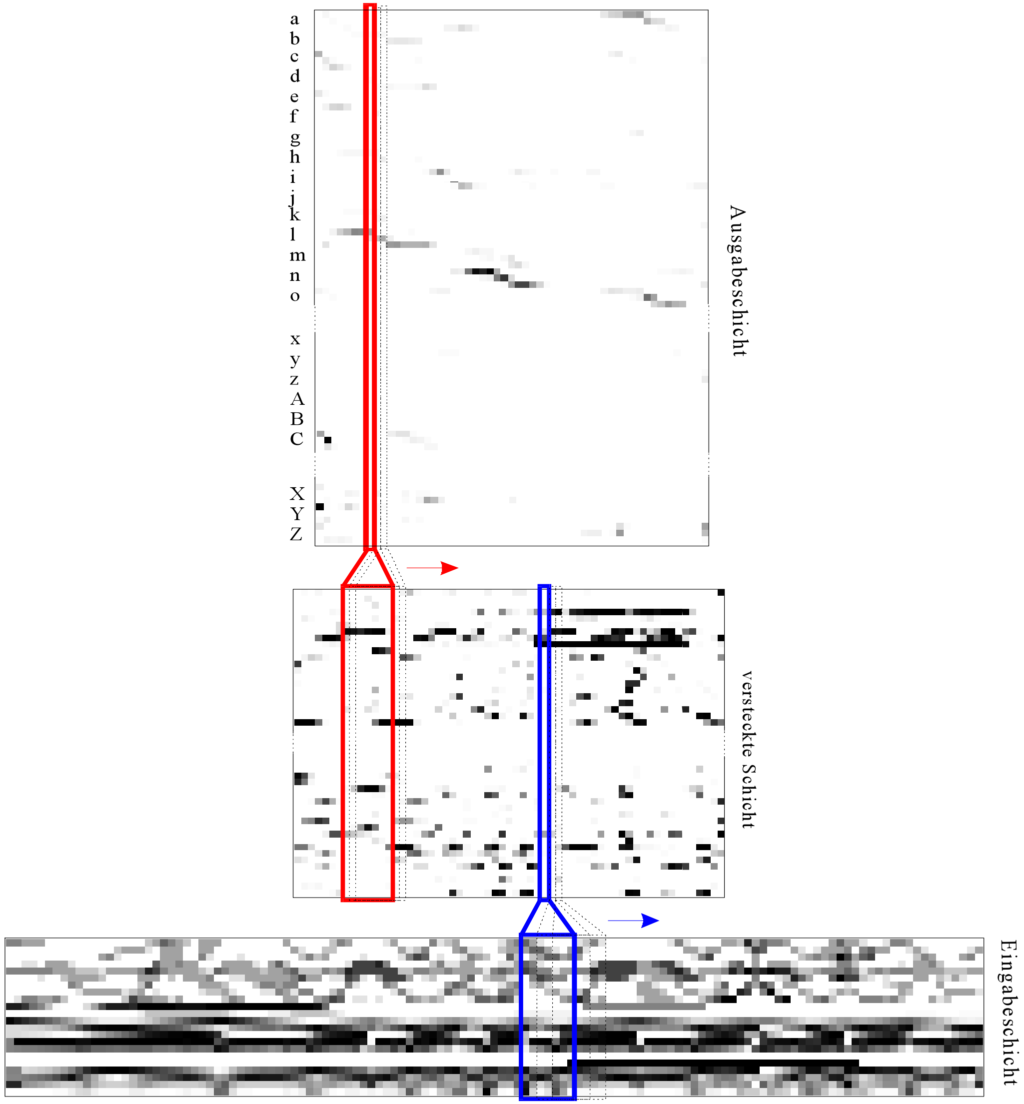
Word Modelling

- Each word of the dictionary is represented as its character sequence, where each character itself is modeled by a three state hidden markov model
- The character's states model the initial, middle and final section of the character's coordinate sequence

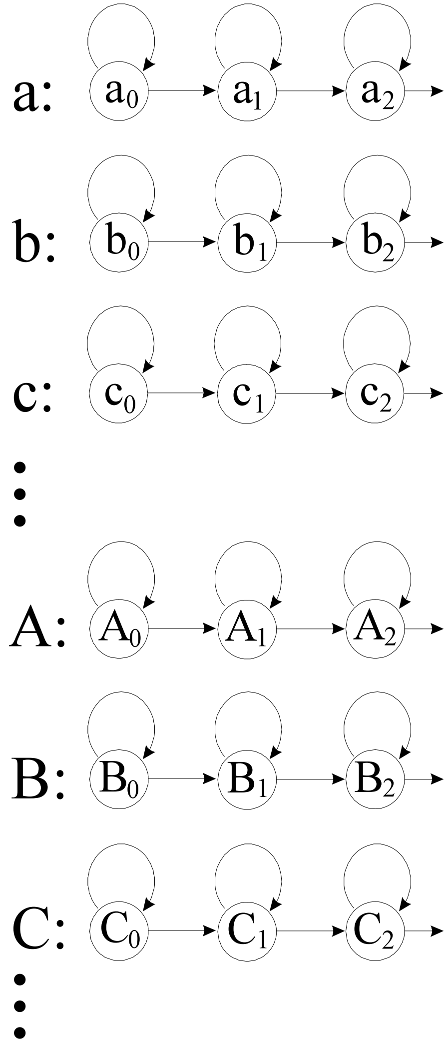
E.g. the modeling for word "able" is



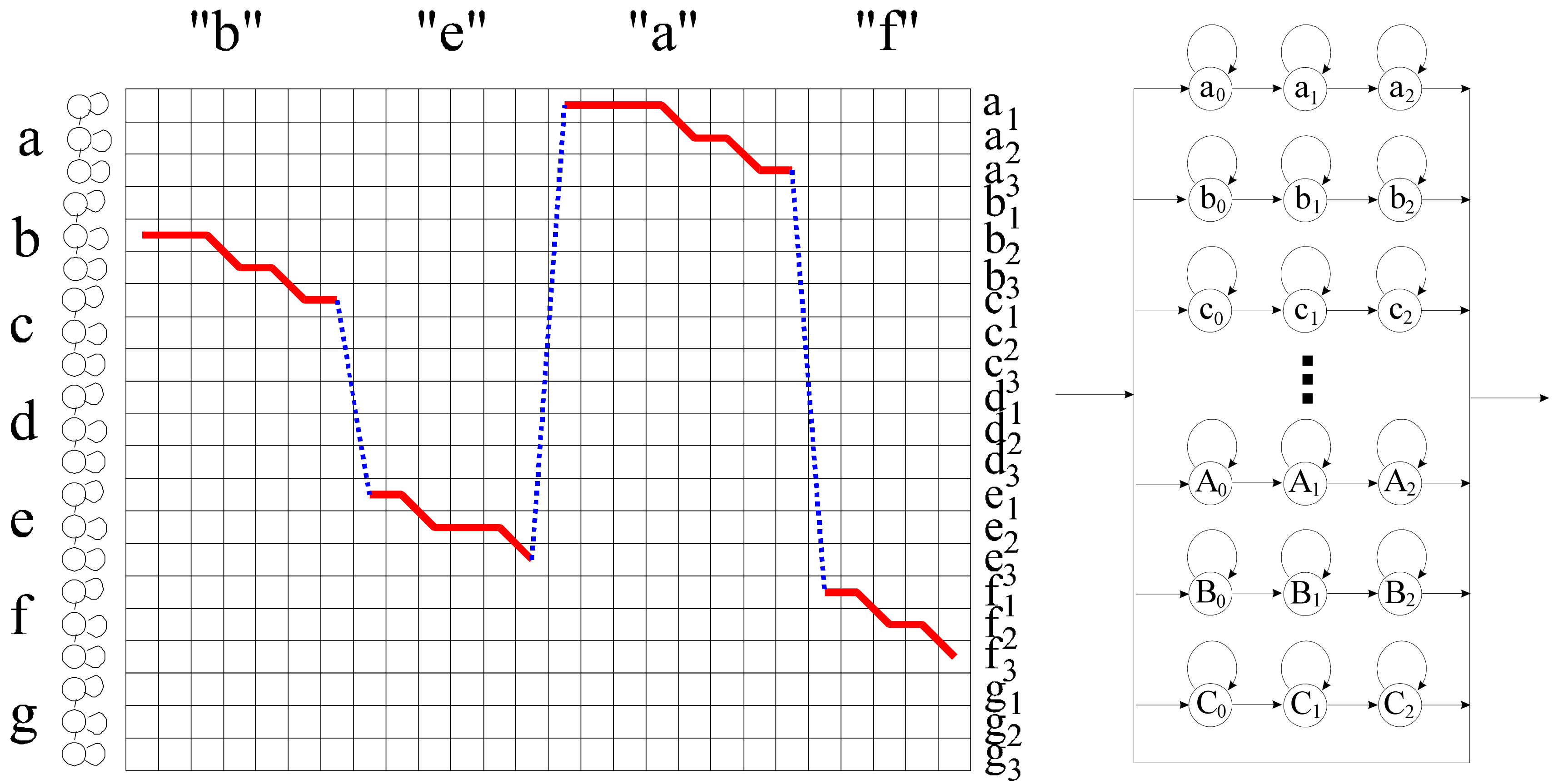
MS-TDNN Architecture



Modeling:



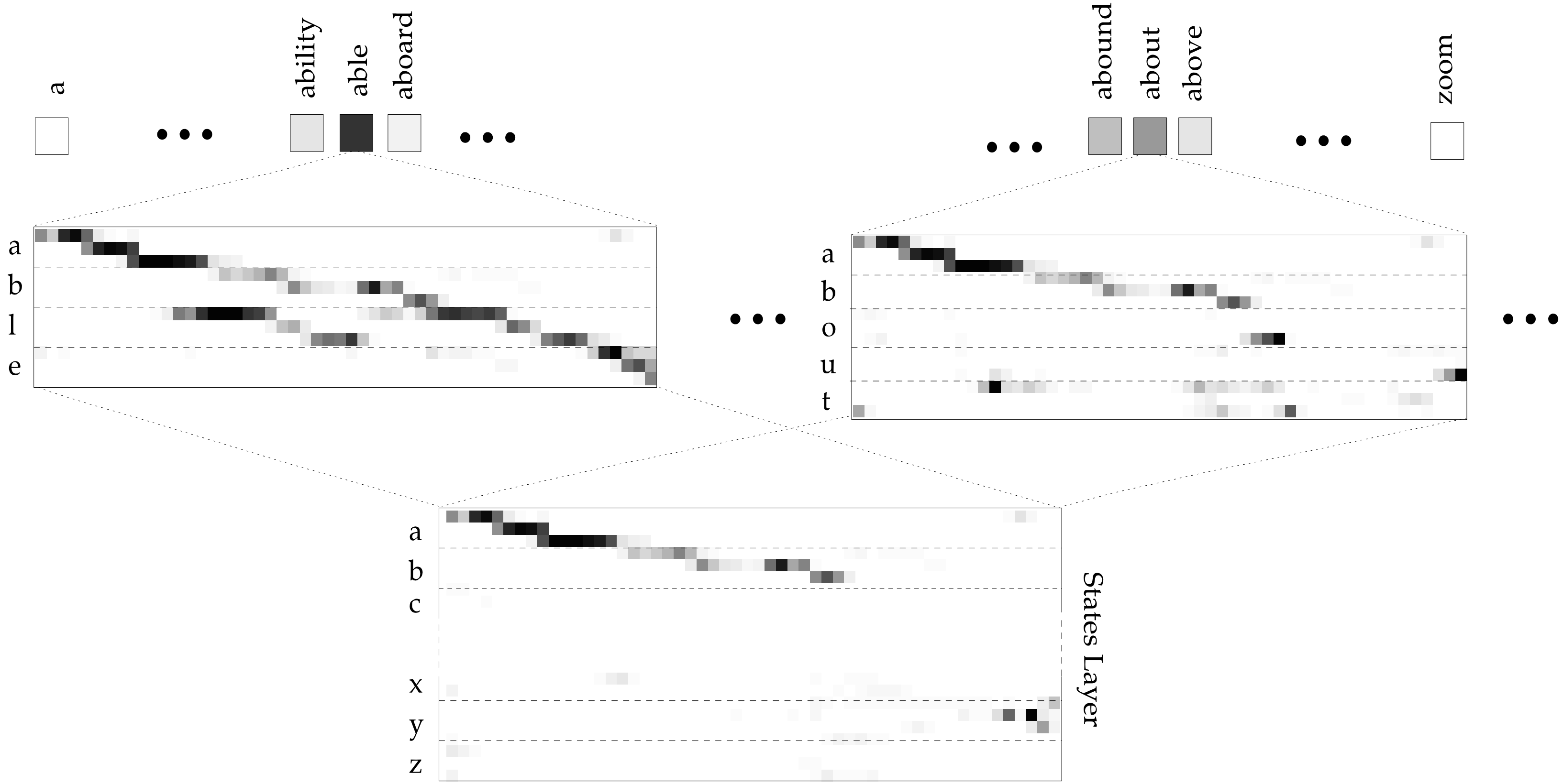
MS-TDNN Architecture



Use of Language Models and Dictionaries

Flat Search Approach

Given the probabilities of the states the score for each word in the dictionary is defined to be a Viterbi approximation of the log likelihoods of the feature vector sequence:



Karlsruher Institut für Technologie

Decoding

The Viterbi Algorithm:

– Find the state sequence **Q** which maximizes **P(O, Q | λ)**
– Similar to Forward Algorithm except **MAX** instead of **SUM**

$$VP_t(\mathbf{i}) = MAX_{q_0, \ldots q_{t-1}} \; P(O_1 O_2 \cdots O_t, \mathbf{q}_t{=}\mathbf{i} \mid \lambda)$$

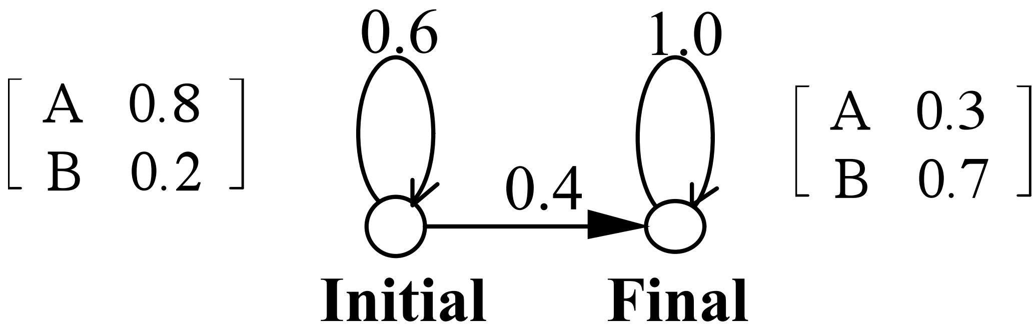
Recursive Computation:

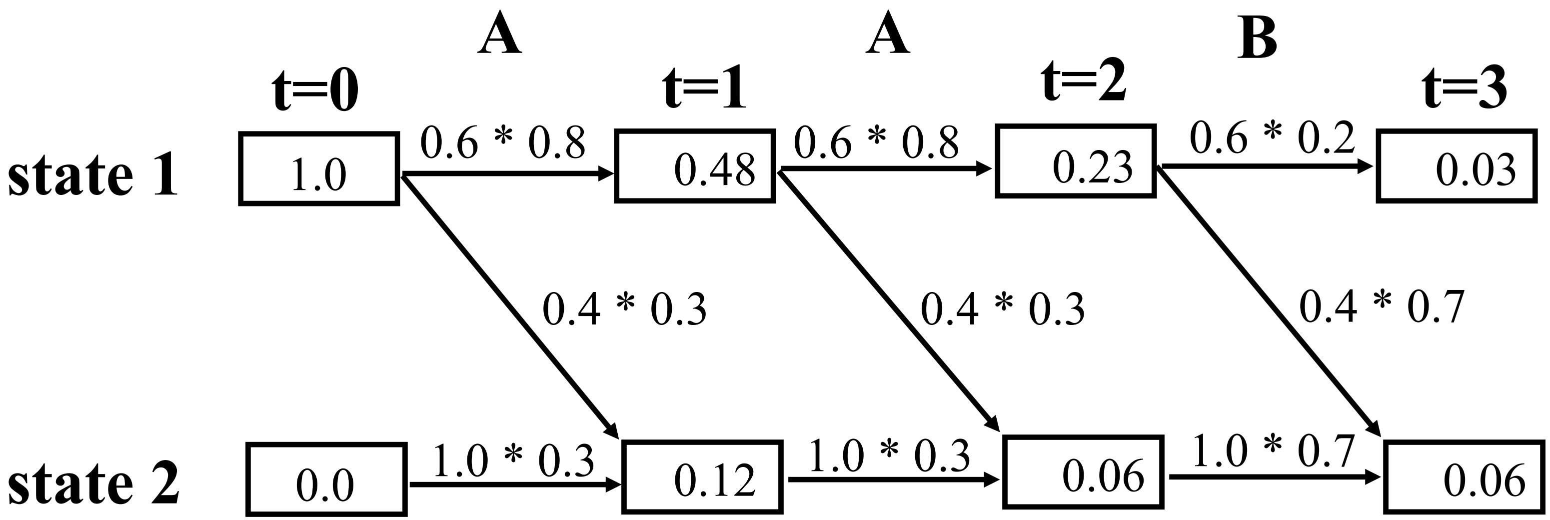
$$VP_t(\mathbf{j}) = MAX_{i=0, \ldots, N} \; VP_{t-1}(\mathbf{i}) \;\; \mathbf{a}_{ij}\mathbf{b}_j(O_t) \qquad t > 0$$

$$P(O, Q \mid \lambda) = VP_T(S_N)$$

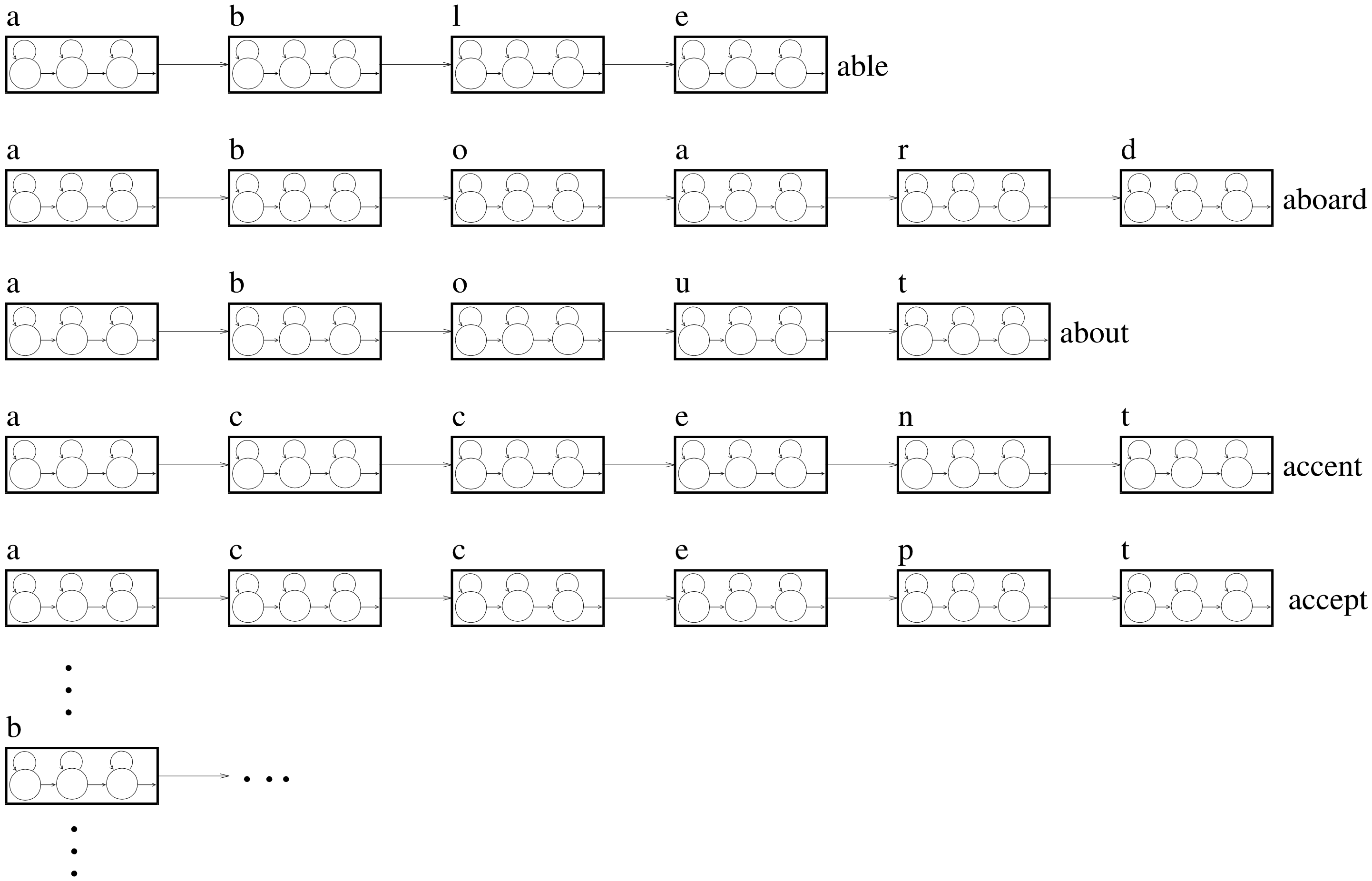
Save each maximum for backtrace at end

Viterbi Trellis



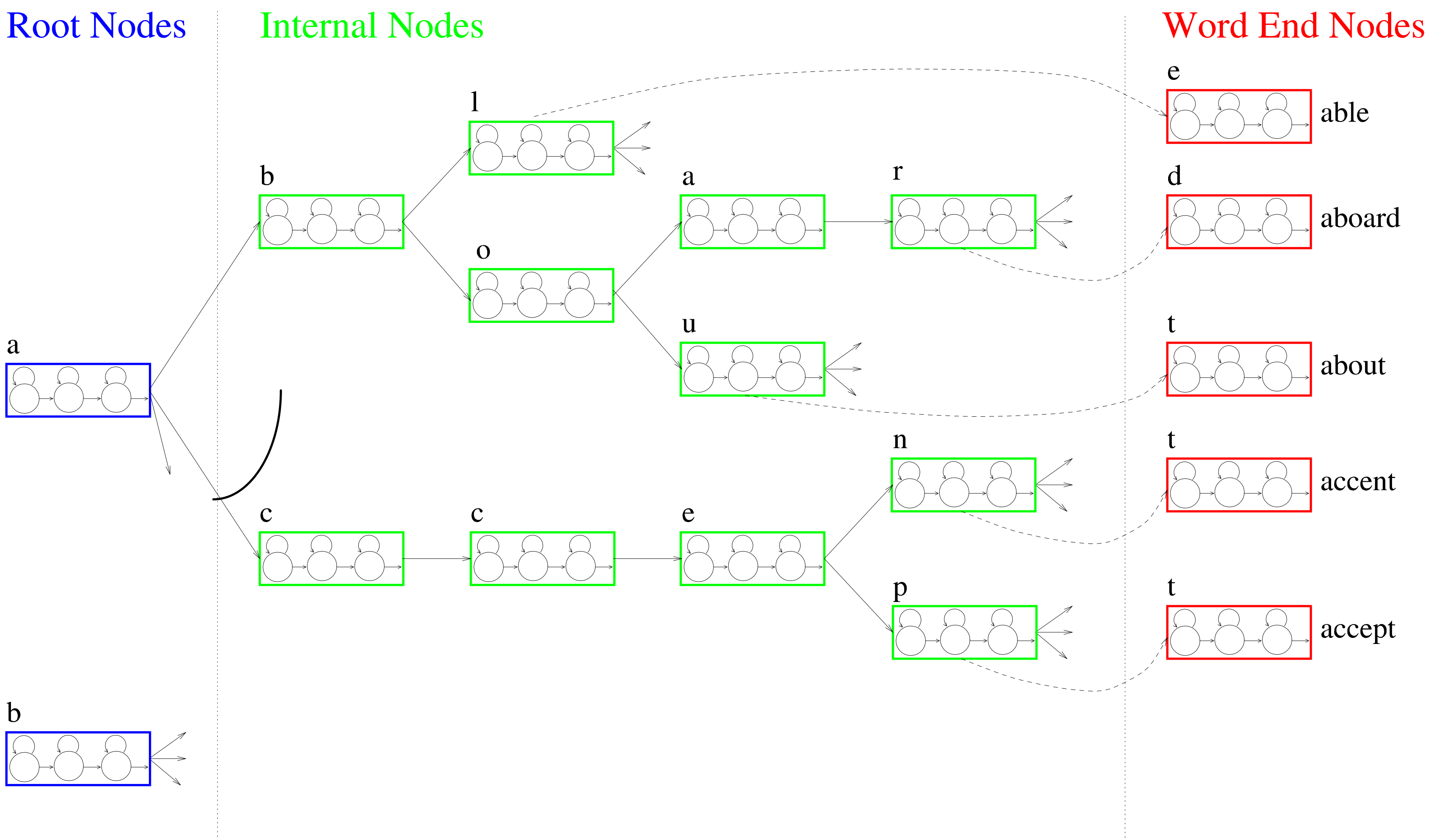


Flat Search

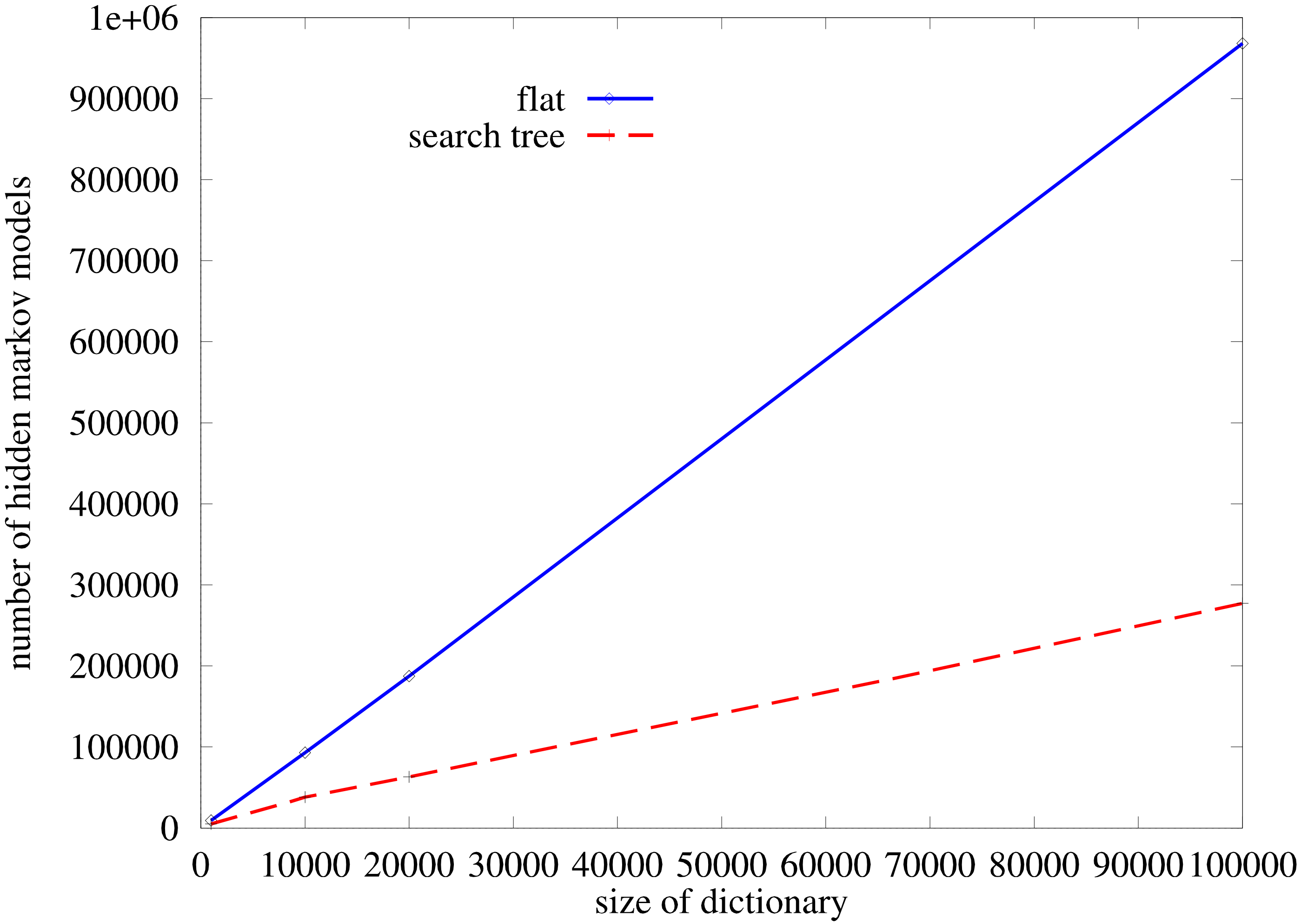


a
b
l
e
able
a
b
o
a
r
d
aboard
a
b
o
u
t
about
a
c
c
e
n
t
accent
a
c
c
e
p
t
accept
b
. . .

Tree Architecture



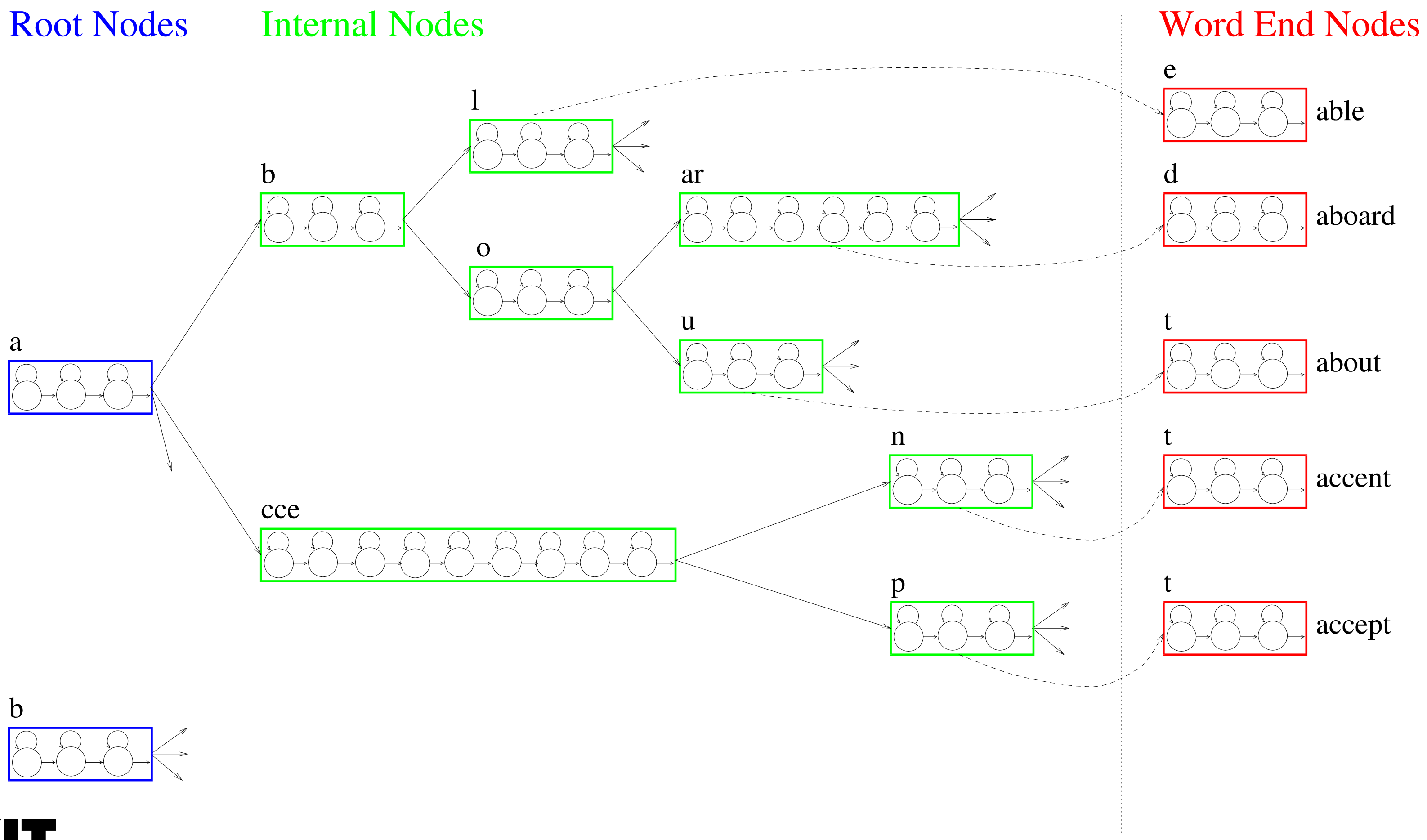
Number of Nodes



Karlsruher Institut für Technologie

Compressed Tree Architecture

- Linear lists of nodes can be merged into single nodes
- This results in fewer active nodes during search



Tree Search Algorithm

Problem:

- The tree structure in itself does not yield enough of a benefit with respect to run-time efficiency
- There is still a linear scaling of run-time with the dictionary size

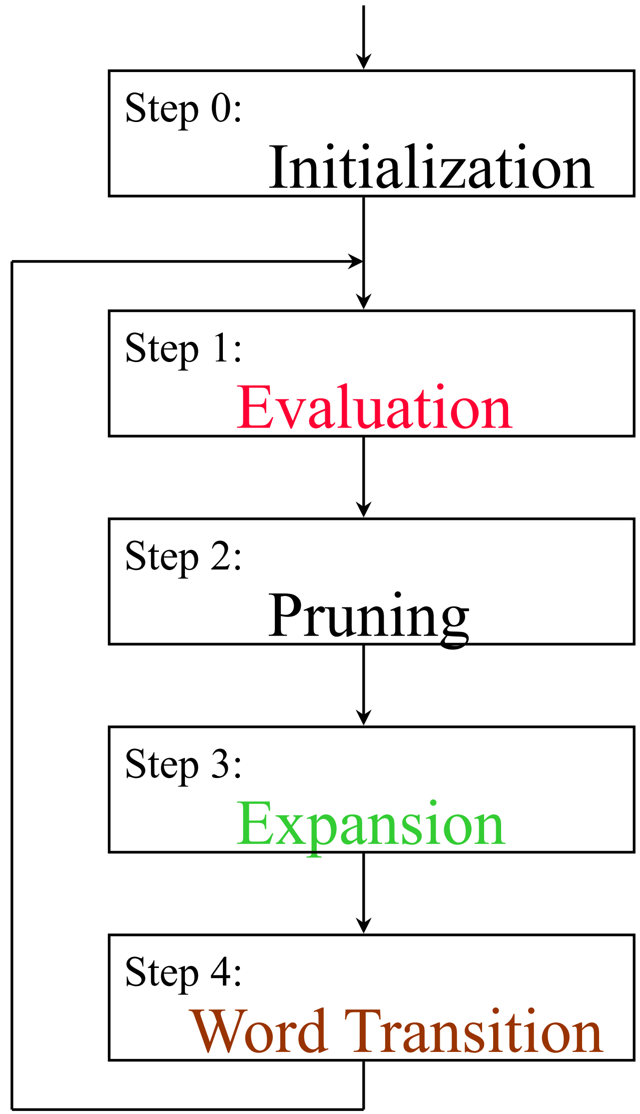
Solution:

- We give up evaluating all nodes (HMMs) of the search tree
- This leads to active and inactive nodes and a set of pruning rules which specify when to turn on an inactive node and when to turn off an active one

Initialization and Basic Concept

- The search is initialized by setting all root nodes to be active and all other nodes (internal and word end nodes) to be inactive

- Then for each frame of the states layer of the neural network the algorithm goes through steps 1-4



Tree Search Algorithm for One Frame

① Evaluation

- For each active node compute a viterbi step to find the accumulated scores s_{ij} for the next frame
- Compute the best state score s_i within each node and the best score
  - $s' = \max s_i$
- over all evaluated nodes

② Pruning

- Deactivate all currently active nodes in the search tree where the following criterion is fullfilled
  - $s_i < s' - beam$
- I.e. all nodes whose best accumulated score is below a certain threshold will become inactive in the next frame

③ Expansion

- For each currently active node test whether a transition from its last state to the first state of any child node leads to a higher accumulated score in the first state of that child node
- If that holds and the new score is above the pruning threshold the child node is marked to be active in the next frame

④ Word Transition

- For each active word end node we test the transition from that node to any of the tree root nodes as we did in the expansion step above

Karlsruher Institut für Technologie

Adjusting Beam Sizes

- The beam size influences both the recognition accuracy and recognition time:
  - smaller beam means increased speed and decreased accuracy
  - larger beam means decreased speed and increased accuracy
- Therefore the beam has to be adjusted according to the particular needs: e.g. maximum beam for evaluations, smaller beam for run-time version of the system
- For isolated word recognition (i.e. no word transitions allowed) only one single beam for all tree nodes is used
- For continuous recognition (sentence recognition) separate beams for different node types can be used. E.g.:

  a beam for all root and internal nodes

  a beam for all word end nodes

  a beam for word transitions

Summary – On-Line Recognition

- A tree architecture reduces the number of nodes (HMMs) to be evaluated to approx. 1/3
- The tree architecture itself does not improve recognition time compared to a flat search approach
- But combined with efficient pruning techniques the search space can be reduced dramatically
- The benefit in run-time is much higher than the small decrease in recognition accuracy

Experiments (Isolated Words)

- Task:
  - isolated english words
  - writer independent
  - no restrictions on writing style
  - dictionary sizes ranging from 1,000 to 100,000 words

- Database:
  - collected at University of Karlsruhe and Carnegie Mellon University
  - mixture of german and english writers
  - 307 different writers (13,000 words)
    - 204 writers used for training (9,000 words)
    - 103 writers used for testing (4,000 words)

Karlsruher Institut für Technologie

Recognition Rates

Recognition of single symbols:

Npen++:	0-9	96.5%
	A-Z	92.7%
	a-z	91.1%

Recognition Rates

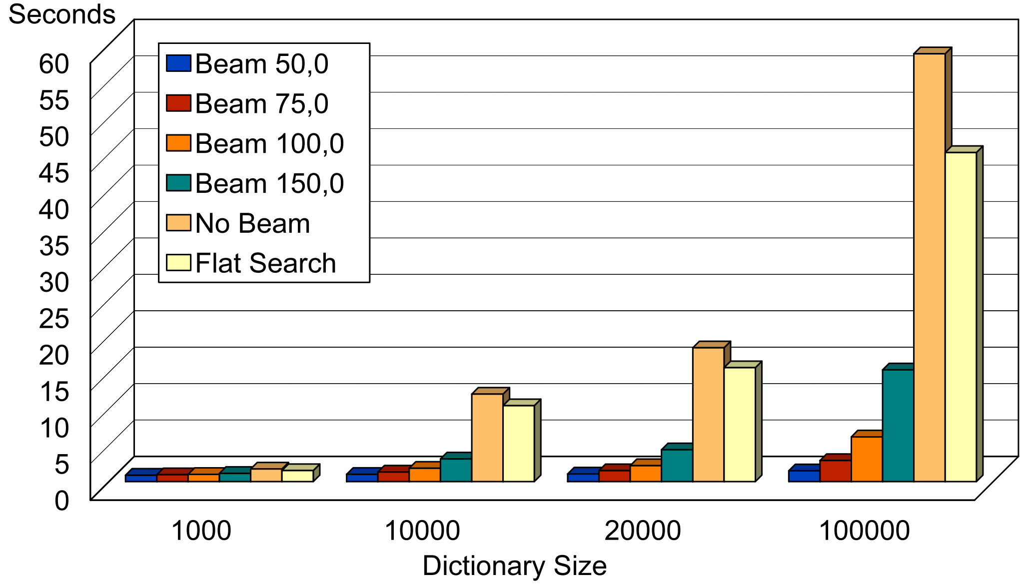
Recognition of words:

On-line recogniton rate is higher than off-line recognition rates:

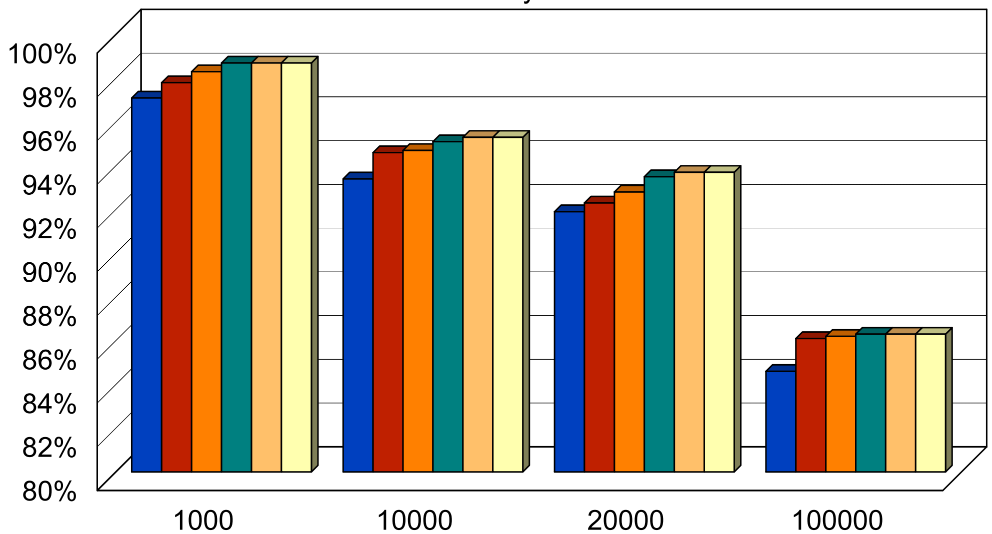
Off-line: 95% on 500 words
On-line: 95% on 5.000 words
90% on 50.000 words

Results (Isolated Words)

Recognition time versus dictionary and beam sizes



Recognition accuracy versus dictionary and beam sizes



KIT Karlsruher Institut für Technologie

Experiments (Sentences)

- Task:
  - sentences from Wall Street Journal (WSJ)
  - writer independent
  - no restrictions on writing style
  - 20,000 word WSJ dictionary

- Database:
  - collected at University of Karlsruhe and Carnegie Mellon University
  - mixture of german and english writers
  - training set:
    - ~ 20,000 isolated words
    - ~ 10,000 sentences (~ 70.000 words)
  - test set:
    - ~ 200 sentences from WSJ (from 40 different writers)

Karlsruher Institut für Technologie

Results (Sentences)

- 64% word recognition rate (without any language model)
- 85% word recognition rate (using a WSJ language model)

The word recognition rate is defined as
(#words - #insertions - #deletions -#substitutions) / #words

Remarks:
~ 90% of the errors are substitutions
~ 50% of these substitutions are caused by capitalization errors and ending-"s"

Handwriting Recognition

- On-line and Off-line recognition techniques merge:
  - extracting on-line information from static off-line data (?)
  - using off-line techniques in on-line systems
  - combining off-line and on-line recognizers (multiple experts)
- Issues:
  - Techniques: Stroke-based recognition, Context Dependence, Adaptations, ...
  - Benchmarks
  - Repair
  - The New Word Problem
  - Other symbol sets: Formulas, Gestures, Drawings, Icons, ..
  - Abbreviations
  - Combinations of symbol sets... *All* Pen Based Activity
  - Cursive off-line handwriting

Systems

- Individual Characters
  - Many PDA's (particularly Japan, China)
  - Simplified Characters: Graffiti
- Continuous Handwriting
  - TDNN Based:
    - Npen++
    - Similar Architectures: LeCun, Rumelhart
  - HMM Based:
    - BBN, …

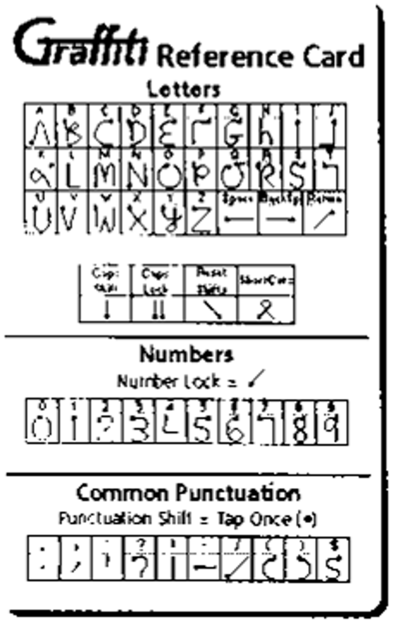
The Graffiti Recognition System

PDAs könnten aus ihrer engen Marktnische heraustreten

Graffiti auf dem Pen-Computer macht Schrifterkennung einfach

Von Fereydun Khunde*

Erweiterte Funktionalitäten machen Personal Digital Assistants (PDAs) attraktiver. Zugleich haben die Software-Entwickler das alte Problem der Schrifterkennung heute wesentlich besser im Griff. Ein besserer Zuspruch der potentiellen Käufer scheint nun möglich.



Off-Line Recognition

- Optical Character Recognition (OCR)
  - Mostly Printed Text
- Extract Text Sequence
  - Document/Scene Analysis
  - Extract/Scan Bitmap
  - Extraction and Computation of Features
  - Representation is Spatial, *not* Termporal
- Recognition Algorithm
  - Deal with Shift Invariance
  - Integrate Characters into Words (segmentation!)

Off-line Handwriting Recognition

- Possible applications include
  - check reading
  - postal address reading
  - document analysis, ...
- Input consists of scanned bitmaps without any temporal information
- Eventually location of handwriting needs to be found (document analysis)
- Stroke order doesn´t influence recognition
- But: problems through overlapping or touching characters and noisy input

Document Analysis

Figure 9. Original.

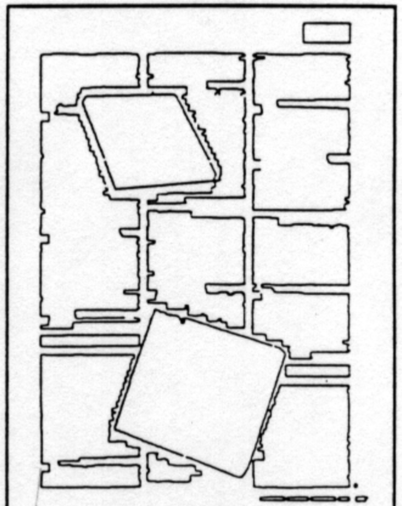


Figure 10. Result.

Figure 11. Original image with 15° skew.

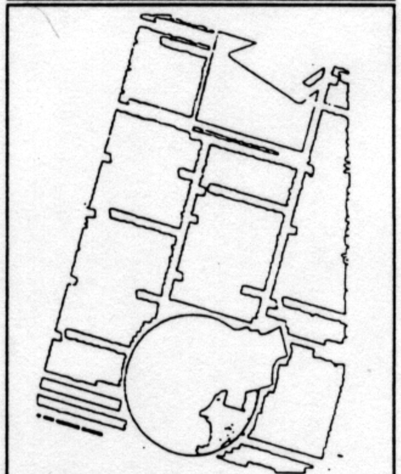
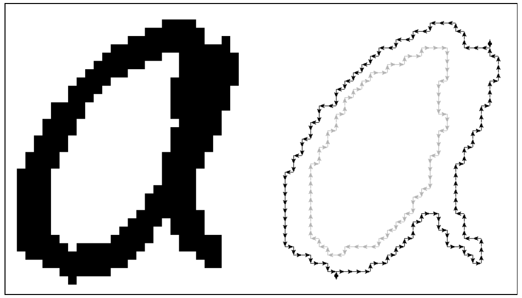


Figure 12. Segmentation result.

Off-line Preprocessing

Binary Connected Components



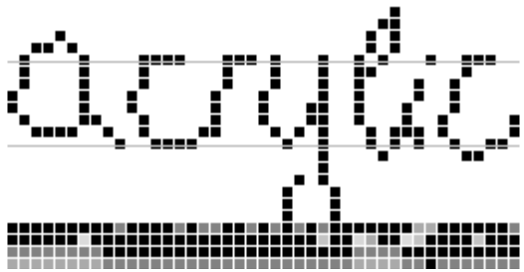
Baseline Normalization

Deskewing

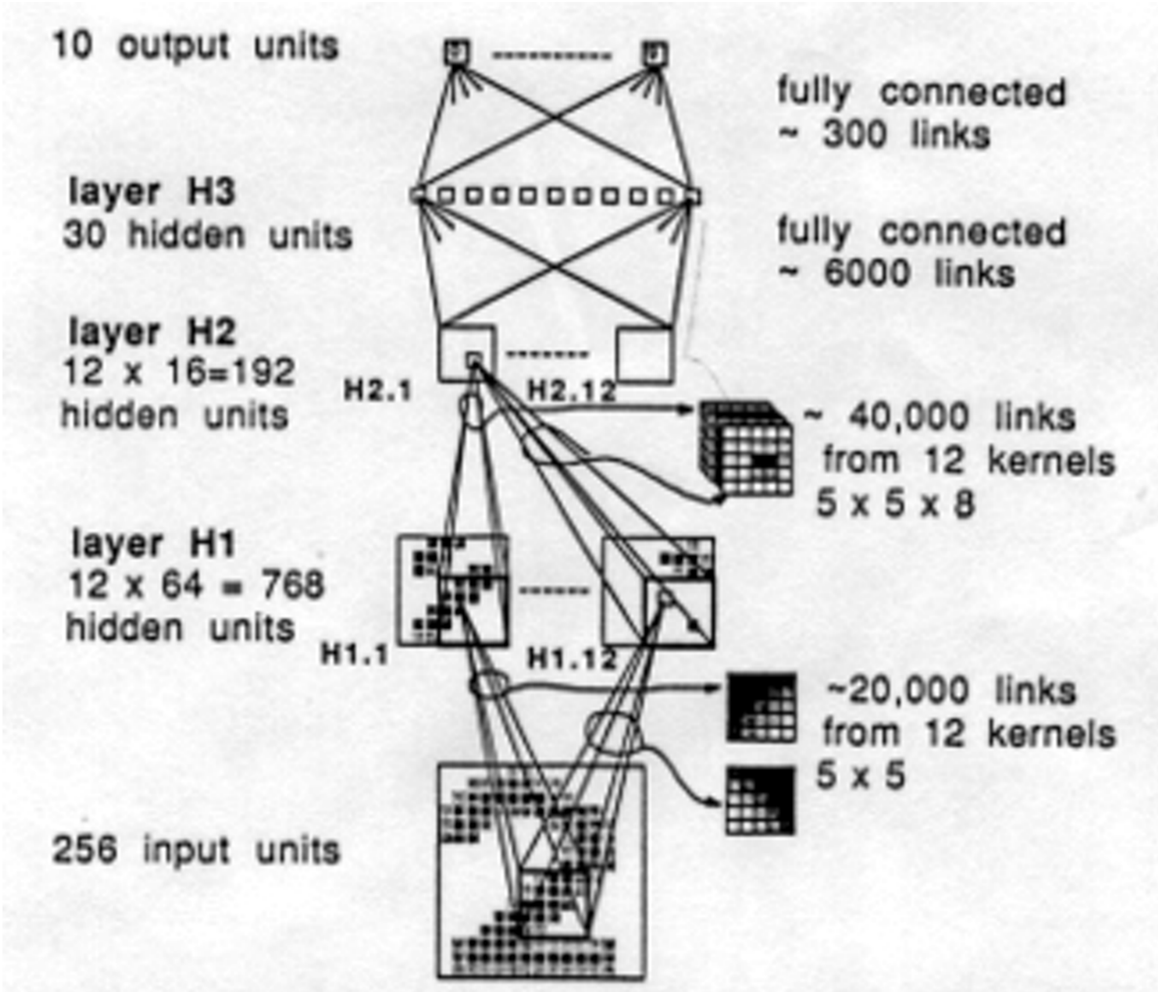
Skeletonization

Approximation

Feature Extraction

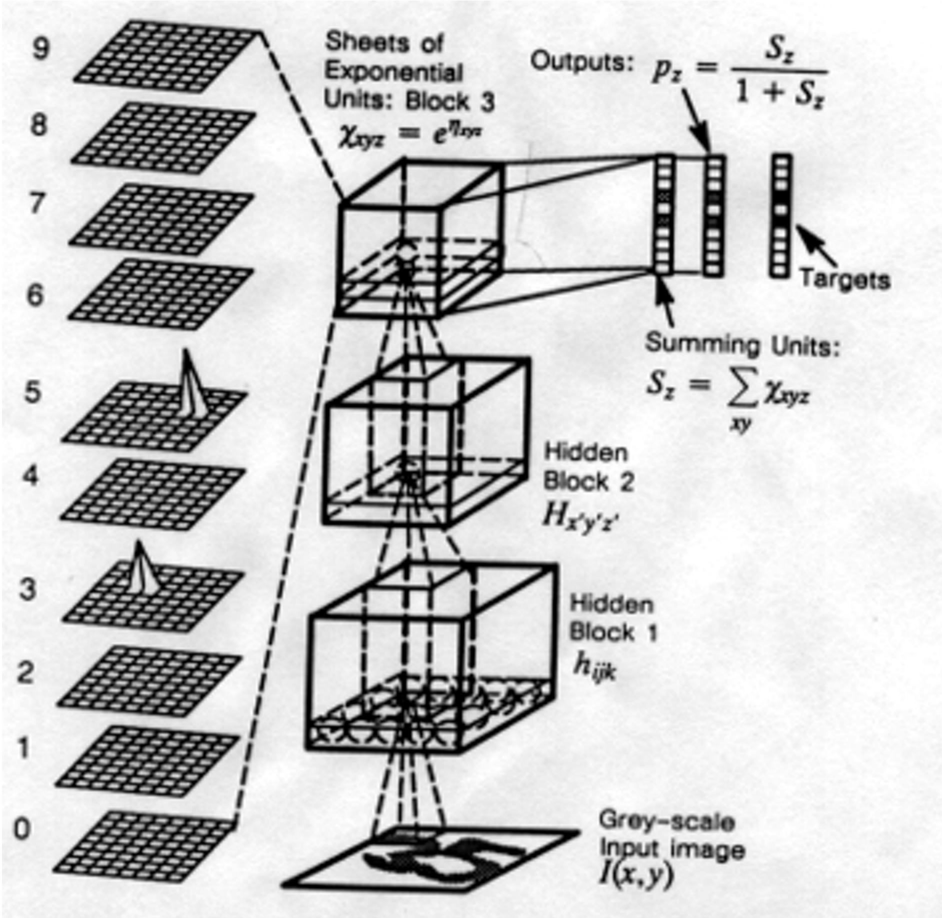


LeNet Architecture



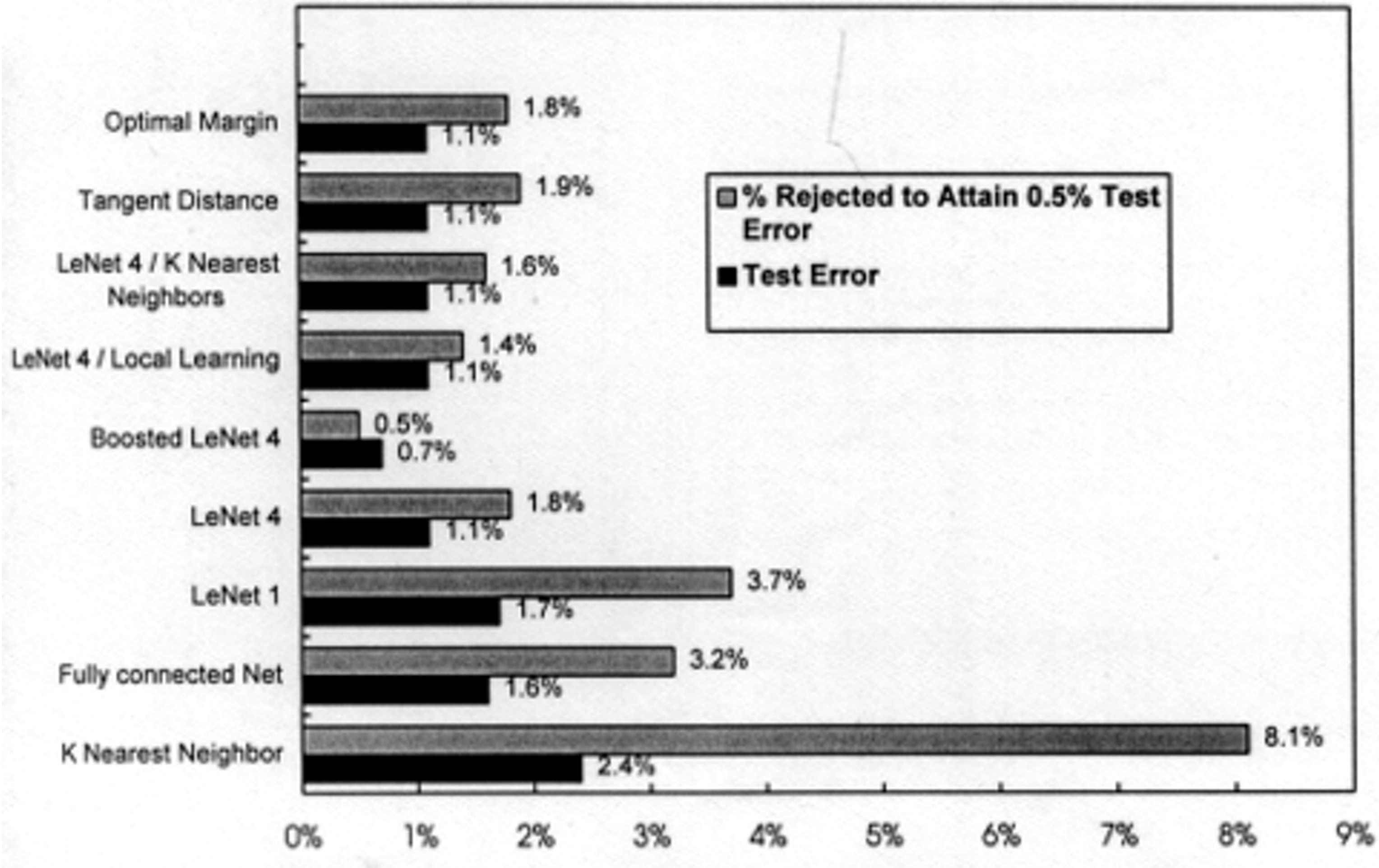
Reference: Yann LeCun et al. „Handwritten Zipcode Recogntion with Multi-layer Networks“, Proceedings of the ICPR-90, Atlantic City, 1990

Rummelhart Architecture



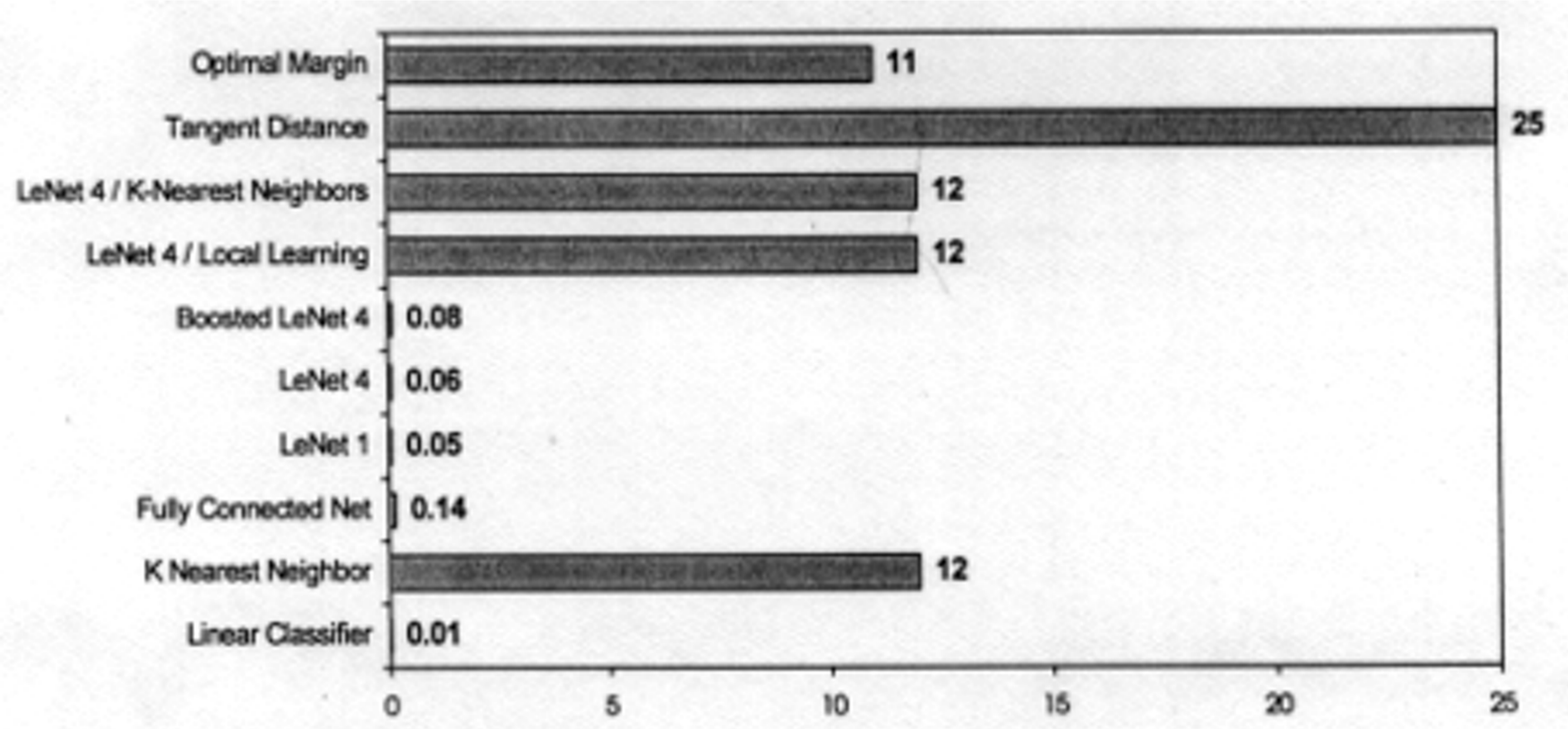
Reference: J. Keeler, D. E. Rummelhart. „A Self-Organizing Integrated Segmentation and Recognition Neural Net", Advances in Neural Information Processing Systems, Morgan Kaufman, 1991.

Comparison of Classifier Methods

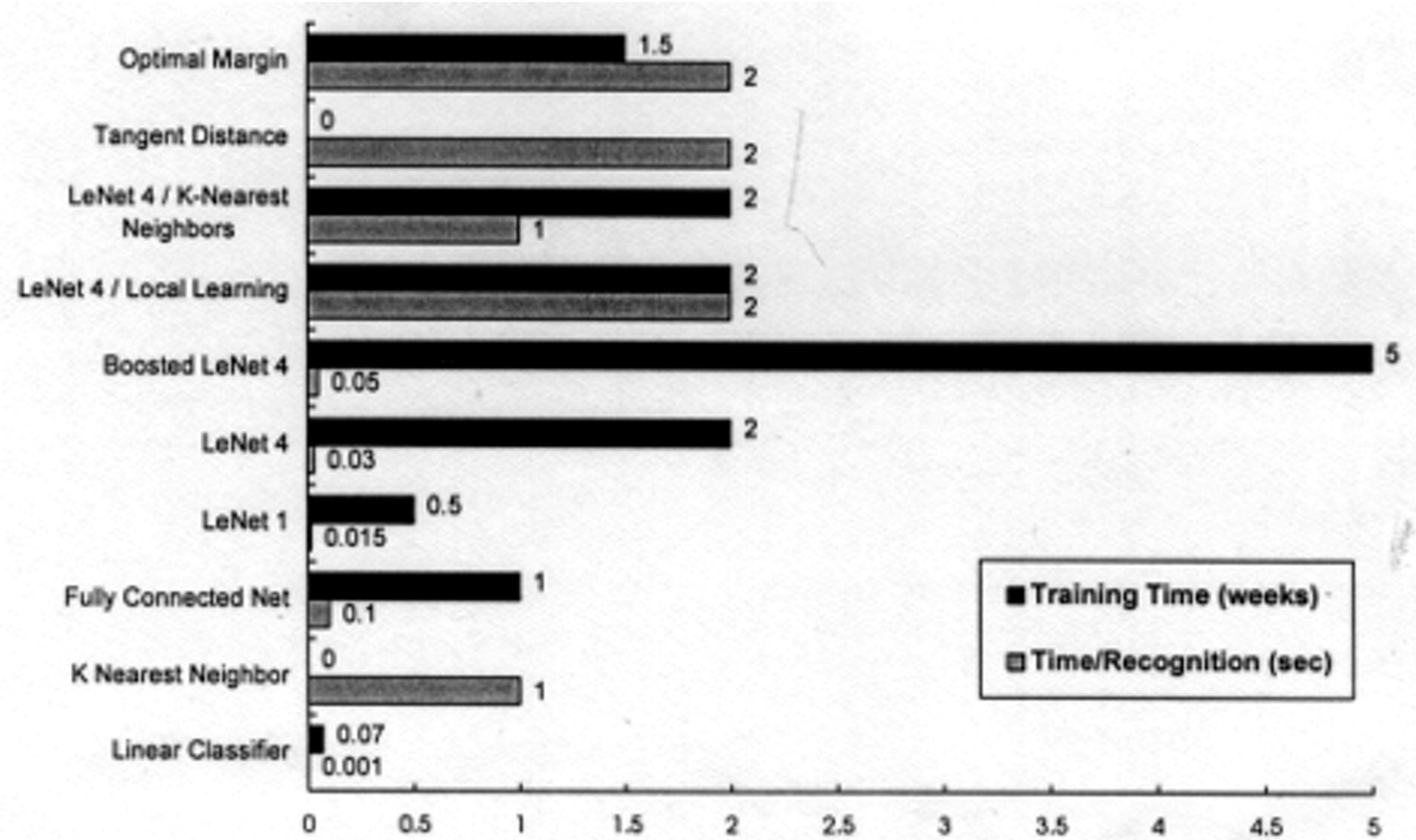


- Reference: L. Bottou et al. „Comparison of Classifier Methods: A Case Study in Handwritten Digit Recognition“, Proceedings of the ICPR-94, Jerusalem 1994)

Memory Requirements



Training and Run Time



The Language Challenge
Signs, Visual Text

Signs


ご迷惑をお掛けして
まことにすみません
安全には特に気を付けて
作業しておりますから
高視眼鏡
NATION
GARE DE L'EST
GARE DU NORD
PERE LACHAISE

Challenges of Sign Translation

- Detection
  - No motion information
  - No approximate shape and position information
  - No color reflectance information
- Recognition
  - Blured image
  - Low resolution
  - Character deformation
  - Reflections, Colors, Fonts
- Translation
  - Short sentence/Phrase
  - Abbreviation

Sign Translator

Travel with Confidence MLT Mobile Language Translator

Traveling to a foreign country can be a confusing and disorienting experience. Signs written in a different language—often with different characters—can be an intimidating obstacle to information. Without the ability to translate foreign languages, signs become useless while vital—perhaps even life-saving—information remains out of reach.

Until now, no satisfactory solution has been available. Dictionary look-up is awkward and time consumingat best, and nearly impossible when unfamiliar characters are involved.

Thankfully, **Mobile Technologies** provides the solution: **Mobile Language Translator**®, A PDA based road sign translator equipped with a camera attachment and MobileTech's proprietary software. Using the device is extremely easy: just point the camera at the sign, take a picture, and the translation appears on the screen, right under the image of the sign. Simple as that.

Want to know the meaning of an individual character?
Simple: just circle the character on the PDA screen, and the translation appears.

Want to know how it is pronounced?
Easy: Select speech synthesis and hear how it sounds, with a phonetic spelling of the word appearing on-screen.

Travel with Confidence and feel at home in foreign lands with **Mobile Language Translator**® from **Mobile Technologies**.

Hardware Requirements:
Mobile Language Translator® runs on most PDAs running Windows CE. A minimum of 16 MBytes of memory must be provided. For best results a PDA with camera attachment such as the HP Jornada should be used.

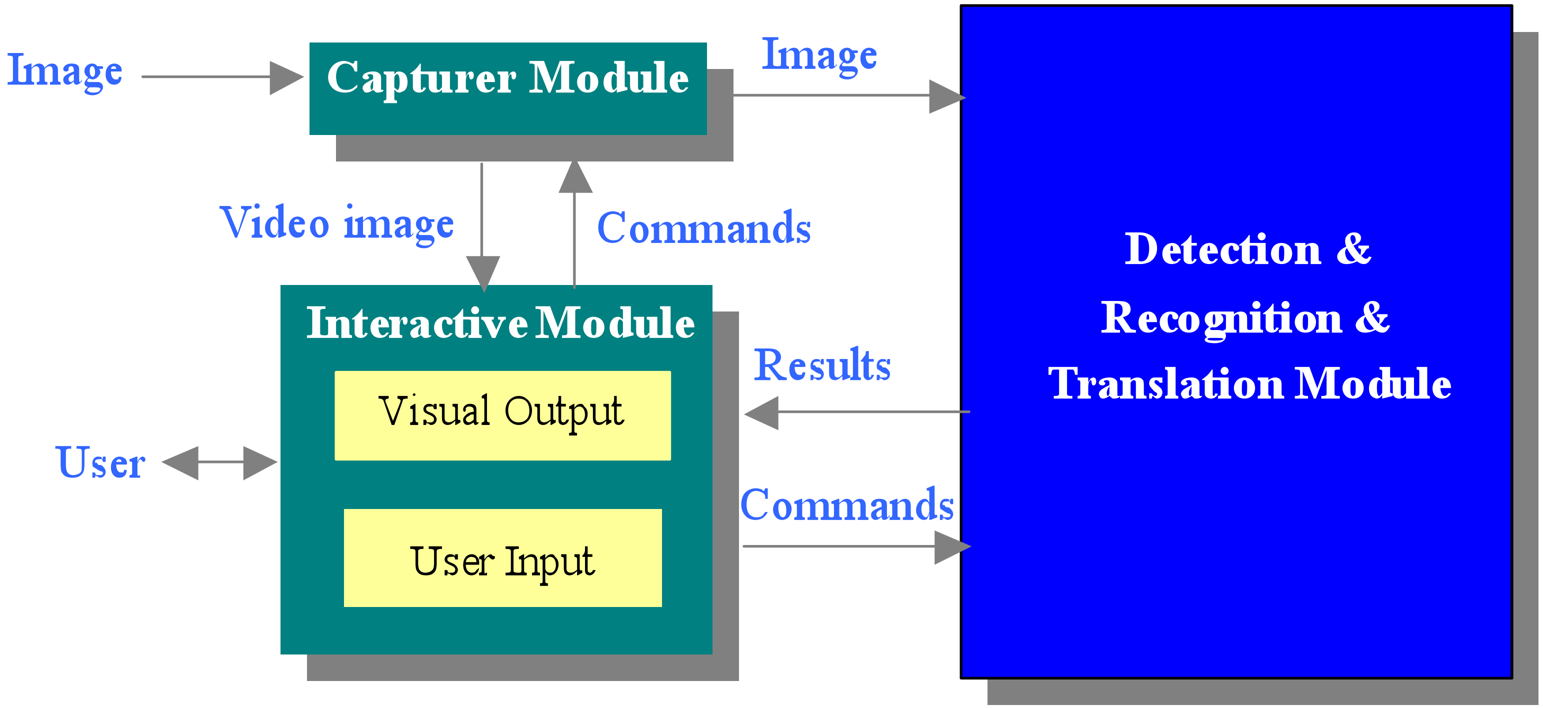
Capabilities:
Language support: Chinese Input, English or Chinese Output
Character Support: More than 3,000 Chinese characters (covers most common signs)

Other languages in preparation; Please call MobileTech for details.

MOBILETECHNOLOGIES L.L.C.

619 Windsor Ave
Pittsburgh, PA 15221
Phone: 412-351-5529
FAX: 412-351-5410
Email: nonchin@icubed.com

Sign Translator - Architecture



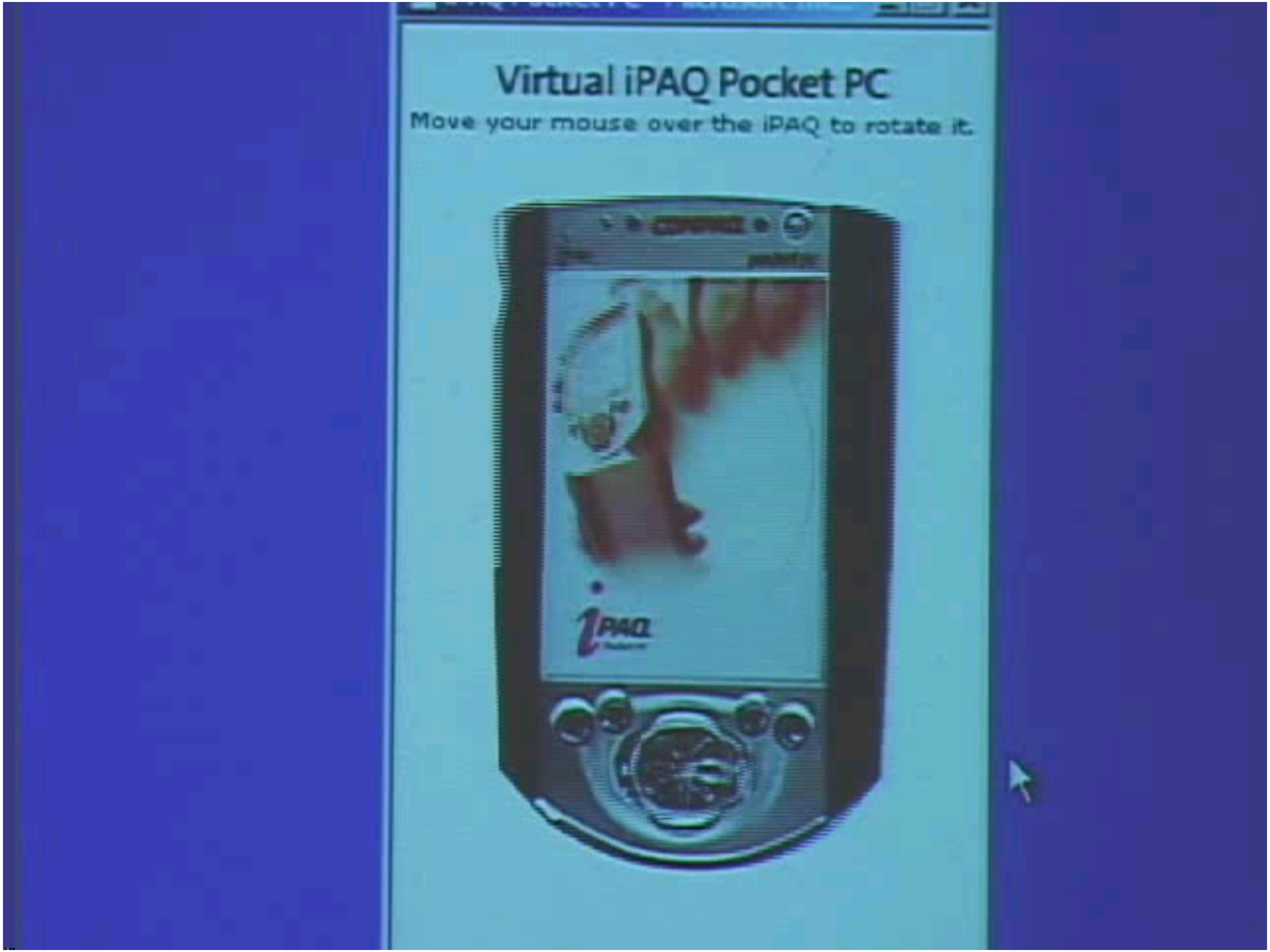
Sign Detection

- A Hierarchical Approach
  - A multi-resolution edge detection algorithm
  - Adaptive searching in the neighborhood of initial candidates based on layout syntax
  - Layout analysis of the detected sign areas

Character Recognition

- Intensity-based Approach
  - Feature from Gabor Transformation
  - LDA for the feature selection
- Result
  - Character set: 3755 Level 1 Chinese characters in Chinese national standard character set GB2312-80
  - Accuracy: 92.4%

Some Results


Virtual iPAQ Pocket PC
Move your mouse over the iPAQ to rotate it.