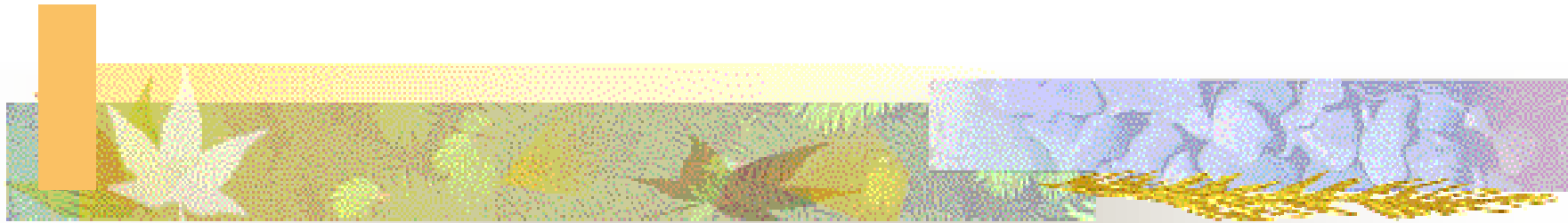


A Handwriting Interface to Mathematical Expressions



May 11, 2006
Western Ontario University

Masakazu Suzuki
Kyushu University,
InftyProject



Plan of the talk

■ About InftyProject

- Research scope: Mathematical Information Processing
- 2. Software release: *InftyReader*, *InftyEditor*, etc.

■ Handwriting Interface

- Online handwritten math recognition / printed math recognition,
- Character recognition (*Voting method*),
- Formula structure recognition (*Automatic writing method*),
- Adaptability (*User's script registration*),
- Connection to computer algebra system.



Plan of the talk

■ About InftyProject

- Research scope: Mathematical Information Processing
- 2. Software release: *InftyReader*, *InftyEditor*, etc.

■ Handwriting Interface

- Online handwritten math recognition / printed math recognition,
- Character recognition (*Voting method*),
- Formula structure recognition (*Automatic writing method*),
- *Adaptability (User's script registration)*,
- *Connection to computer algebra system.*



1. Infty Project

- The beginning :
 - Started as a research project to help visually impaired people in scientific fields in 1995.
- Current research subjects :
 - Recognition and understanding of math documents,
 - User interface and data conversion, etc.
- Policy:
 - Priority in practical system development.



1. Infty Project

- URL : <http://www.inftyproject.org>

[Go](#)



1. Infty Project

Software:

- Mathematical Document Digitization
 - *InftyReader* : Mathematical document reader (Paper → XML)
- Interface to Mathematics
 - *InftyEditor* : Authoring tool, Handwriting interface for math,
 - *ChattyInfty*: Voice output, Braille output, etc.,
 - *AsirPad* : Handy computer algebra tool (PDA),
 - *ActiveInfty*: Math Edit Box on the Web.



1. Infty Project

Software:

- Mathematical Document Digitization
 - *InftyReader* : Mathematical document reader (Paper → XML)
- Interface to Mathematics
 - *InftyEditor* : Authoring tool, [Handwriting interface for math](#),
 - *ChattyInfty*: Voice output, Braille output, etc.,
 - *AsirPad* : Handy computer algebra tool (PDA),
 - *ActiveInfty*: Math Edit Box on the Web.



2. Handwriting Interface for math

- Demonstration...



2. Handwriting Interface for math

Comparison with printed math recognition:

- Printed math document recognition
 - Difficulty : *Noise*, Variation of *layout* and *fonts*, etc.
 - Needs *speed* and *high accuracy* (99.5% - 99.9%),.
- Online recognition of handwritten math
 - Main Difficulty : *Distortion* of symbol shape, size and position,
 - Advantage in char recognition:
 - Accuracy 95% is good, 90% is already usable,
 - *stroke* information is usable,
 - Correction method of *structure analysis error* is important,



2. Handwriting Interface for math

Some Features of our method:

- Character and Symbol recognition
 - Combination of two different methods (*Voting*),
 - Restriction of usable symbols,
- Structure Analysis of Math Expressions
 - *Automatic rewriting method*
 - Accuracy of structure analysis,
 - Immediate correction of recognition error,
 - Subscript/Superscript decision algorithm.



2. Handwriting Interface for math

Some Features of our method:

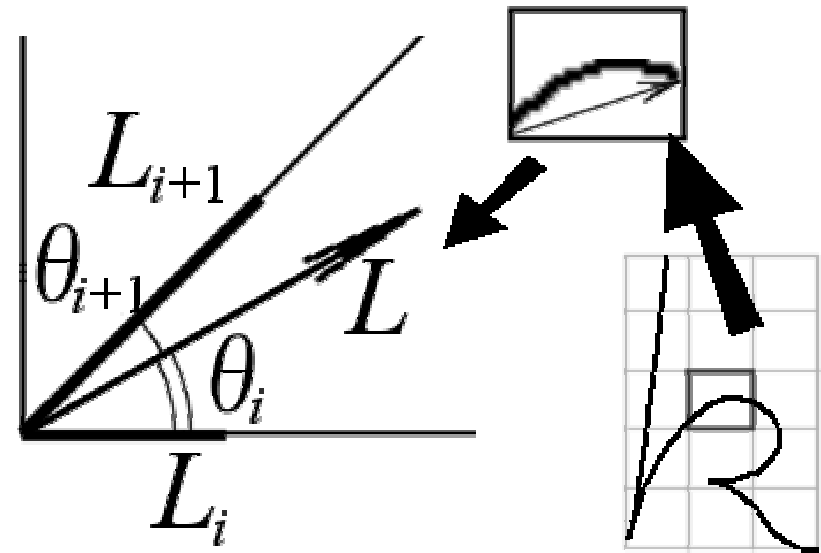
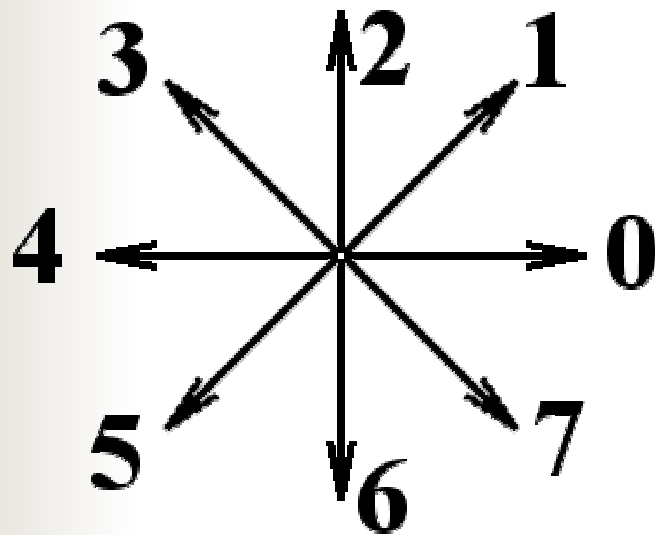
- Character and Symbol recognition
 - Combination of two different methods (*Voting*),
 - Restriction of usable symbols,
 - *Adaptability (User's script registration)* --- new
- Structure Analysis of Math Expressions
 - *Automatic rewriting method*
 - Accuracy of structure analysis,
 - Immediate correction of recognition error,
 - Subscript/Superscript decision algorithm.



3. Character Recognition

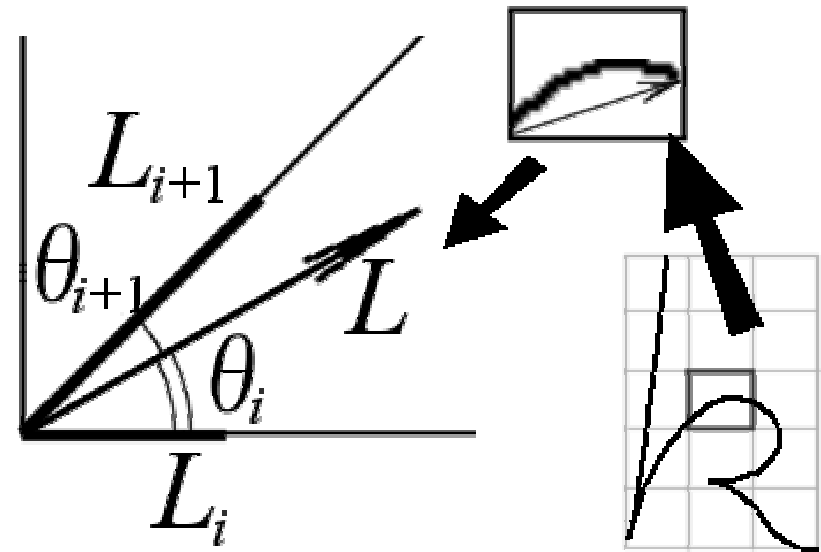
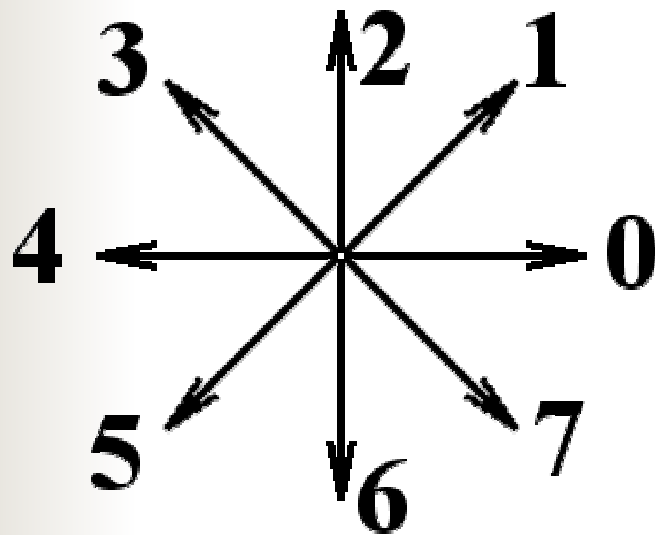
- To get recognition results by *voting*, we implemented two different methods of character recognition using:
 1. Direction element feature,
 2. Matching of segmented stroke sequence.

3.1. Direction Element Feature



- A character is separated into 3×5 blocks and each block has 8 direction elements.

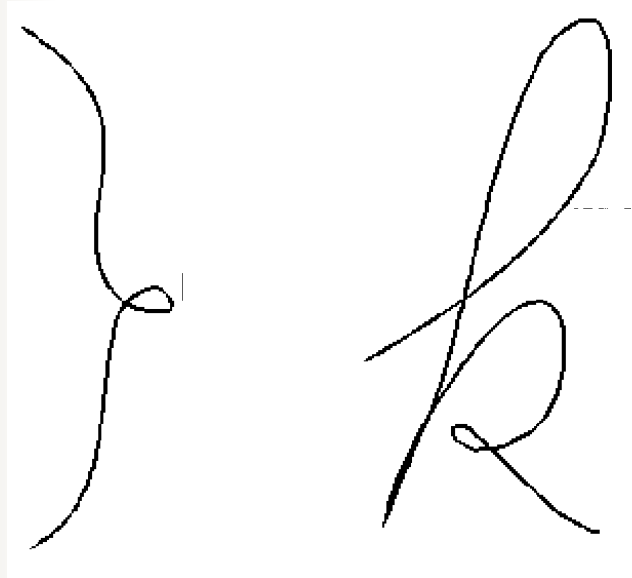
3.1. Direction Element Feature



- That is, we consider 120 dimensional vector as a direction element feature.

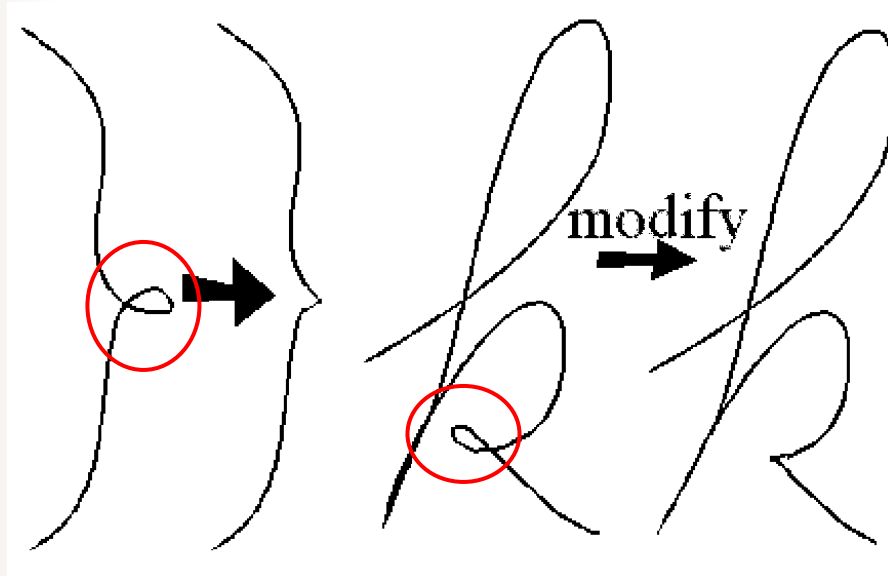
3.2. Matching of Segmented Stroke Sequence

- A written stroke is modified and segmented as follows:



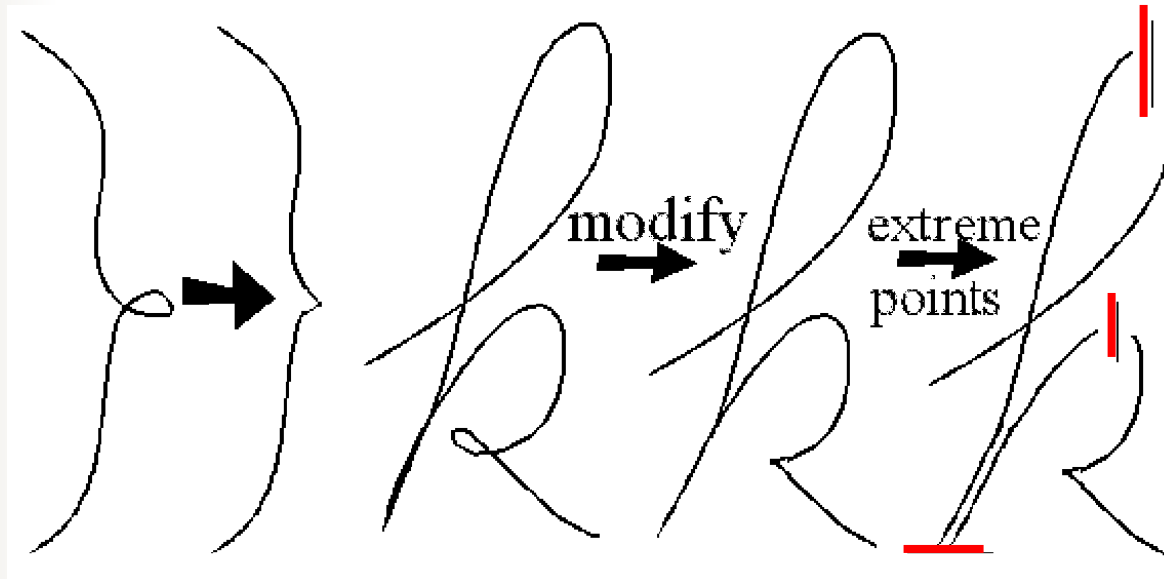
3.2. Matching of Segmented Stroke Sequence

1. Modify small loops to cusps.



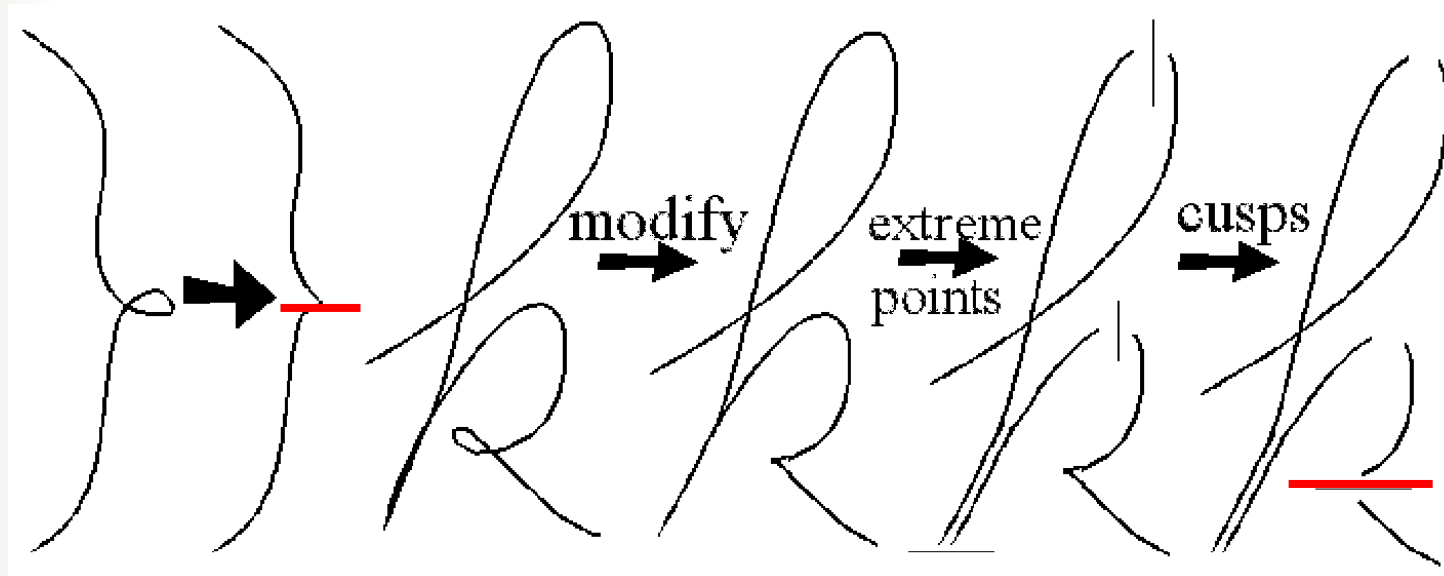
3.2. Matching of Segmented Stroke Sequence

2. Segment the stroke at extreme points on the vertical coordinate.



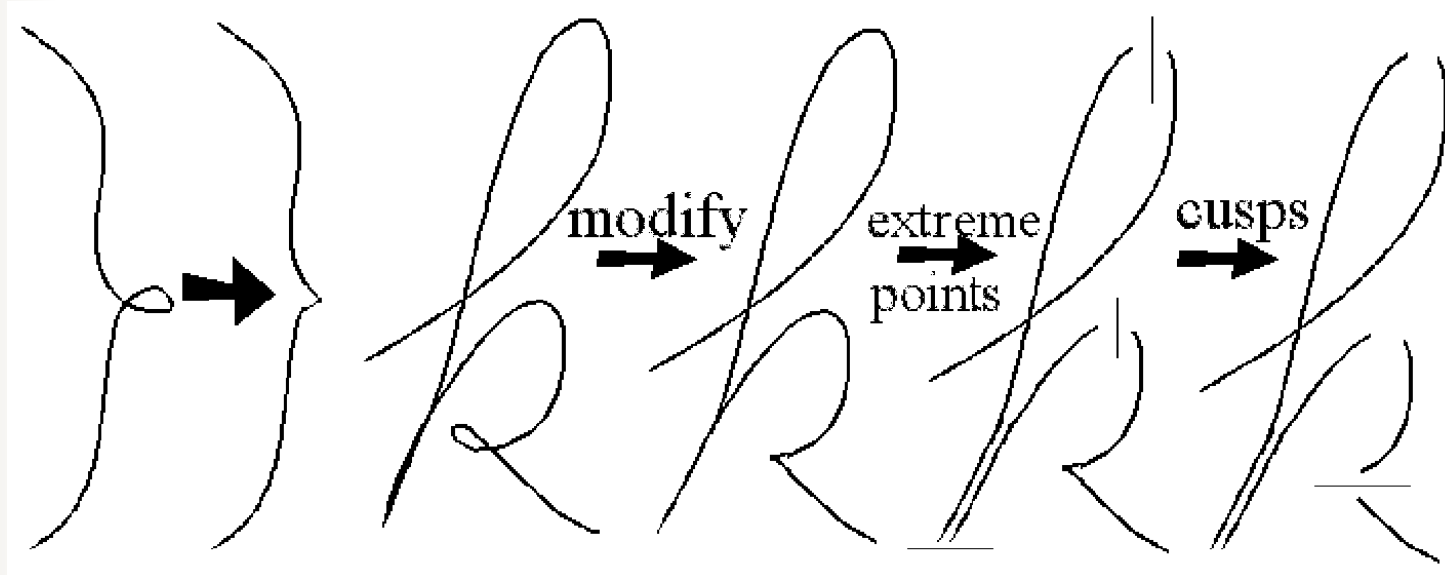
3.2. Matching of Segmented Stroke Sequence

3. Finally, segment at cusp points again.



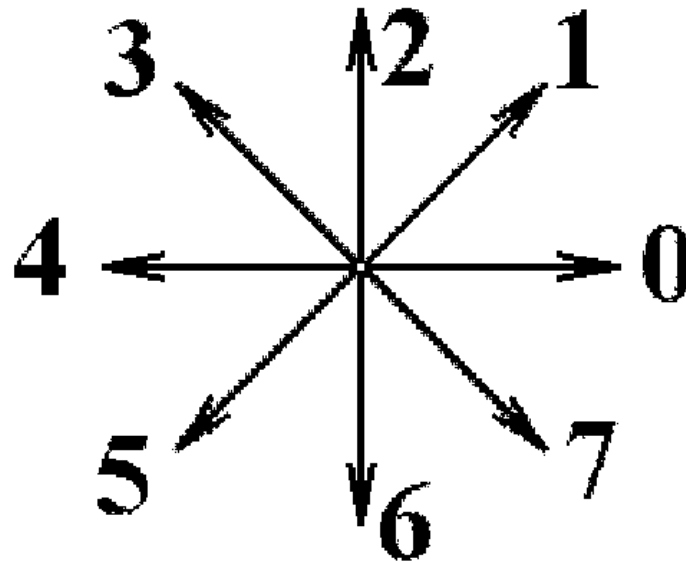
3.2. Matching of Segmented Stroke Sequence

- Segmented strokes are classified into 28 patterns.



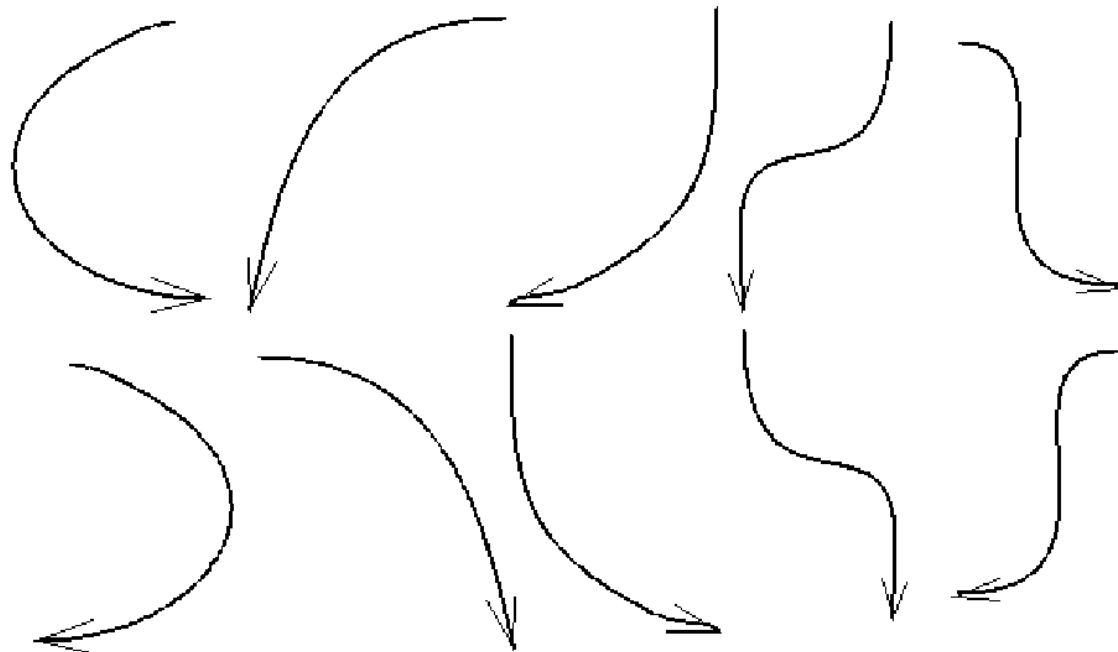
3.2. Matching of Segmented Stroke Sequence

1. Straight lines are classified into 8 class by their direction.



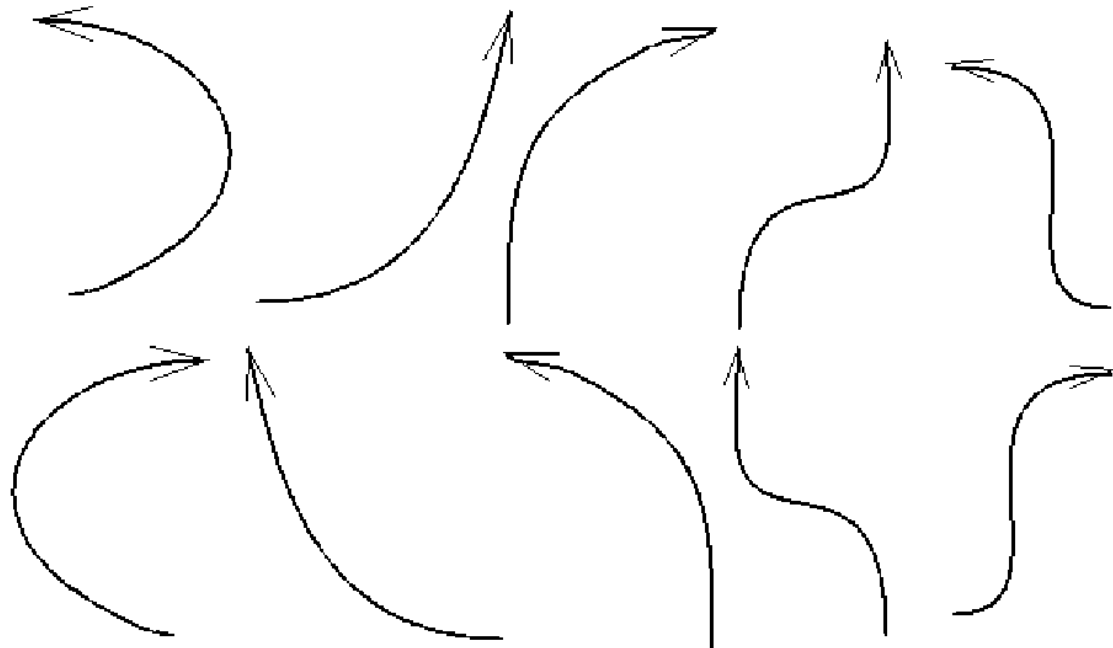
3.2. Matching of Segmented Stroke Sequence

2. Winding downstrokes are classified into 10 patterns.



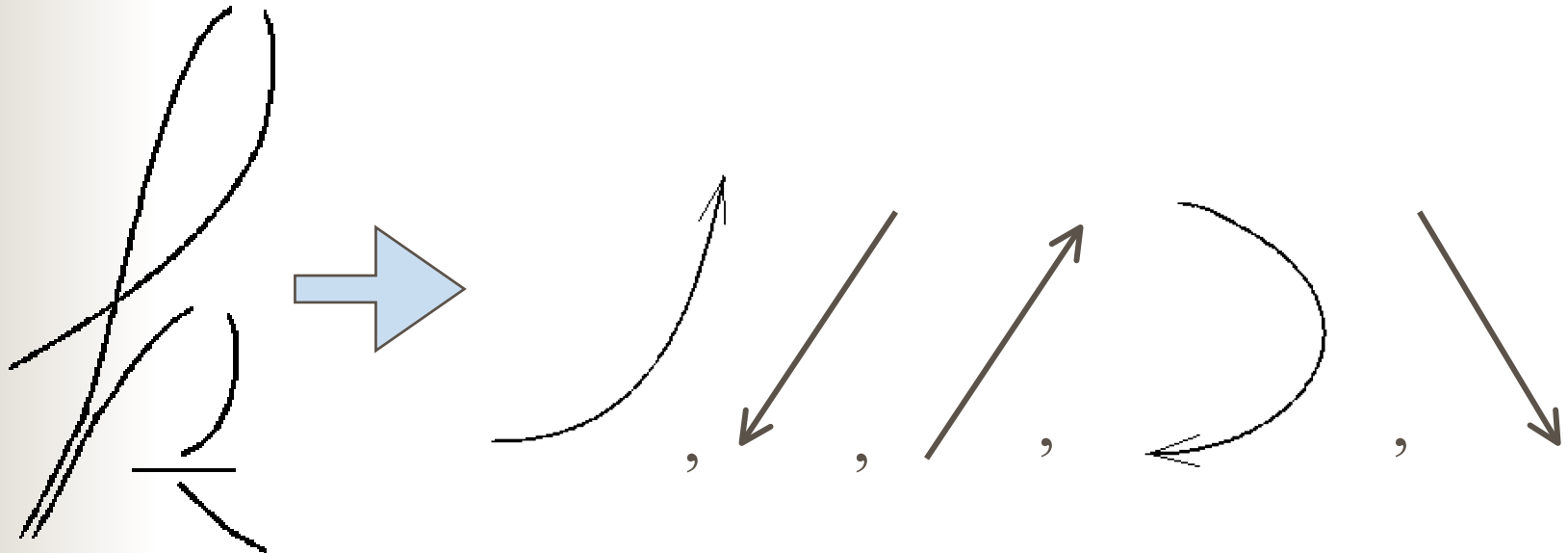
3.2. Matching of Segmented Stroke Sequence

3. Winding upstrokes are also classified into 10 patterns similarly.



3.2. Matching of Segmented Stroke Sequence

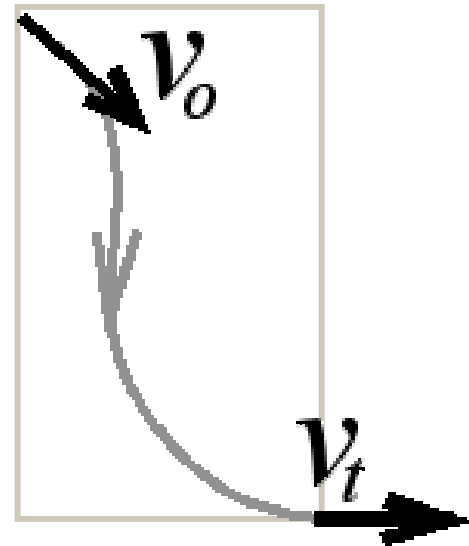
- The character recognition is done by the matching of the sequences of segment stroke patterns thus obtained.



3.2. Matching of Segmented Stroke Sequence

To calculate the recognition cost,
the following features are used:

1. segment strokes' aspect ratios
2. the positions of their bounding rectangles,
3. the directions of their original vectors v_o and terminal vectors v_t .





3.3. Voting

- Each of the two recognition methods returns *three* ordered candidates with costs.
- The *voting cost* of the candidate
=the ratio of its cost to the third candidate's.
- The final recognition results is determined by the *ascending order* of the *sum* of the voting costs of the two recognition methods.

3.4 Restrictions of Usable Characters

Alphabets	$a, b, \dots, z, A, B, \dots, Z$
Numerals	$0, 1, 2, \dots, 9$
Greek letters	$\alpha, \beta, \gamma, \delta, \varepsilon, \theta, \lambda, \mu, \pi, \phi, \varphi, \psi, \omega, \Phi, \Psi, \Omega$
Special symbols	$\sqrt{}, \int, \Sigma, \infty, \partial$
Operators	$+, -, \pm, \times, \div, /, \cup, \cap$
Relational operators	$=, \neq, \equiv, \not<, \leq, >, \geq, \in, \ni, \subset, \supset, \rightarrow$
Function names	$\lim, \log, \sin, \cos, \tan$
Parenthesis	$(,), \{, \}, [,]$

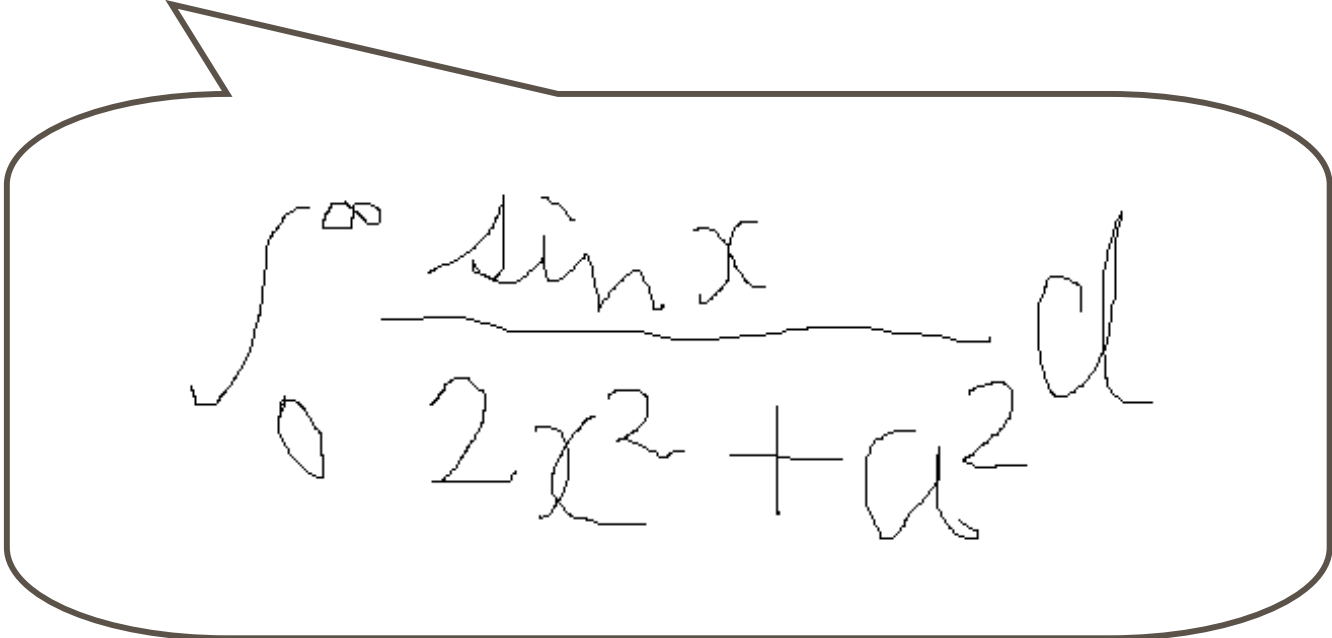


4. Structure Analysis

- Handwritten characters and math expressions input using pen interface have generally severe *distortion*.

4. Structure Analysis

- Handwritten characters and math expressions input using pen interface have generally severe *distortion*.



A speech bubble containing a handwritten mathematical expression. The expression is an integral from 0 to infinity of sin x over (2x^2 + a^2) dx. The handwriting is somewhat messy and distorted, particularly in the denominator and the differential dx.

$$\int_0^{\infty} \frac{\sin x}{2x^2 + a^2} dx$$



4. Structure Analysis

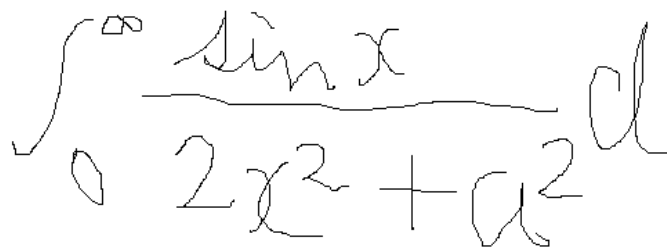
- Handwritten characters and math expressions input using pen interface have generally severe *distortion*.
- An *error of link structure* often destroys the structure analysis of the remaining part.

$$\int_0^{\infty} \frac{\sin x}{2x^2 + a^2} dx$$

is

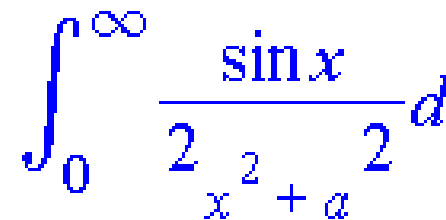
$$\int_0^{\infty} \frac{\sin x}{2x^2 + a^2} dx$$

- Handwritten character using pen interface has distortion.
- An error of link structure often destroys the structure analysis of the remaining part.



$$\int_0^{\infty} \frac{\sin x}{2x^2 + a^2} dx$$

is



$$\int_0^{\infty} \frac{\sin x}{2x^2 + a^2} dx$$

- Handwritten characters using pen interface have distortion.
- An error of link structure often destroys the structure analysis of the remaining part.
- The labors for the *correction* of this kind of errors disturb seriously the smooth input of mathematical expressions



4. Structure Analysis

- *Automatic Rewriting Method* is introduced to overcome this difficulty.



4. Structure Analysis

- *Automatic Rewriting Method* is introduced to overcome this difficulty.
- In our system, each written character is rewritten by *neat* strokes in an appropriate *size* and *position*, immediately and automatically.

4.1. Determination of Characters

Extendable Character:

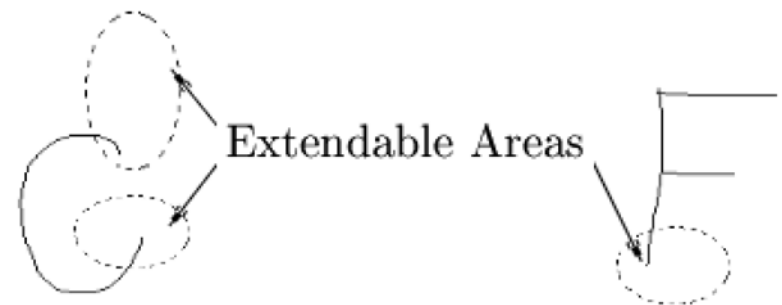
$F \rightarrow E$, $C \rightarrow G$ or d , $= \rightarrow \neq$, etc.

Unextendable Character:

A , B , E , etc.

Extendable Area:

The area in which the next stroke is supposed to be pushed down when it is extended to other character by adding some strokes.





4.1. Determination of Characters

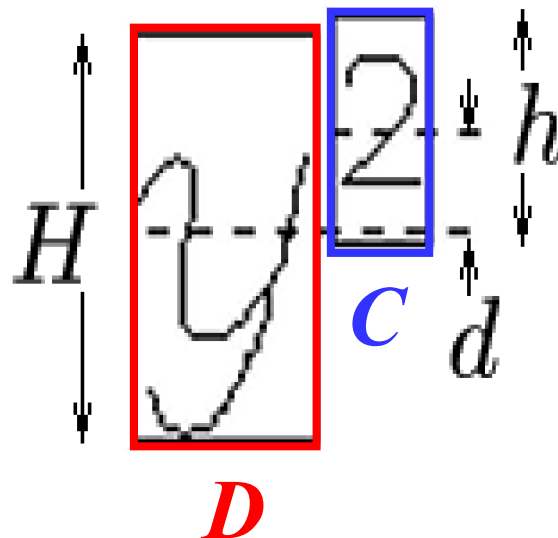
Each time a stroke is written, the character recognition is performed by the system.

The written strokes are rewritten neatly by computer in the following cases:

1. The recognition cost is low (good) enough and either the recognition result is an unextendable character, or the next stroke is pushed down outside of its extendable area.
2. An interval of one second has elapsed with no input of new stroke.

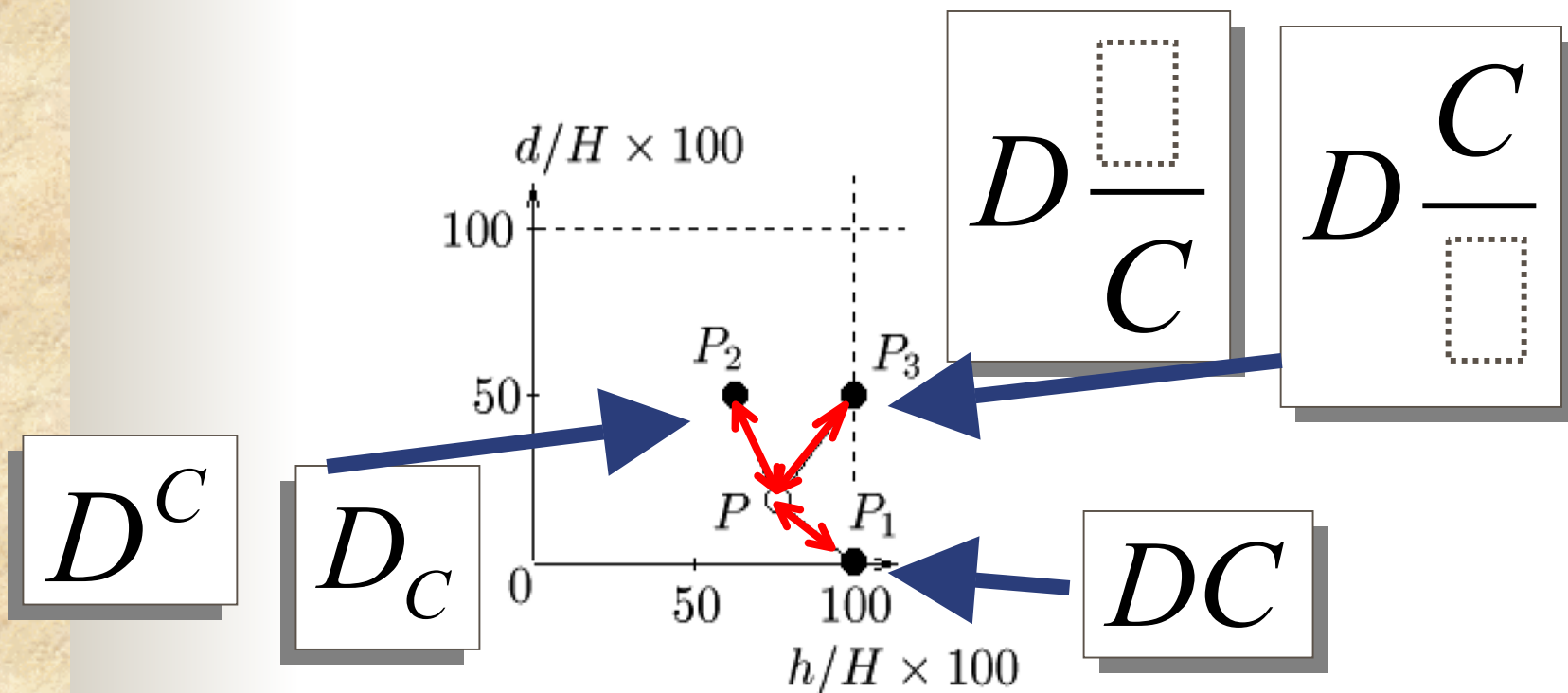
4.2. Determination of Positions and Sizes of Characters

For two written characters C , D ,
the *relation point* $P(C,D)$ is defined by
 $P(C,D):=(h/H \times 100, d/H \times 100)$.



4.2. Determination of Positions and Sizes of Characters

To determine the relation between C and D , select the nearest *ideal relation point*.





5. Adaptation to user

Registration of user's script:

- Demonstration.



6. Further development

- Accuracy
 - Grammer
 - Bigrams
 - Frequently used expressions
- Editing facility by pen.
 - Insert/Delete
 - Select, Cut, Paste



Conclusion

- Brief introduction of InftyProject
- Difficulty of handwritten math recognition
- Sketch of our method:
 - Character recognition
 - Structure analysis (Automatic rewriting)
- Future work



“INFTY”

R&D project for math information processing

Thanks you!

Masakazu Suzuki
Graduate School/ Faculty of Math.
Kyushu University
suzuki@math.kyushu-u.ac.jp
<http://math.kyushu-u.ac.jp/~suzuki>

InftyProject: <http://www.inftyproject.org>