# Recognition System for On-line Sketched Diagrams

Martin Bresler[1], Truyen Van Phan[2], Daniel Průša[1], Masaki Nakagawa[2], Václav Hlaváč[1]

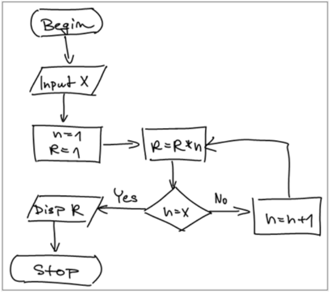
[1]Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University in Prague
166 27, Praha 6, Technická 2, Czech Republic

[2]Department of Computer and Information Science
Tokyo University of Agriculture and Technology
Tokyo 184-8588, Japan

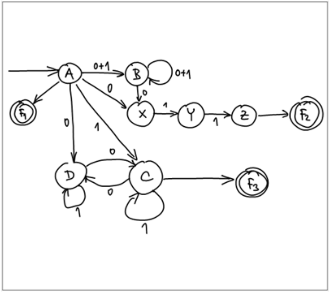
4th September 2014

# Diagram Structure

- Diagrams consisting of:
  1. Symbols with relatively stable appearance (**uniform symbols**) connected by arrows.
  2. Text can label both, the uniform symbols and the arrows.
- Diagram examples:




(a) Flowchart




(b) Finite automata

# Recognition Pipeline

1. Text separation
   - Classify single strokes into two classes: **text** and **shapes**.
   - Ideally remove all text strokes.
   - Practically difficult $\Longrightarrow$ do not remove controversial strokes.

# Recognition Pipeline

1. Text separation
   - Classify single strokes into two classes: **text** and **shapes**.
   - Ideally remove all text strokes.
   - Practically difficult $\Longrightarrow$ do not remove controversial strokes.
2. Estimation of a distance threshold
   - Estimated from shape strokes only. Referenced as **distThresh**.
   - Necessary in the following steps of the pipeline to determine **proximity** of strokes or/and points.

# Recognition Pipeline

1. Text separation
   - Classify single strokes into two classes: **text** and **shapes**.
   - Ideally remove all text strokes.
   - Practically difficult $\Longrightarrow$ do not remove controversial strokes.
2. Estimation of a distance threshold
   - Estimated from shape strokes only. Referenced as **distThresh**.
   - Necessary in the following steps of the pipeline to determine **proximity** of strokes or/and points.
3. Symbol candidates detection
   - Detect symbols with relatively stable appearance first. Referenced as **uniform symbols**.
   - Classification based on appearance of stroke groups obtained by an **oversegmentation**.
   - **Arrows** having varying appearance detected as connectors between two uniform symbols.

# Recognition Pipeline

4. Structural analysis
   - Selection of symbol candidates forming a valid diagram.
   - Each symbol candidate has a score assigned depending on: its appearance, relations with other symbol candidates.
   - Search for a solution with the highest score – optimization task (**max-sum** problem).

# Recognition Pipeline

4. Structural analysis
   - Selection of symbol candidates forming a valid diagram.
   - Each symbol candidate has a score assigned depending on: its appearance, relations with other symbol candidates.
   - Search for a solution with the highest score – optimization task (**max-sum** problem).
5. Text recognition
   - Utilize already known structure of the diagram to define **text blocks** and assign them to symbols.
   - Recognize meaning of the text blocks.

# Text Separation

- Based on Van Phan and Nakagawa ICFHR 2014.
- We used a Bidirectional Long-Short Term Memory (**BLSTM**) Recurrent Neural Network (**RNN**) as a classifier.

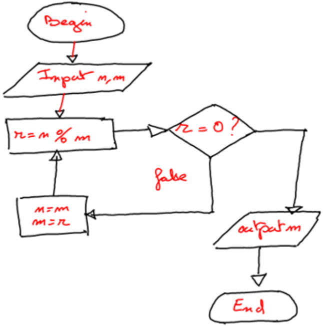
# Text Separation

- Based on Van Phan and Nakagawa ICFHR 2014.
- We used a Bidirectional Long-Short Term Memory (**BLSTM**) Recurrent Neural Network (**RNN**) as a classifier.
- Combination of unary and binary features:
  - Unary features express how appearance of a stroke fits a concrete class.
  - Binary features express how relations with neighbouring strokes support a class transition.

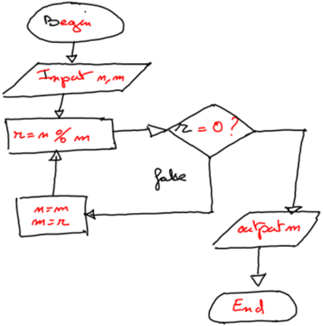
# Text Separation

- Based on Van Phan and Nakagawa ICFHR 2014.
- We used a Bidirectional Long-Short Term Memory (**BLSTM**) Recurrent Neural Network (**RNN**) as a classifier.
- Combination of unary and binary features:
  - Unary features express how appearance of a stroke fits a concrete class.
  - Binary features express how relations with neighbouring strokes support a class transition.
- The precision of 97.8 % achieved:
  - 97.3 % in the shapes class
  - 98.1 % in the text class
- The classifier biased to have smaller error in the shapes class:
  - 99.2 % in the shapes class
  - 89.7 % in the text class

- Example showing the need for the bias:




(a) Unbiased recognition result




(b) Biased recognition result

# Distance Threshold

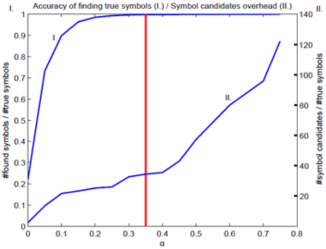
- Diagrams have different sizes, users have different writing styles, and different devices are used for the ink collection.
- **The distance threshold must be extracted from data**.
- We define the threshold as $distThresh = \alpha \cdot Dmed$.
  - $Dmed$ is the median of values determined as lengths of diagonals over bounding boxes of all single shape strokes present in a diagram.
  - $\alpha$ is a coefficient which we empirically chose to be $\alpha = 0.35$.
- Strokes grouping task was used to tune the parameter $\alpha$.

# Distance Threshold – Strokes Grouping

- Oversegmentation is done by grouping of strokes.
- All possible sets of strokes fulfilling the following conditions are created:
  - Strokes in a set are spatially close.
  - Set does not contain more than $n$ (5 for flowcharts) strokes.
  - Set consists of two consecutively drawn parts at most.

# Distance Threshold – Strokes Grouping

- Oversegmentation is done by grouping of strokes.
- All possible sets of strokes fulfilling the following conditions are created:
  - Strokes in a set are spatially close.
  - Set does not contain more than $n$ (5 for flowcharts) strokes.
  - Set consists of two consecutively drawn parts at most.
- Two strokes are spatially close if:
  - The distance between their two closest points $d < distThresh$.
- We tuned the prameter $\alpha$ by searching for a value, where the strokes grouping algorithm finds the most true symbols (biggest recall).

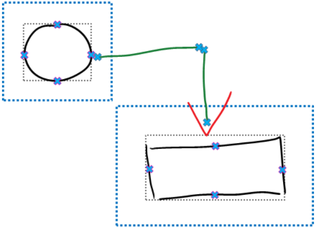



(b) Estimation of $\alpha$

# Uniform Symbols Detection

- **SVM** classifier based on the trajectory-based normalization and direction features by Liu and Zhou 2006 [1].
- Classifying groups of spatially and temporarily close strokes.
- Trained with negative examples to have **rejection ability**.
- Samples of one class **clustered** into **sub-classes** based on the descriptor.
- Logistic regression for **posterior probability** (score).
- Top-3 results, recall 92.0 %, precision 42.0 %.

- Arrows detected between pairs of symbol candidates:
  1. Find an arrow **shaft** as a sequence of strokes leading from the first symbol to the second symbol.
  2. Find an arrow **head** determining orientation of the arrow.
  3. Compute a score of the found arrow: $score = \exp\left(\ln(0.5) \cdot \frac{distSum}{distThresh}\right)$, where *distSum* is a sum of the distances between connector's endpoints and corresponding connection points of symbols and distances between consecutive strokes of the connector.



(a) An arrow example

# Structural Analysis

- Each symbol candidate has its own score.
- Symbol candidates might be in a relation with other symbol candidates, each relation has its own score:
  1. Conflict – symbol candidates share stroke(s). $score = -\infty$
  2. Overlap – bounding boxes of symbol candidates overlaps.
     $score = -\frac{S_{A \cap B}}{\min(S_A, S_B)}$, where
     $A, B \ldots$ bounding boxes of the first and the second symbol
     $S_A, S_B, S_{A \cap B} \ldots$ area of $A, B$, and their intersection

# Structural Analysis – Max-Sum

- The pairwise max-sum labeling problem is formulated as:

$$\max_{k \in K^V} \Big[ \sum_{u \in V} g_u(k_u) + \sum_{\{u,v\} \in E} g_{uv}(k_u, k_v) \Big],$$

where an undirected graph $G = (V, E)$, a finite set $K$, and numbers $g_u(k_u), g_{uv}(k_u, k_v) \in \mathbb{R} \cup \{-\infty\}$ are given.

# Structural Analysis – Max-Sum

- The pairwise max-sum labeling problem is formulated as:

$$\max_{k \in K^V} \Big[ \sum_{u \in V} g_u(k_u) + \sum_{\{u,v\} \in E} g_{uv}(k_u, k_v) \Big],$$

where an undirected graph $G = (V, E)$, a finite set $K$, and numbers $g_u(k_u), g_{uv}(k_u, k_v) \in \mathbb{R} \cup \{-\infty\}$ are given.

- Our model:

$V$ – symbol candidates

$E$ – pairs of interacting nodes

$K = \{0, 1\}$ – labels indicating presence in the solution

$g_u(k_u)$ – score of symbol candidates

$g_{uv}(k_u, k_v)$ – score of relations

$g_u(0) = 0, g_u(1) = s$ for each node $u$ with score $s$

$g_{uv}(1, 1) = -\infty$ if $u$ and $v$ are in conflict or they are both arrows connected to the same connection points

$g_{uv}(0, 1) = -\infty$ if $u$ is a symbol and $v$ its arrow

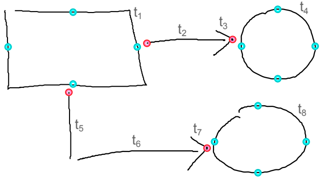
$g_{uv}(1, 1) = s_2$ if $u$ and $v$ are two overlapping symbols

$g_{uv}(k, \ell) = 0$ for all other cases

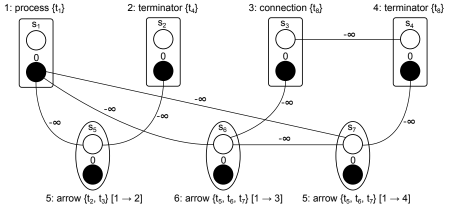
# Structural Analysis – Example

- Suppose that the following symbol candidates were detected in the example below:

  1: process $\{t_1\}$ – $s_1$
  2: connection $\{t_4\}$ – $s_2$
  3: connection $\{t_8\}$ – $s_3$
  4: terminator $\{t_8\}$ – $s_4$
  5: arrow $\{t_2, t_3\}$ $[1 \to 2]$ – $s_5$
  6: arrow $\{t_5, t_6, t_7\}$ $[1 \to 3]$ – $s_6$
  7: arrow $\{t_5, t_6, t_7\}$ $[1 \to 4]$ – $s_7$



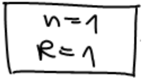

(a) Diagram example

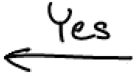



(b) Corresponding max-sum model

# Text Recognition

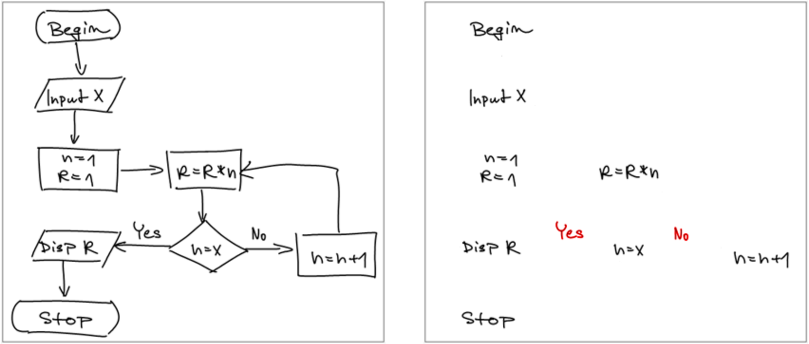
- All **unused strokes** are considered.
- Text blocks are formed with the **knowledge of the diagram structure**.
- Text blocks are recognized by MS Text Recognizer.




Two possibilities: text inside a symbol and text labeling an arrow

# Text Recognition




Demonstration that the text block are salient objects

# Experiments

- Experiments done on two databases – flowcharts (FC) by Awal et al. 2011 [2] and finite autamata (FA).
- Comparison with the state-of-the-art by Carton et al. 2013 [3].

| Class | Correct stroke labeling [%] | | Correct symbol segmentation and recognition [%] | |
|---|---|---|---|---|
| | Carton et al. | Ours | Carton et al. | Ours |
| Arrow | 83.8 | 85.3 | 70.2 | 74.4 |
| Connection | 80.3 | 93.3 | 82.4 | 93.6 |
| Data | 84.3 | 95.6 | 80.5 | 88.8 |
| Decision | 90.9 | 90.8 | 80.6 | 74.1 |
| Process | 90.4 | 93.7 | 85.2 | 87.2 |
| Terminator | 69.8 | 89.7 | 72.4 | 88.1 |
| Text | 97.2 | 99.0 | 74.1 | 87.9 |
| **Total** | **92.4** | **95.2** | **75.0** | **82.8** |

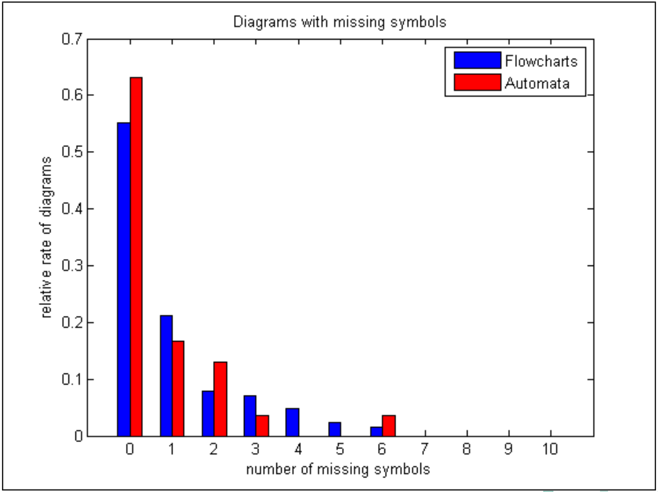
Recognition results for the FC domain.

| Class | Symbols by structure | Symbols by strokes | Labeling |
|---|---|---|---|
| Arrow | 91.2 | 84.4 | 89.3 |
| Arrow in | 84.3 | 80.0 | 78.5 |
| Final state | 95.3 | 93.8 | 96.1 |
| State | 98.7 | 94.5 | 95.2 |
| Label | 96.5 | 96.0 | 99.1 |
| **Total** | **94.6** | **91.5** | **94.5** |

Recognition results for the FA domain.

# Experiments

- Histogram showing how many diagrams were recognized with specific number of errors:



# Performance

- Implemented in C# and tested on a standard tablet PC Lenovo X230 (Intel Core i5 2.6 GHz, 8GB RAM) with 64-bit Windows 7 operating system.
- We are able to recognize a diagram of the average size in less than 1.5 seconds (1.39s).
- It makes our system faster than the system proposed by Carton et al. with average recognition time 1.94s.

| | minimal | maximal | average | median |
|---|---|---|---|---|
| optimization | 0.02 / 0.04 | 3.98 / 4.93 | 0.59 / 0.56 | 0.51 / 0.51 |
| whole recognition | 0.23 / 0.25 | 8.83 / 15.86 | 1.39 / 2.37 | 1.02 / 1.73 |

Running time in seconds for FC / FA databases.

# Summary

Thank you for your attention.

References:


[1] C.-L. Liu and X.-D. Zhou: *Online Japanese Character Recognition Using Trajectory-Based Normalization and Direction Feature Extraction*, IWFHR 2006.

[2] A.-M. Awal, G. Feng, H. Mouchére, and C. Viard-Gaudin: *First Experiments on a New Online Handwritten Flowchart Database*, DRR 2011.

[3] C. Carton, A. Lemaitre, and B. Couasnon: *Fusion of Statistical and Structural Information for Flowchart Recognition*, ICDAR 2013.

# Demo

(Loading Video...)