Overview of my talk

1. Review of the last talk
2. Actual approach
3. Results
4. Future work

1. Review of the last talk

My diploma work tries to replace the standard input device for the online recognition of hand-written text – the graphics tablet.

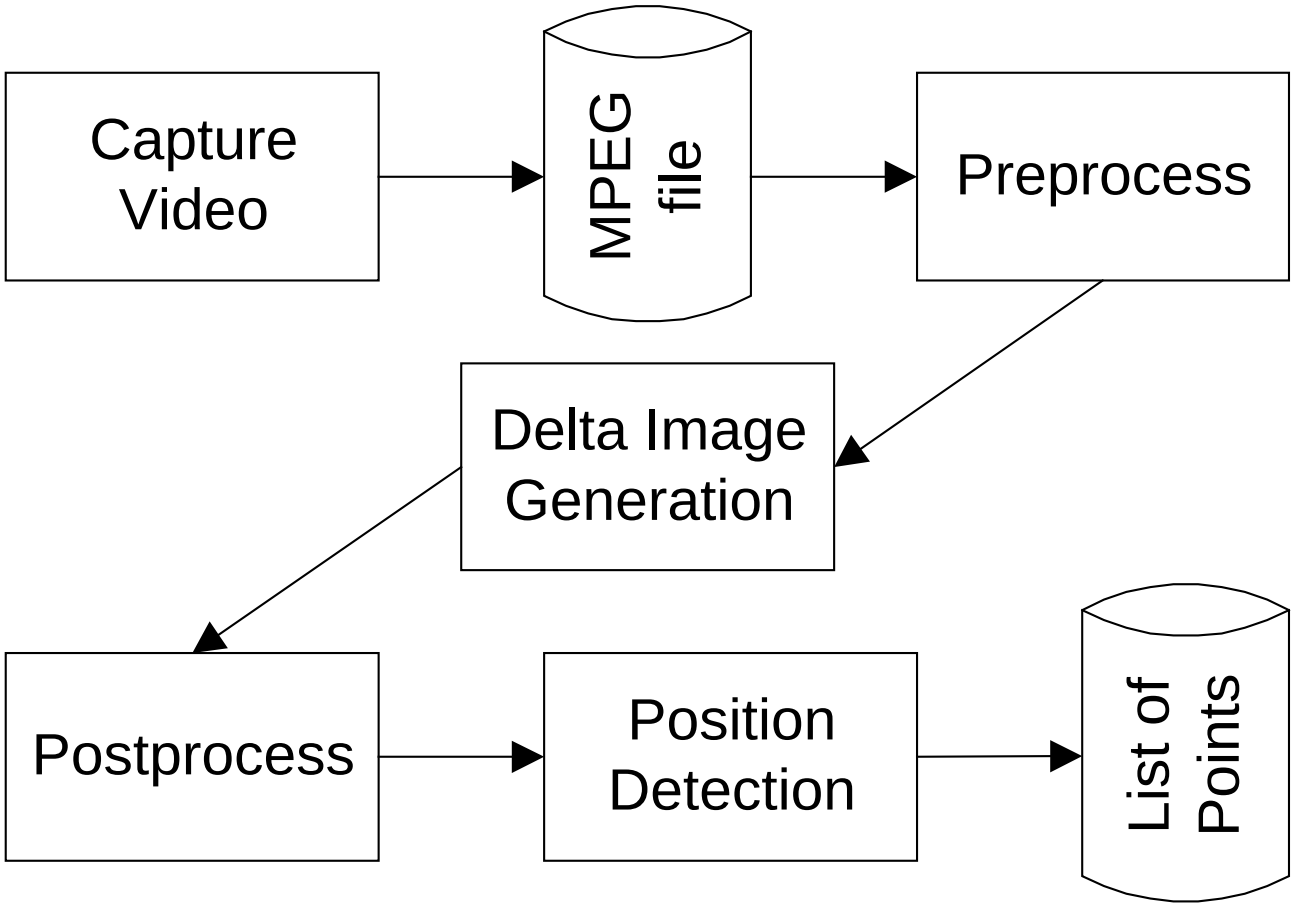
The user writes with a common pen on a sheet of paper. Doing this he's filmed by a video camera in front of him.

The image sequence coming from the camera is converted to connected pen positions over time.

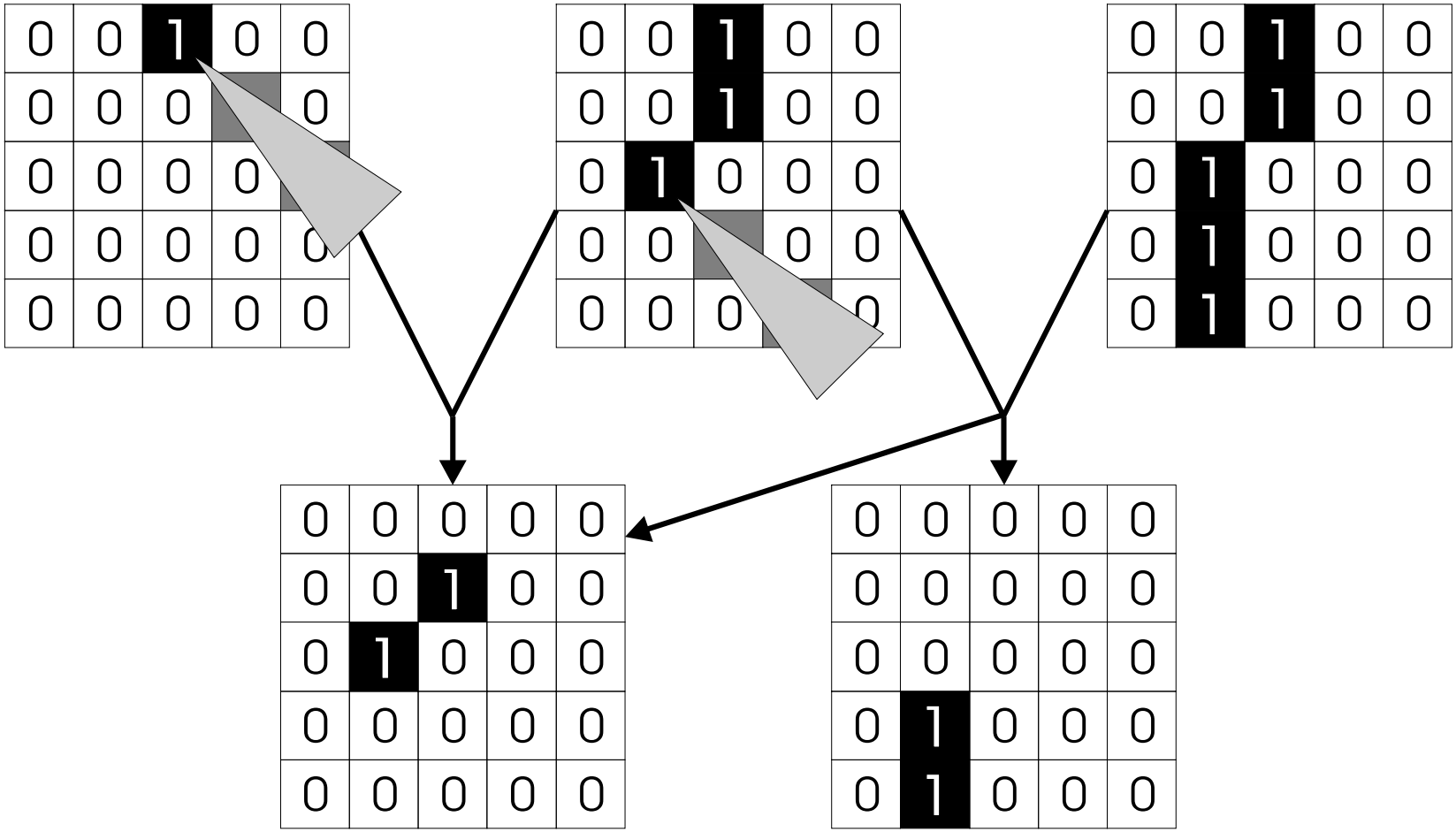
The following restriction apply:

- White pen with dark ink (good contrast)
- No movement of the bright white paper
- Letters are about 1.5 cm high
- General bright illumination of the room

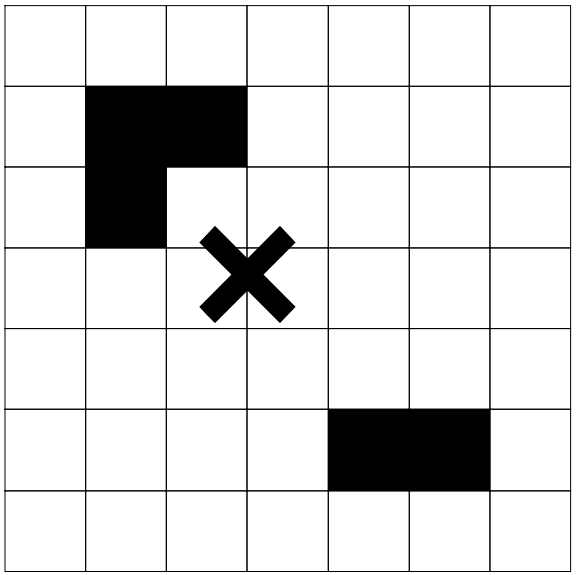
High-level Data Flow of the application



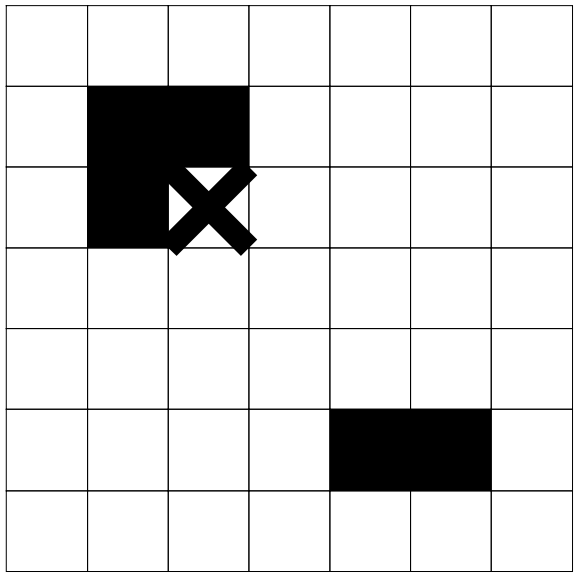
Corrected generation of the delta images



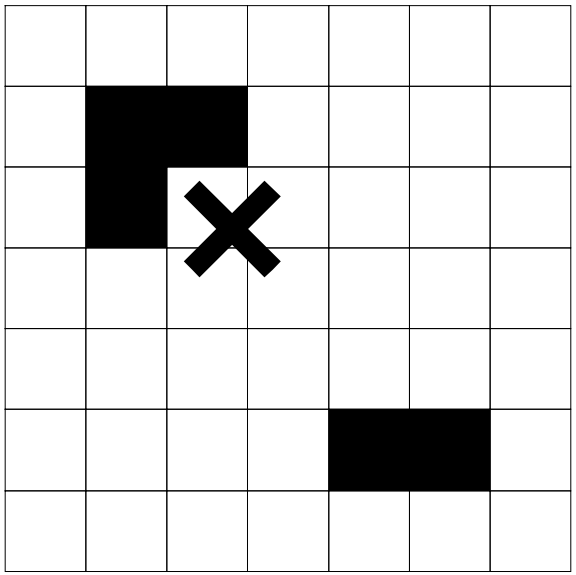
Detection of the pen position out of the delta



Simple



Median



Iterative

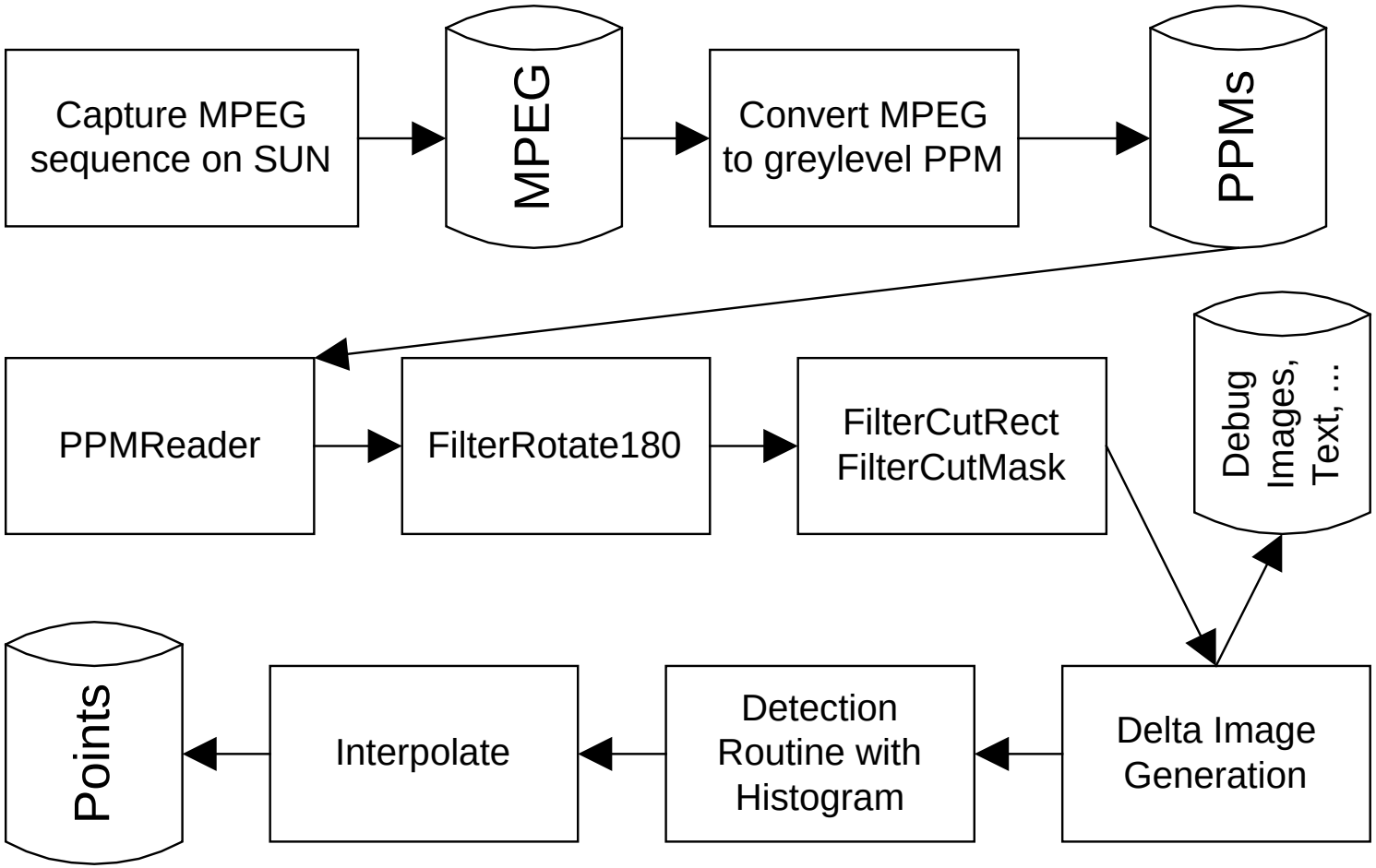
Problems of the approach from June 1997

- Small MPEG-framerate (only ~10 fps)
- Decision, whether this delta image should result in a position or not, was unstable
- Detection methods generate new pixels
- Huge Memory Leaks
- Slow processing speed

2. Actual approach

- Whole process works on greylevel images
- Written line of text generates a mask
- Reasoning about Otsu based on Histogram
- Neighbour Detection method
- Interpolation of detected pixels
- Processing Tool with simple Load-Sharing

Actual, detailed processing pipeline



Generation of the mask out of the line

Original:

B&W:

Expanded:

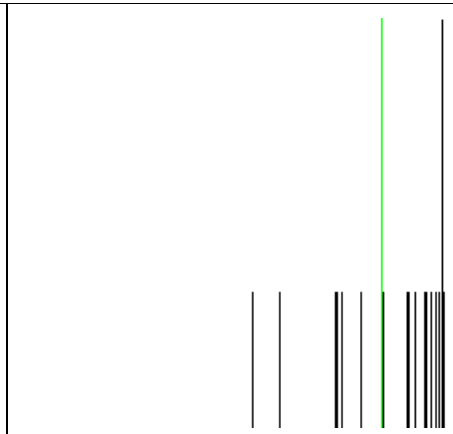
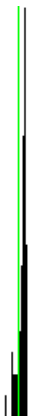
Masked:

Histogram routines

Otsu calculates threshold of delta image out of the Histogram. This threshold will be used to binarize the delta image for the detection.

The variance of Otsu and/or the bandwidth of the Histogram are used to reason about the sense of the value returned by Otsu.

Histogram, Image and result examples Otsu

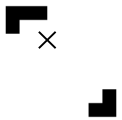
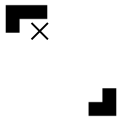
Threshold	214		
Variance	1260		
Width	120		
Threshold	249		
Variance	8		
Width	13		

As a new detection routine I use a criteria based on the sum of distances of a pixel from all the others.

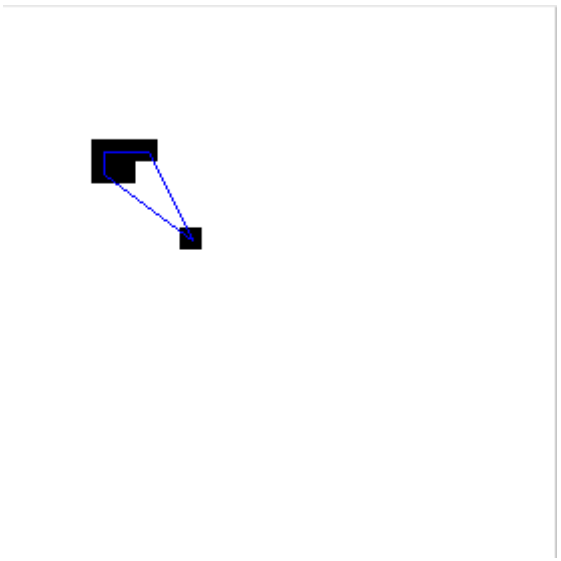
The pixel with the smallest sum is considered to be in the "centre" of all the pixels.

The resulting position is a already detected pixel and not a new one outside the mask!

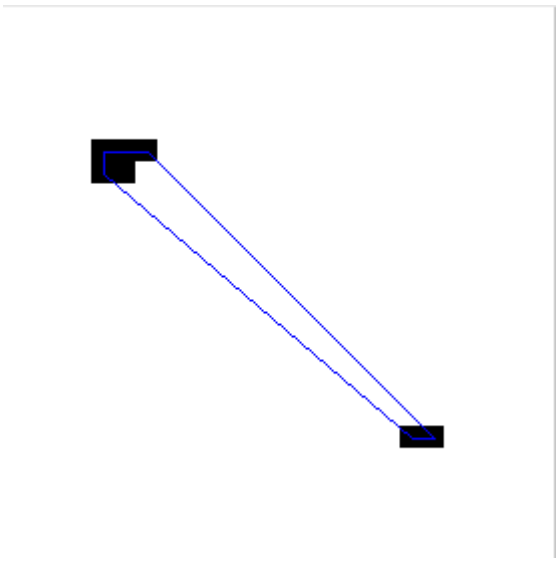
Comparison of different detection routines

Iterative				
Median				
Neighbour				

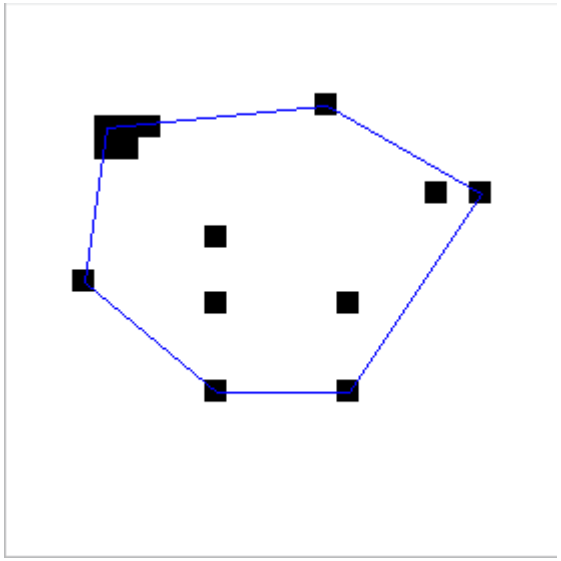
If the spanned area of the black pixels is too large compared to the detection window this result is ignored ...



6 / 576 = 1 %



27 / 576 = 5 %



169 / 576 = 29%

The resulting positions are connected to line strokes. This result I try to improve with the usage of interpolation routines.

Small gaps between two position are filled with linear interpolated positions.

A connection between 2 position which is too long is cut away.

Tools used to work with the data:

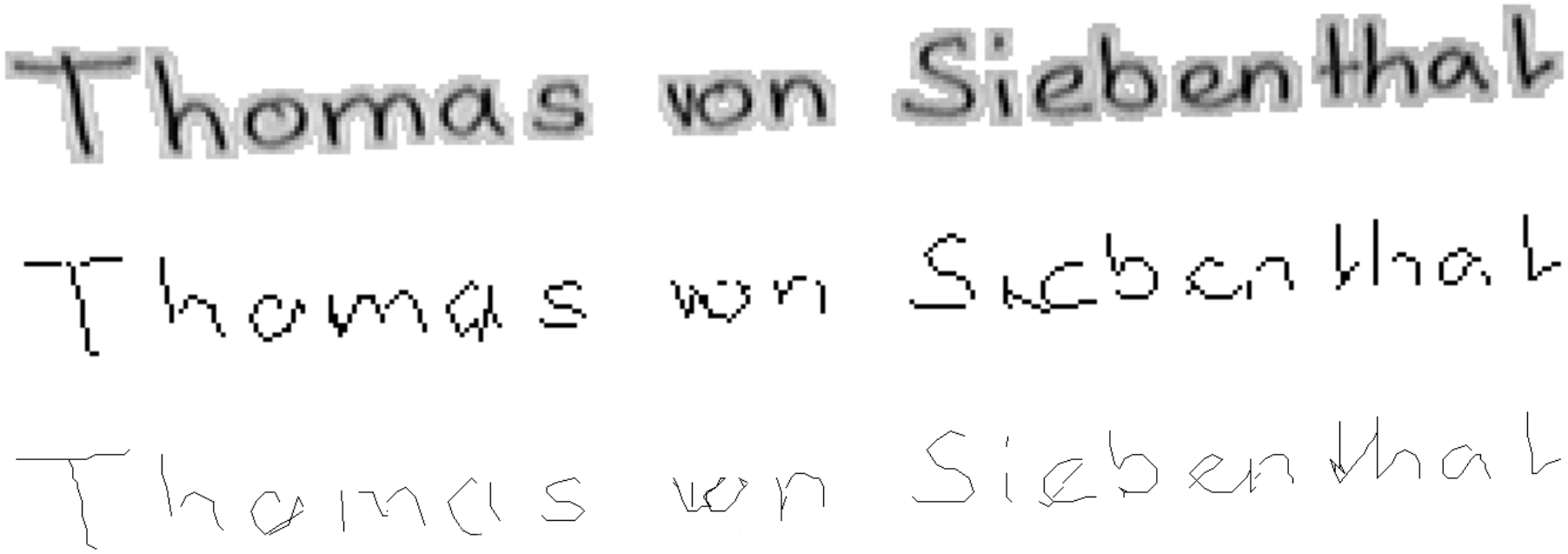
- Interactive GUI with full control
- Command line utility (full speed, no GUI)
- PERL-Script to calculate a lot of parameter combinations on different hosts
- GUI to test the resulting linestrokes
- Debug output is presented in HTML

3. Results

I have 3 different test data sets:

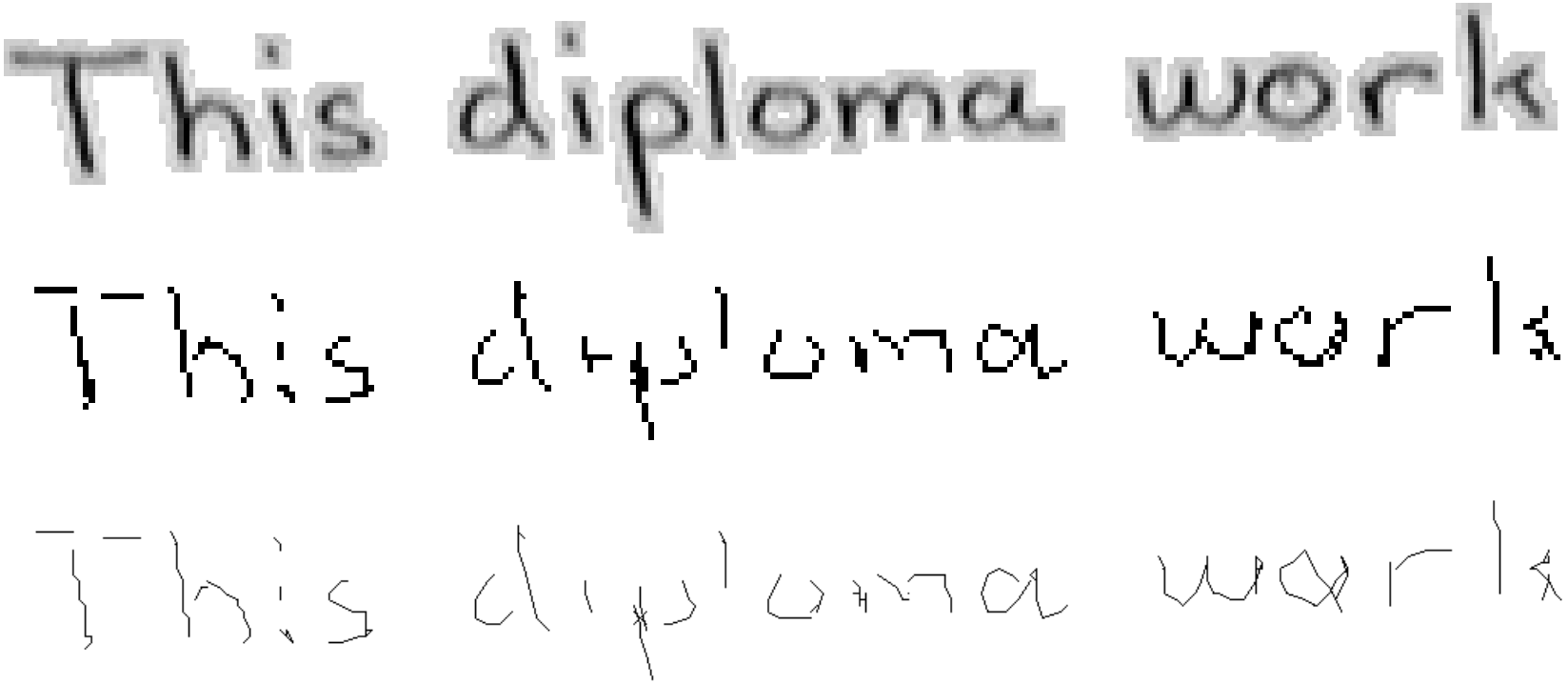
- old sequences with either 2 hard shadows around the pen or a small framerate
- 10 words of German text, usable
- 7 small sentences, with totally ~50 English words, written by 2 writers (~ 100 words)

Example #1 of older test data



Example #2 of new test data

Example #3 of new test data

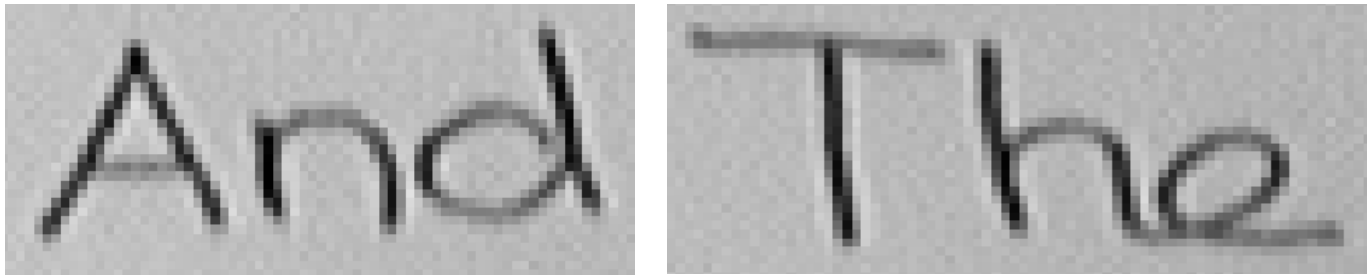


Actual problems:

- Horizontal lines too bright
- Wrong detected upstrokes in "e"
- Wrong detected "double-strokes" in "m"
- Parameter tuning
- Points which are runaways
- Amount of data

Horizontal lines too bright

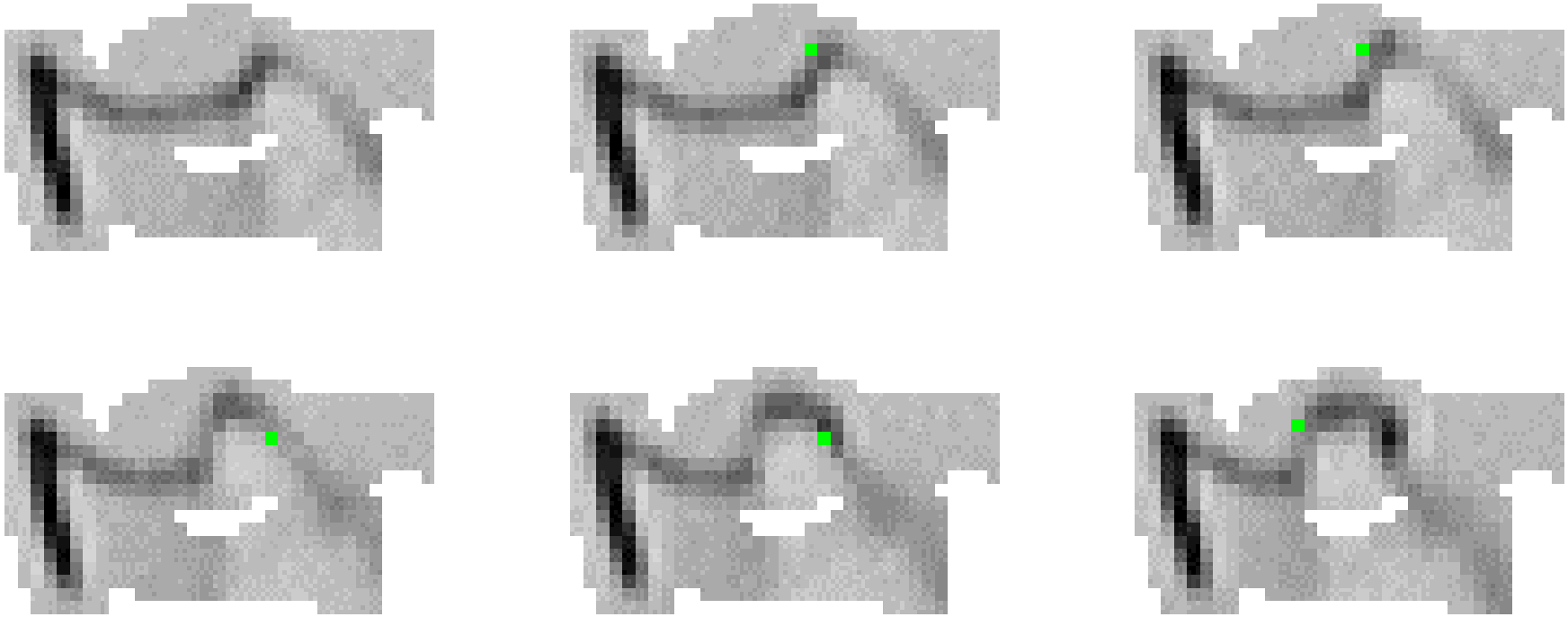
Example:



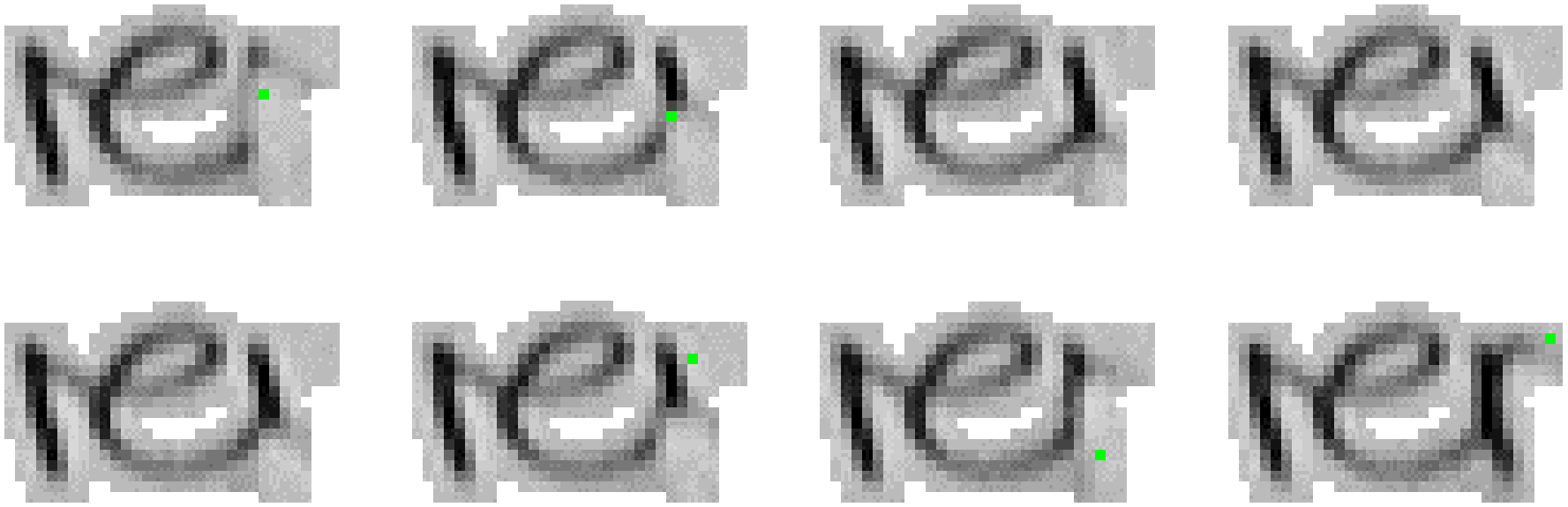
Because the camera records horizontal interlaced images there will be an interpolation between the lines which smoothes the pixels.

Thicker written lines would look better ...

Upstroke in “e” is covered by pen and so the ink appears in the wrong order for the camera.



The first vertical line of the „m“ is written twice, but only detected the first time (downwards)



4. Future work

Test with ETHZ at 29.1.98 (Dr. M. Schenkel)

a) Output of the test data sets is converted for the ETHZ and tested on their DB

b) Input for the ETHZ with the graphics tablet is filmed, converted & recognized. Compare direct and indirect results

-> Try to show usability of my work

Future work (continued)

- Tuning of the stability
- Cleanup of some Memory leaks
- Cleanup of my GUI code
- Documentation
- Diploma exam

Any Questions ?