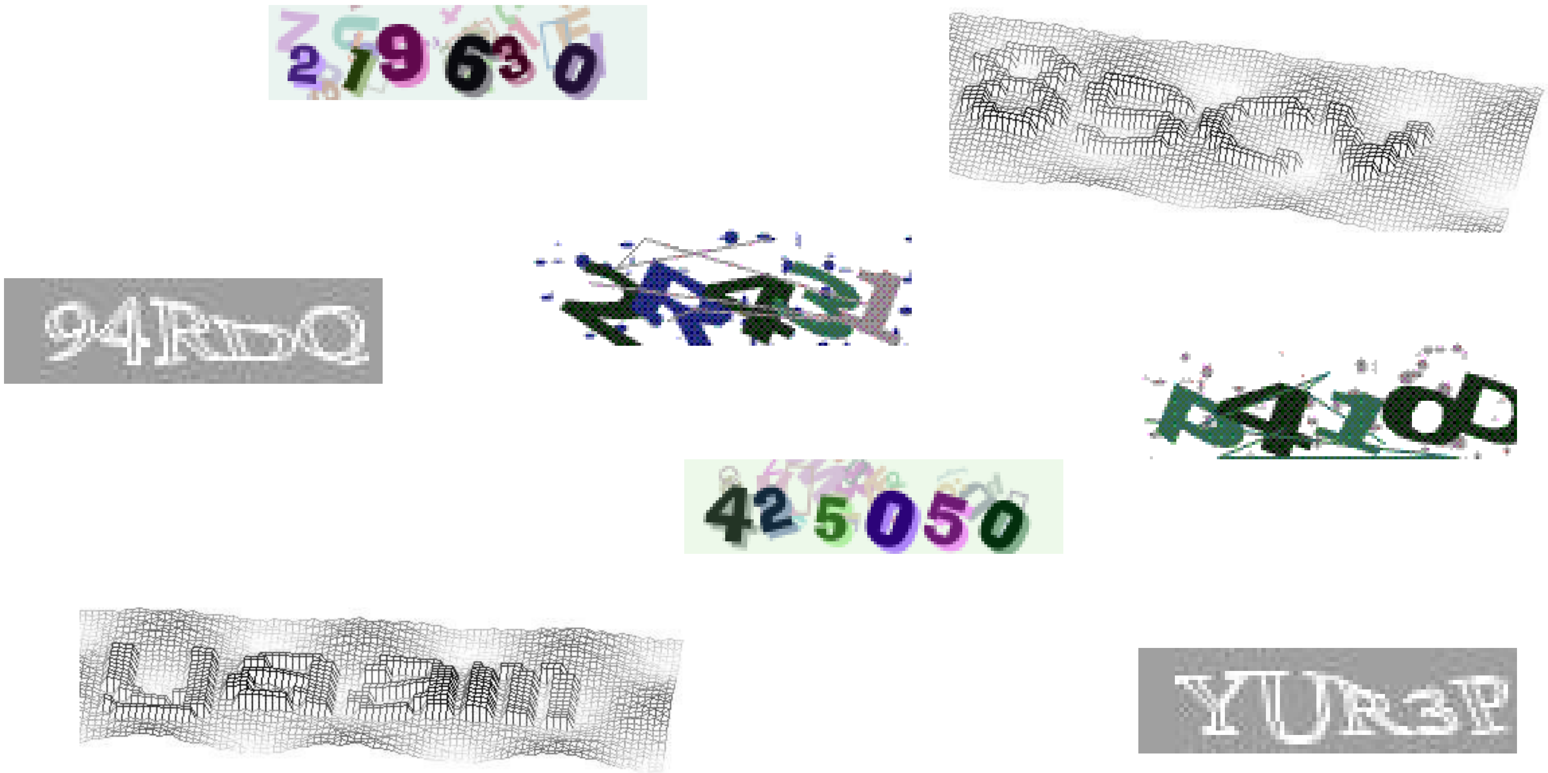
Recognizing some of the modern CAPTCHAs

Dmitry Nikulin

LCME, Saint-Petersburg, 2011

Examples



Stands for

- Completely
- Automated
- Public
- Turing test to tell
- Computers and
- Humans
- Apart

Turing test

- Introduced by the mathematician Alan Turing in 1950
- Aimed to distinguish between a machine and a human
- The classic version is carried out by a human
- Loebner Prize has not been won yet

Reverse Turing Test

- Carried out by a computer
- A widespread example is CAPTCHA
  - Checks for human presence
  - Protects against spam and automated registrations
  - Uses human ability to recognize distorted text (Google reCAPTCHA)

Requirements for a CAPTCHA

- Simple for a human
- Difficult for a machine
- Does not require large computational resources

Let us call a CAPTCHA efficient if a machine can successfully bypass it in no more than 1% of attempts.

Objectives

- Study the efficience of the widespread CAPTCHAs
- CAPTCHAs from the largest Russian mobile network operators web sites were chosen

Reasons of choice

- Operators have enough money to hire a programmer of any qualification
- Operators need to minimize the amount of spam in order to safeguard their reputation

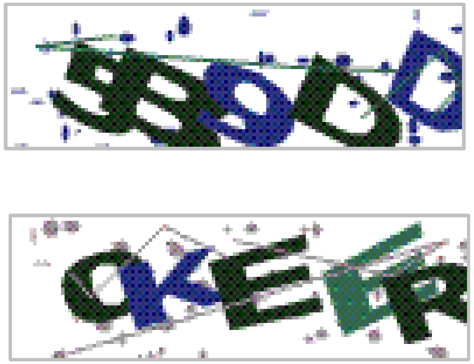
Recognition method overview

- Preprocessing
- Segmentation
- Recognition

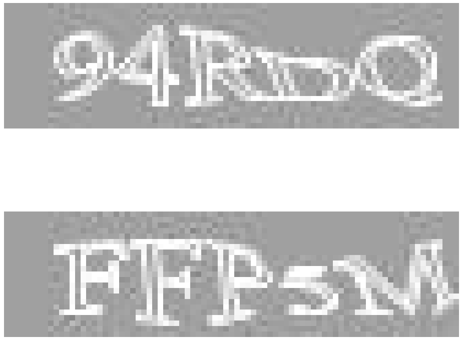
In the following slides details on these stages will be given.

Preprocessing

- Clearing the noise
- Removing distortions



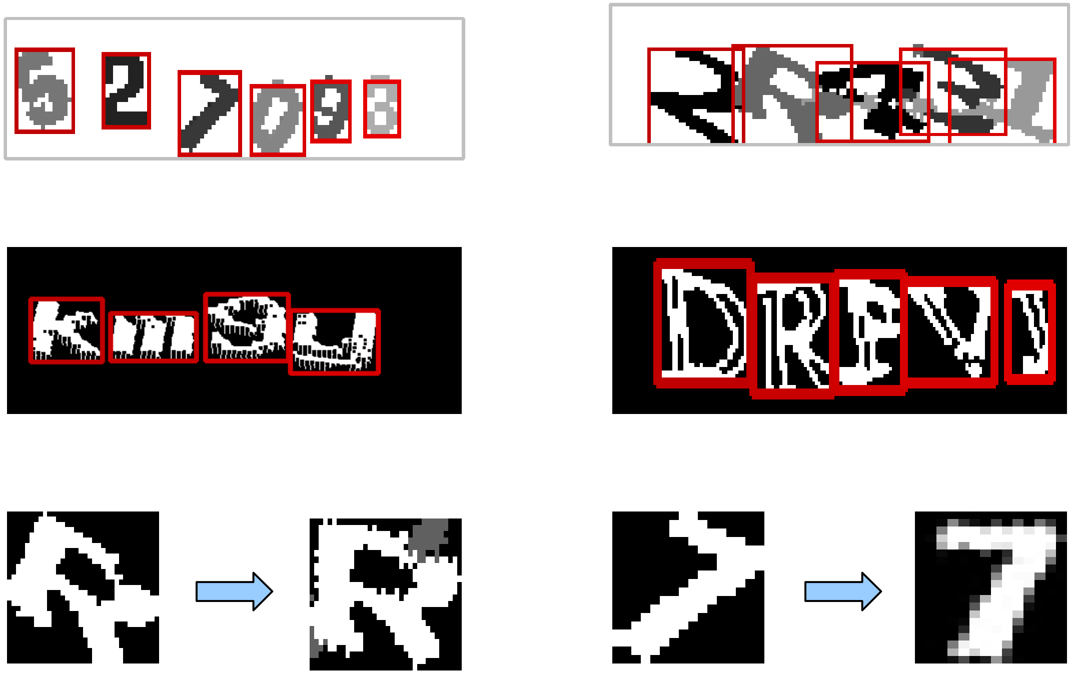
© Beeline



© MTS

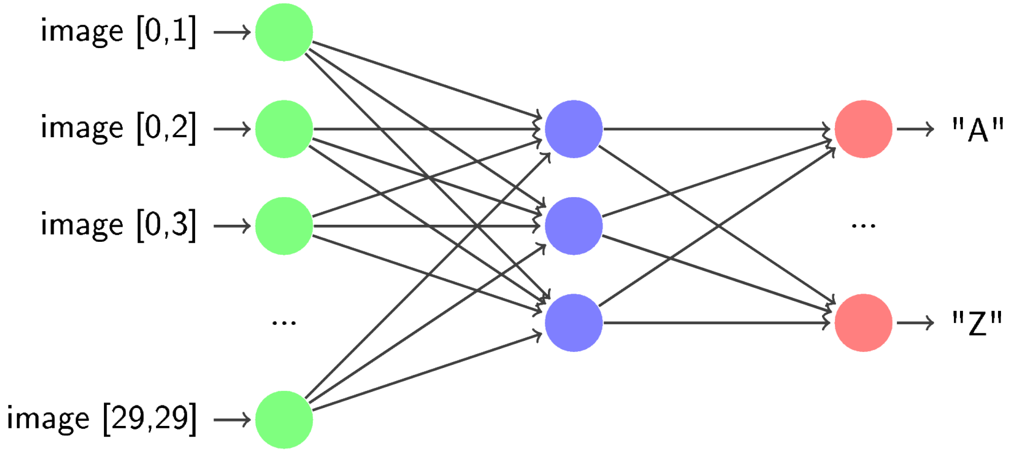
Segmentation

- Extracting characters
- Post-processing characters



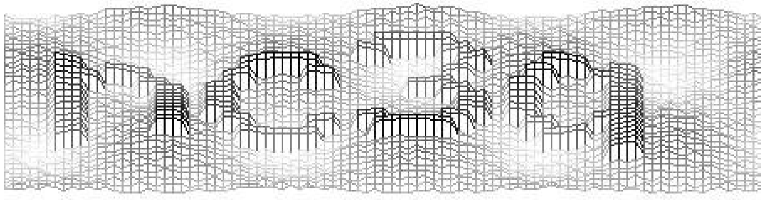
Recognition

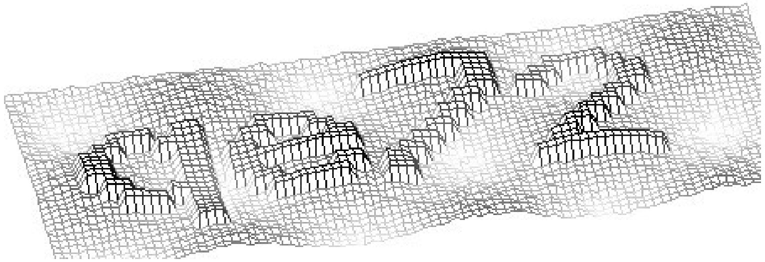
- Classification of characters with a pre-trained neural network

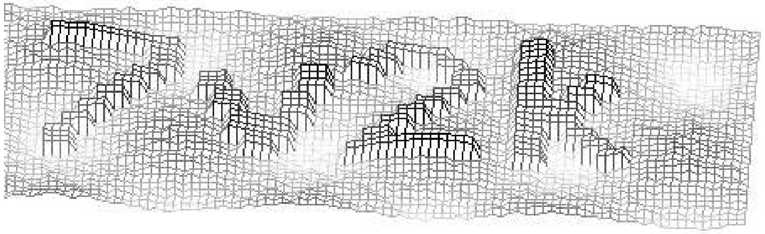


Example

Let us consider the following type of CAPTCHA:







© Megafon

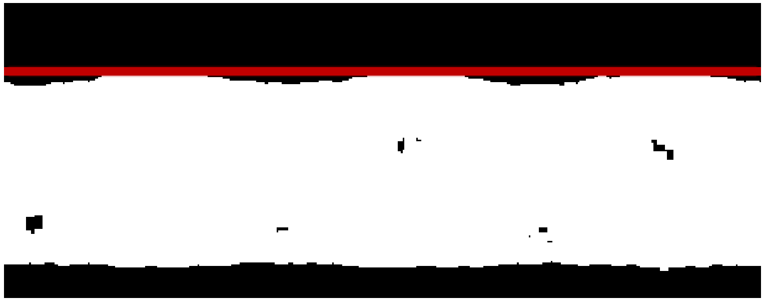
Analyzing the problem

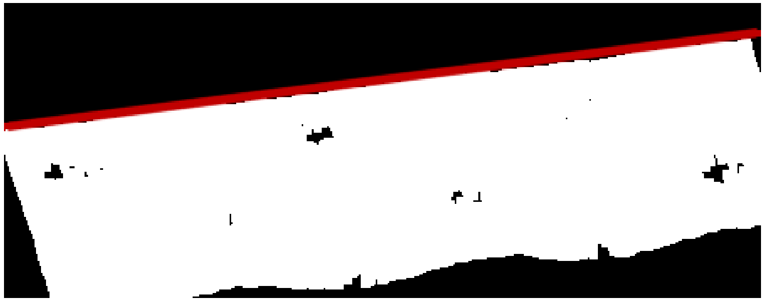
- Characters lie on a 3D wireframe
- The wireframe is rotated and moved
- The brightness is inconsistent
- Seems to be quite bad :(

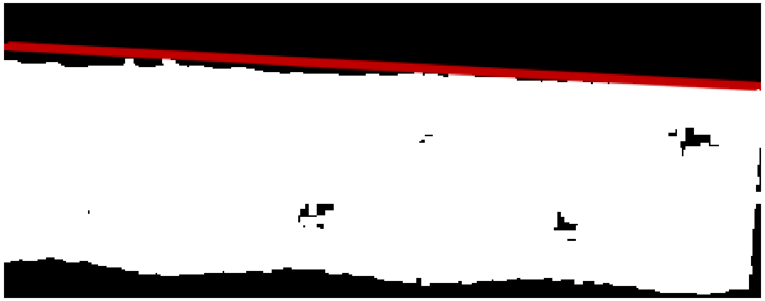
Ideas of the solution

- Ignore the three-dimensionality and use classic methods
- The characters are generally darker than the background and can be separated by brightness
- The upper side of the wireframe is clearly seen – this can be used for the reverse rotation

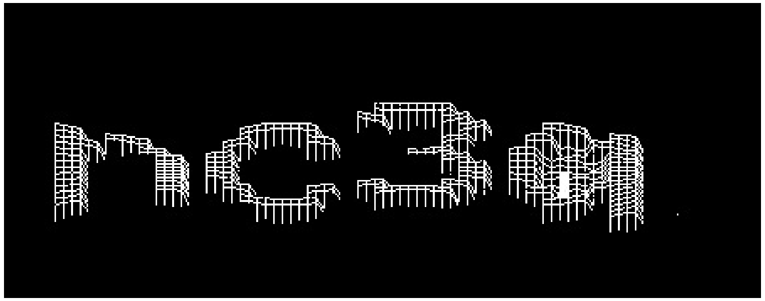
Estimating the rotation angle

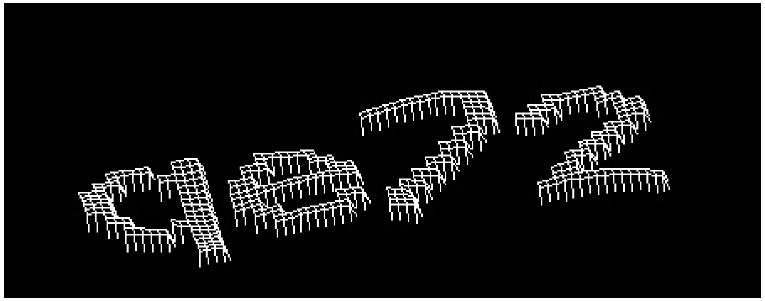


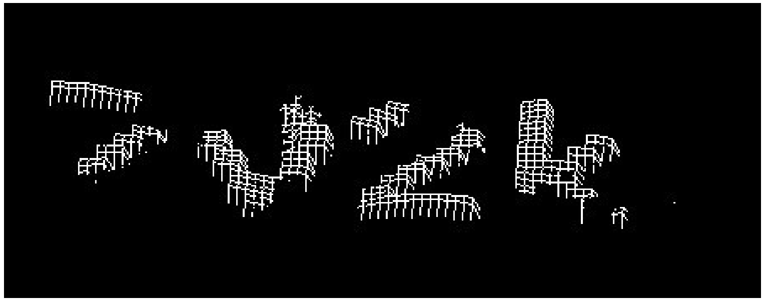




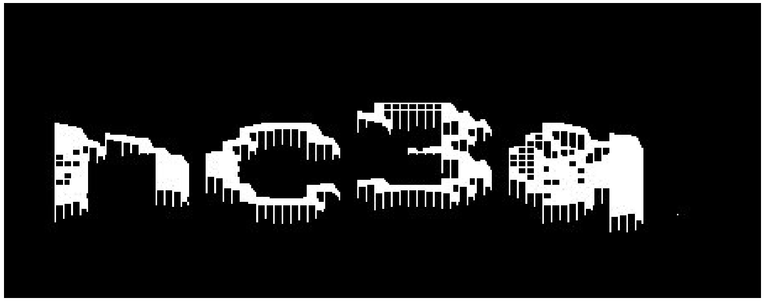
Removing the background

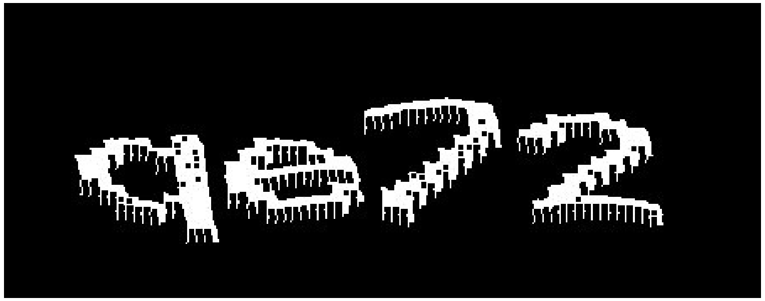


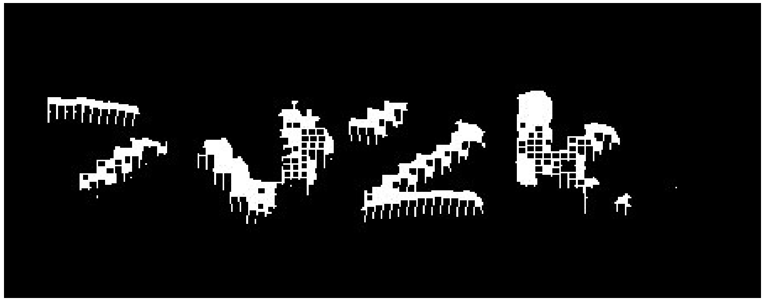




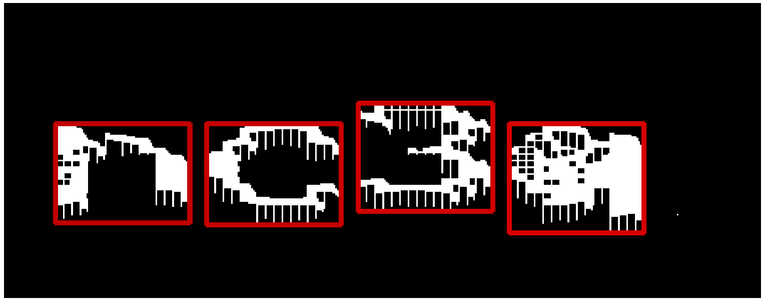
Removing tiny holes

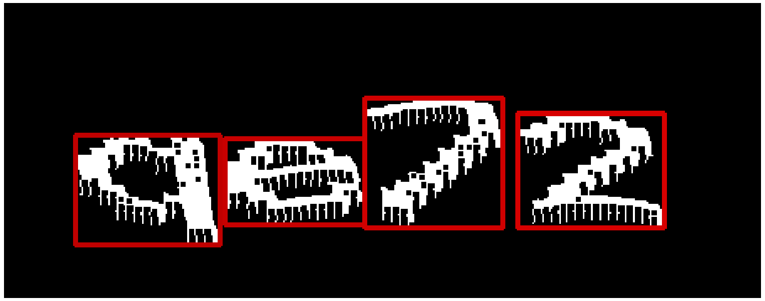


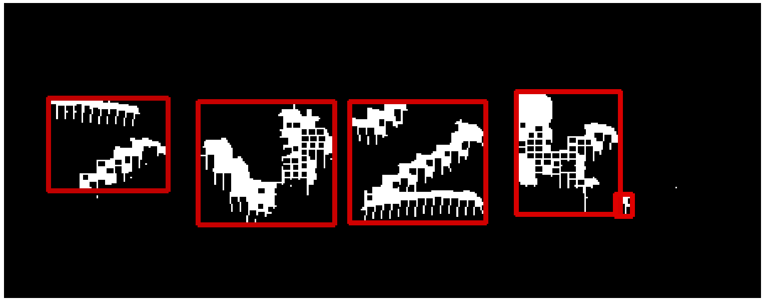




Segmentation







Statistics

- Total number of images – 100
- Recognized successfully – 69
- Recognition error – 31
- Average error – 0.3 characters

Other types of CAPTCHAs

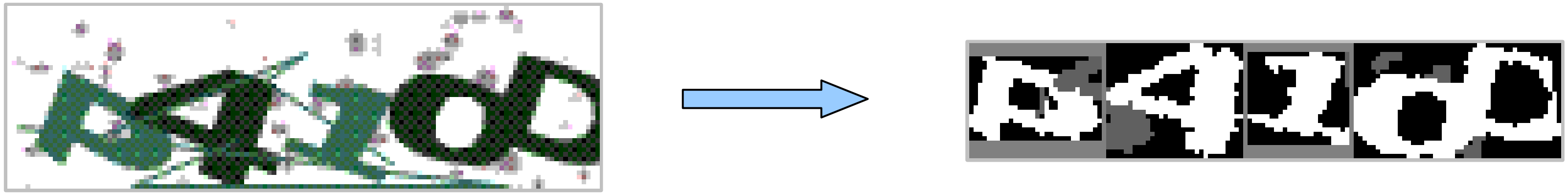
- Preprocessing varies greatly
- Segmentation is quite similar
- Almost identical recognition

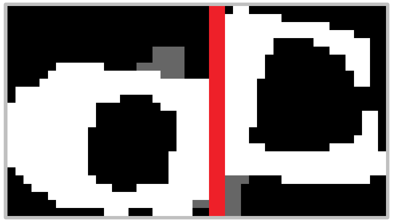
Conclusion — the more transformations are applied to the original image, the more general methods can be used.

Neural network segmentation

- In Beeline's CAPTCHA, the classic method did not show satisfactory results
- A new method which combines the segmentation and recognition was developed

Example





© Beeline

Conclusion

- Only preprocessing varies significantly
- All considered types of CAPTCHAs proved to be inefficient reverse Turing tests

Questions?