# Breaking reCAPTCHA: A Holistic Approach via Shape Recognition

IFIP SEC 2011

Paul Baecher, Niklas Büscher, Marc Fischlin, Benjamin Milde

Darmstadt University of Technology, supported by
DFG Heisenberg and Emmy Noether Programmes

# Introduction

# What Are CAPTCHAs?


- **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part
  - "reverse" Turing test, term coined by [vABHL03]
- challenge/response protocol where
  - response should be easy to observe for humans
  - response should be hard to compute for machines
    - 0.01% according to [CLSC05, vAMM+08]
- application: protect online services from automated use

image: cryptographp

# reCAPTCHA


1st generation

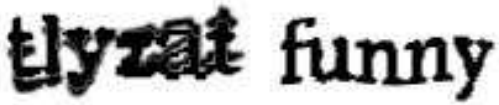

2nd generation


3rd generation


4th generation

- Very popular CAPTCHA service by Google
- may be considered quite “strong”
- unique feature: uses OCR source to generate challenges
  - scan and verification word
- dictionary words. . .

# reCAPTCHA Today

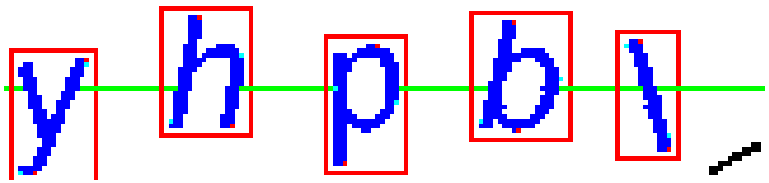



reCAPTCHA as of June 2011
(5th generation)

# Breaking reCAPTCHA

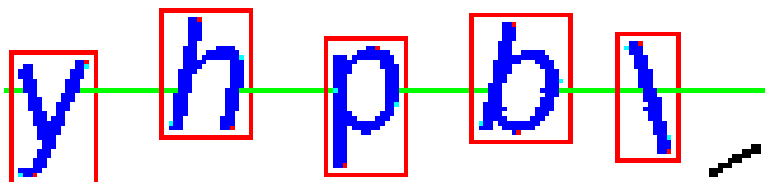
# Breaking reCAPTCHA – Approach

- Typical approach to break text CAPTCHAs
  - segment into individual letters/digits
  - recognize each letter/digit individually



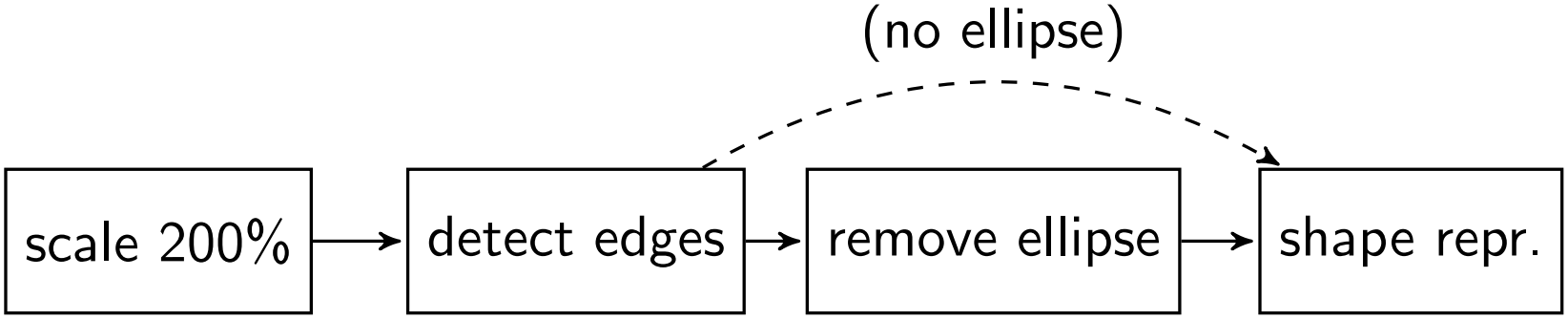
# Breaking reCAPTCHA – Approach

- Typical approach to break text CAPTCHAs
  - segment into individual letters/digits
  - recognize each letter/digit individually



- non-trivial segmentation is considered hard [CLSC05]
- our approach
  - match entire words at once (holistically)
  - i.e. skip segmentation and treat words as letters

# High-level Overview




- Third generation reCAPTCHA challenges add inverted ellipses

# Removing the ellipse

1. **Approximate ellipse center**

original challenge

# Removing the ellipse

1. **Approximate ellipse center**

after erosion operations

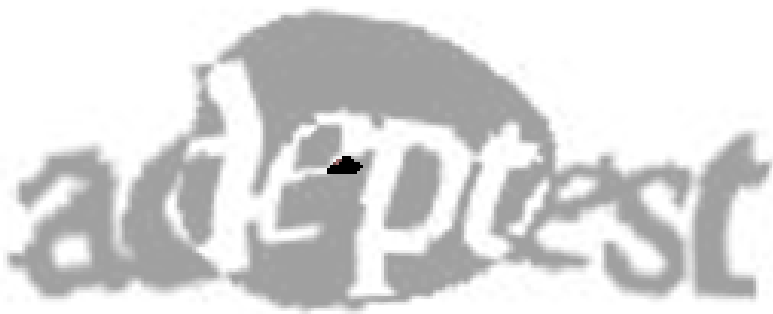
# Removing the ellipse

1. **Approximate ellipse center**

after dilation operations

# Removing the ellipse

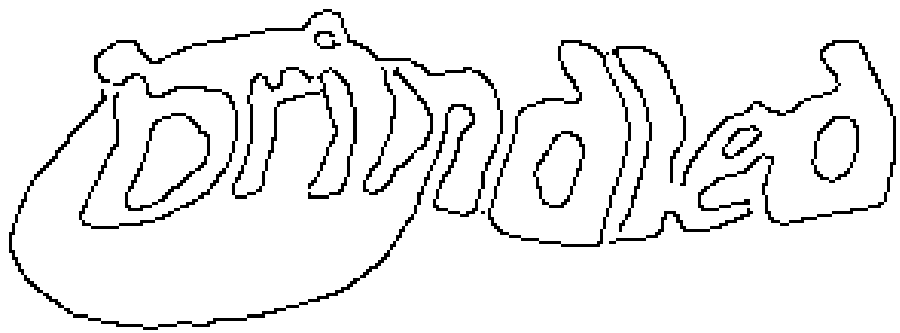
1. **Approximate ellipse center**



center approximated

# Removing the ellipse

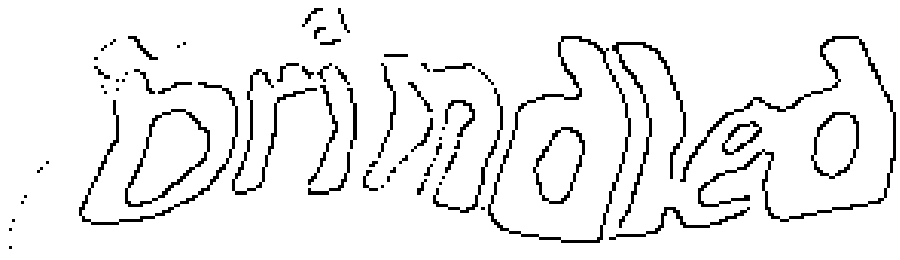
1. Approximate ellipse center
2. **run edge detection on the challenge image**



edge detection

# Removing the ellipse

1. Approximate ellipse center
2. run edge detection on the challenge image
3. **use machine learning to classify contour pixels**



after classification, 1 round

# Removing the ellipse

1. Approximate ellipse center
2. run edge detection on the challenge image
3. **use machine learning to classify contour pixels**

after classification, 4 rounds

# Removing the ellipse

1. Approximate ellipse center
2. run edge detection on the challenge image
3. **use machine learning to classify contour pixels**

after classification, 9 rounds

# Matching Shapes

- Contour line (without ellipse) describes the shape of a word
- reCAPTCHA words are dictionary words
- key idea: prepare a database of all dictionary words and use common shape matching techniques

# Matching Shapes

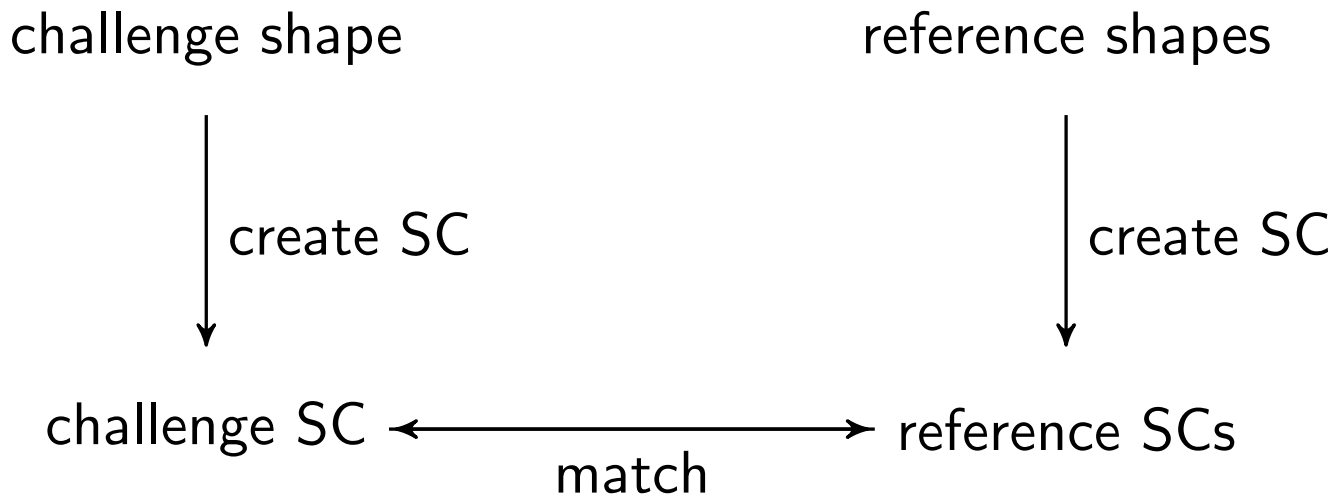
- Contour line (without ellipse) describes the shape of a word
- reCAPTCHA words are dictionary words
- key idea: prepare a database of all dictionary words and use common shape matching techniques

- How to build a database of all dictionary words?
- How to “match” two shapes?

# Shape Recognition

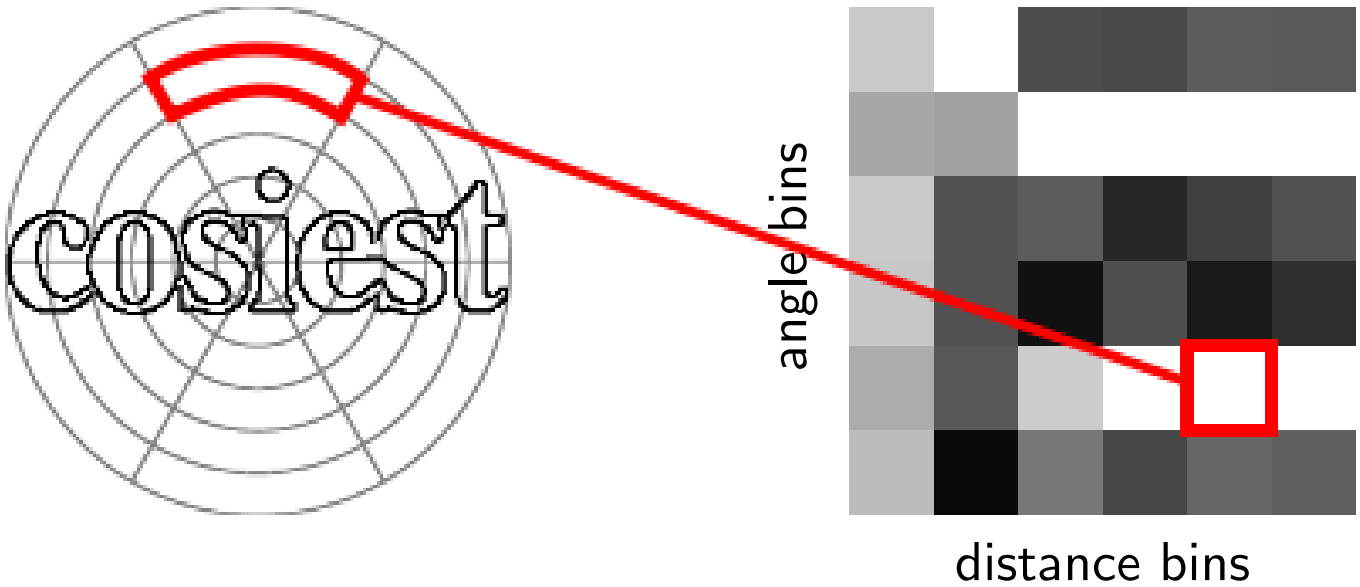
# Shape Recognition

- Well-studied problem in Computer Vision
- powerful technique: Shape Contexts (SC)
- invariant against translation and scaling
- compact description of the shape



# From Shapes to Shape Contexts

- Convert shape (set of points in polar space) into SC (sets of two dimensional histograms)
- example for one point:




- use a $\chi^2$-distance to match sets of histograms

# Matching Shape Contexts Efficiently

- Naive approach is prohibitively slow for 20K dictionary words
- more efficient strategy needed
  - work on a random subset of the sets of points of the shape
  - start with a small subset and double it gradually
  - results in logarithmic search space reduction
- first/last character special treatment
  - easy to detect, allows to prune large chunks

# Experimental Results

# Results

| **reCAPTCHA generation** | **2** | **3** | **4** |
|---|---|---|---|
| Test set size | 496 | 1005 | 301 |
| Total success rate | 12.7% | 5.9% | 11.6% |
| Run time | 24.5s | 17.5s | 15.4s |
| Dictionary success rate | 22% | 10.43% | 23.5% |
| First character detected | 90.2% | 73.2% | 84.6% |

- Recall that a CAPTCHA is considered broken at 0.01%
- performance measurement on verification words only

# The End

Thank you!

?

# References


Kumar Chellapilla, Kevin Larson, Patrice Y. Simard, and Mary Czerwinski.

Building segmentation based human-friendly human interaction proofs (HIPs).
In *HIP*, volume 3517 of *Lecture Notes in Computer Science*, pages 1–26. Springer-Verlag, 2005.

Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford.

CAPTCHA: Using hard AI problems for security.
In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 294–311, Warsaw, Poland, May 4–8, 2003. Springer, Berlin, Germany.

Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum.

reCAPTCHA: Human-based character recognition via web security measures.
*Science*, 321(5895):1465–1468, 2008.