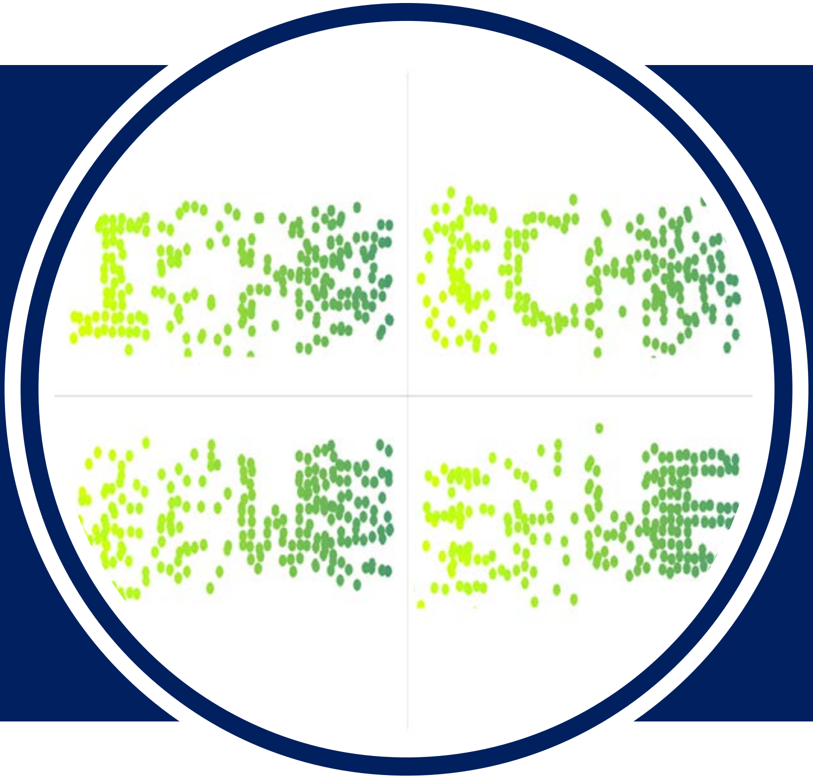
[ICWE 2019 Oral Presentation]



DotCHA: A 3D Text-Based Scatter-Type CAPTCHA

Suzi Kim and Sunghee Choi

Geometric Computing Lab.
School of Computing, KAIST

https://suzikim.github.io/DotCHA/

DotCHA: A 3D Text-Based Scatter-Type CAPTCHA

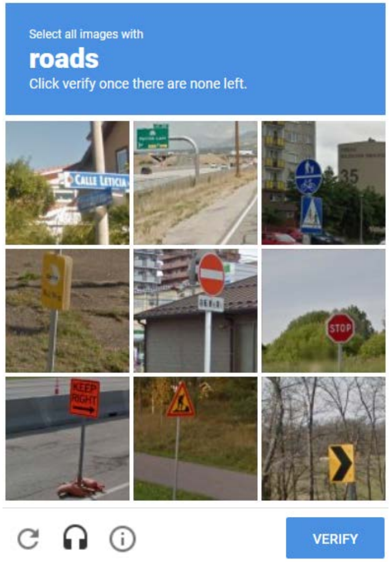
DotCHA: A 3D Text-Based Scatter-Type CAPTCHA

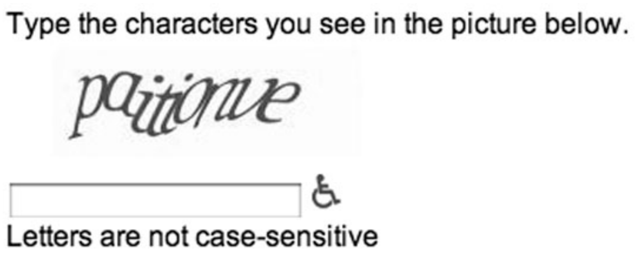
CAPTCHA

***C**ompletely **A**utomated **P**ublic **T**uring tests to tell **C**omputers and **H**umans **A**part*

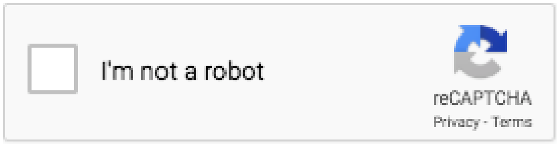
Using tasks that **humans can perform easily** but **machines cannot**

- To discriminate human users from **malicious bots**
- To protect systems from **denial of services (DoS)**


Select all images with
roads
Click verify once there are none left.


Type the characters you see in the picture below.

Letters are not case-sensitive


□ I'm not a robot

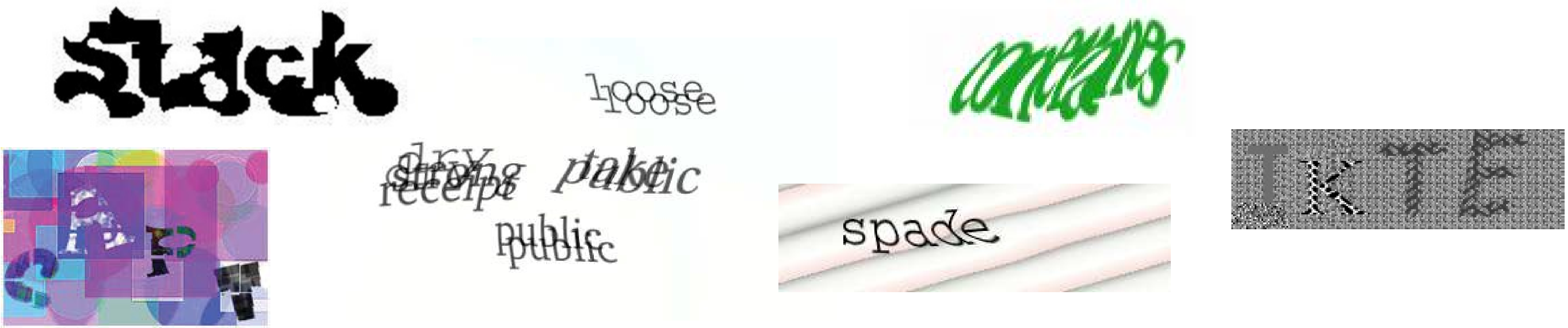
DotCHA: A 3D Text-Based Scatter-Type CAPTCHA

DotCHA: A 3D Text-Based Scatter-Type CAPTCHA

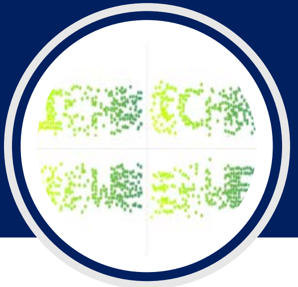
Various Types of CAPTCHAs

Text-based CAPTCHAs

- A sequence of alphabets and numbers
- Most widely used form due to ease of use and simple structure
- Noise and distortion added to make the test robust to automated attacks



Various Types of CAPTCHAs



Text-based CAPTCHAs

- A sequence of alphabets and numbers
- Most widely used form due to ease of use and simple structure
- Noise and distortion added to make the test robust to automated attacks

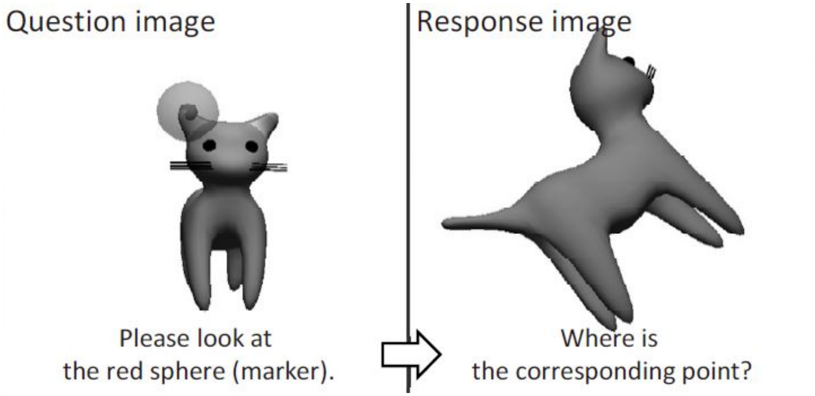
Pros Easy to **generate** and **identify**

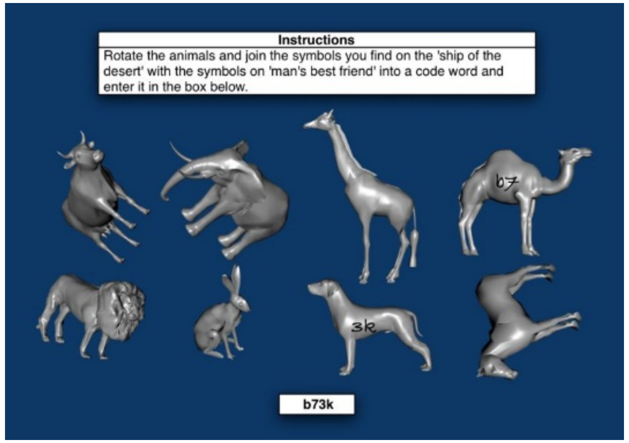
Cons Easy to be **recognized** through **OCR** and **machine learning attacks**

Various Types of CAPTCHAs

3D Model-based CAPTCHAs

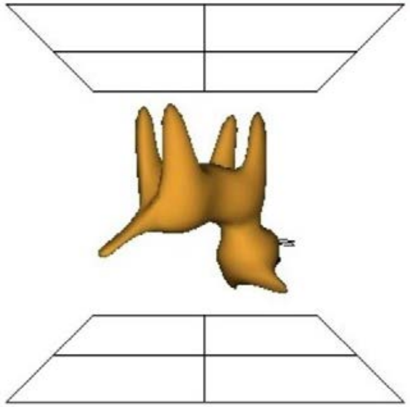
- Using rotation of 3D models
- Based on cognitive ability of humans, called *mental rotation*



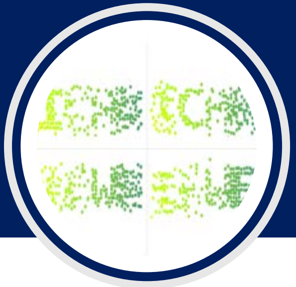


Instructions

Rotate the animals and join the symbols you find on the 'ship of the desert' with the symbols on 'man's best friend' into a code word and enter it in the box below.



Various Types of CAPTCHAs



3D Model-based CAPTCHAs

- Using rotation of 3D models
- Based on cognitive ability of humans, called *mental rotation*

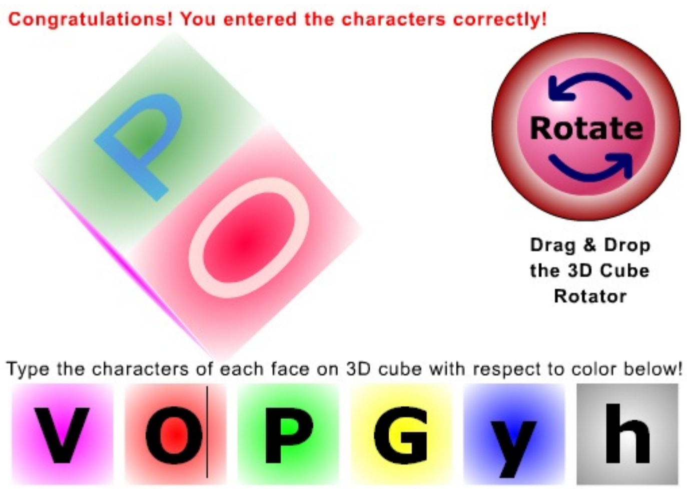
Pros Comparably **robust to bots** by minimizing legibility

Cons **Difficult** to solve even for human

Various Types of CAPTCHAs

3D Text Model-based CAPTCHAs

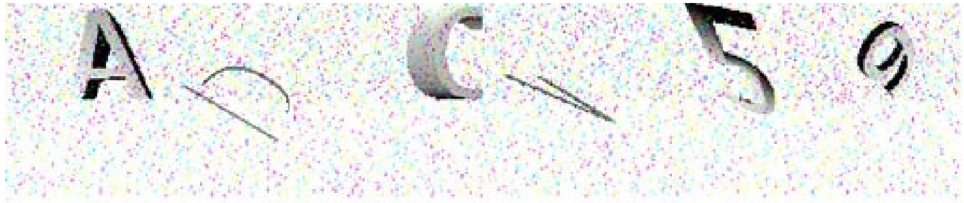
- Using 3D text-based models that originated from 2D text-based CAPTCHA

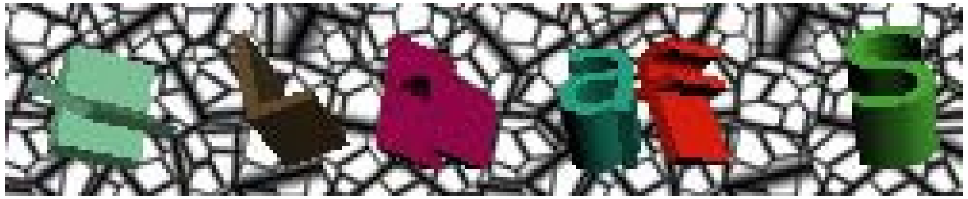


Congratulations! You entered the characters correctly!

Drag & Drop the 3D Cube Rotator

Type the characters of each face on 3D cube with respect to color below!





Various Types of CAPTCHAs

3D Model-based CAPTCHAs

- Using 3D text-based models that originated from 2D text-based CAPTCHA

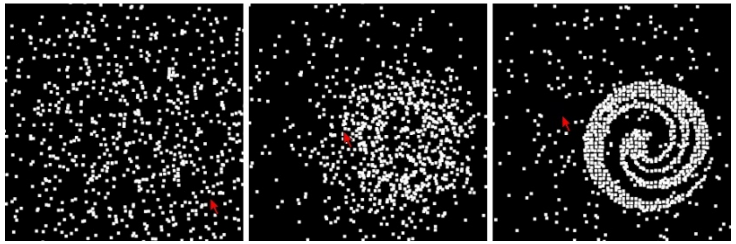
Pros **Familiar** to users who accustomed to 2D text-based CAPTCHA

Cons Vulnerable to **segmentation attacks**

Various Types of CAPTCHAs

Interactive CAPTCHAs

- Relying on user interaction
- Based on cognitive ability of human



Various Types of CAPTCHAs

Interactive CAPTCHAs

- Relying on user interaction
- Based on cognitive ability of human

Pros Mitigating **automated attacks**

Cons **Time-consuming** to solve the problem

DotCHA: A 3D Text-Based Scatter-Type CAPTCHA

DotCHA: A 3D Text-Based Scatter-Type CAPTCHA

Purpose of DotCHA

Security + Usability

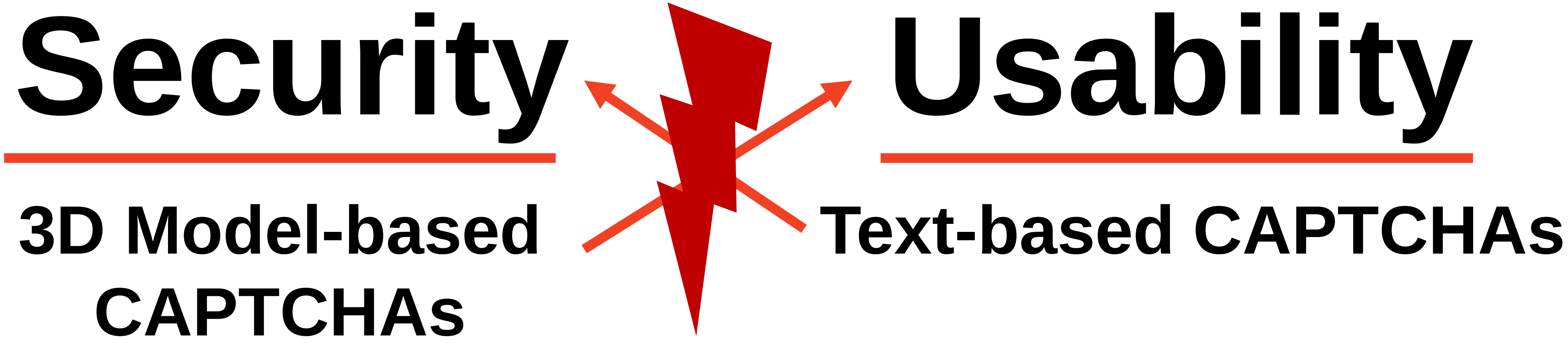
Purpose of DotCHA

Security + Usability

3D Model-based CAPTCHAs

Text-based CAPTCHAs

Purpose of DotCHA



Security

3D Model-based CAPTCHAs

Usability

Text-based CAPTCHAs

Purpose of DotCHA

Problem

Vulnerable to **OCR and machine learning attacks**

Security +

Text-based CAPTCHAs

3D Model-based

Solution

Making each letter **only legible** at a **unique rotation angle**

Purpose of DotCHA

Problem

Low correct response rate due to complicate operations: 1) **recognition** of 3D object and 2**)** **judging and inferring** the answer

3D Model-based CAPTCHAs

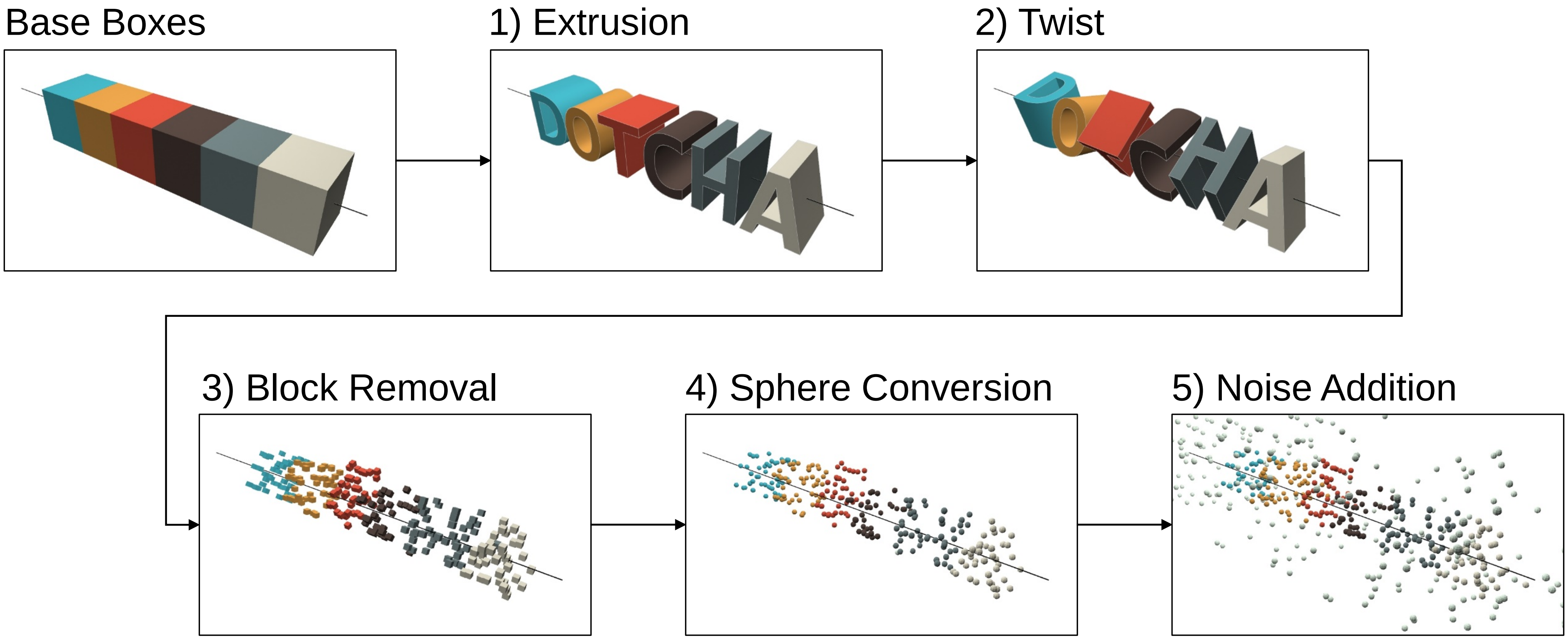
Usability

Solution

Fixing the rotation axis to a **single** axis

Generation of DotCHA

System Pipeline



Base Boxes

1) Extrusion

2) Twist

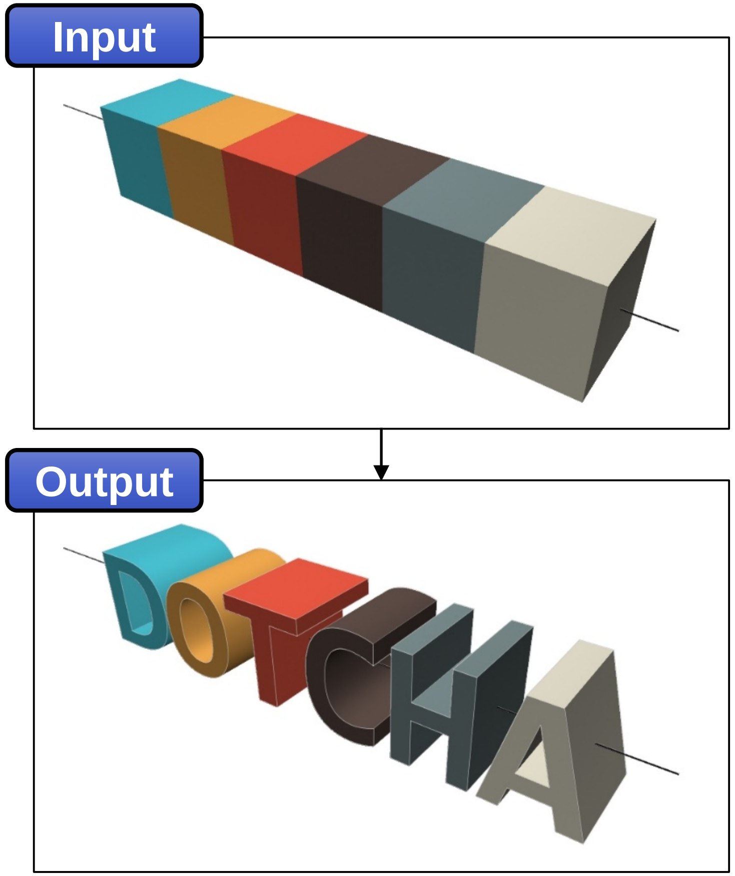
3) Block Removal

4) Sphere Conversion

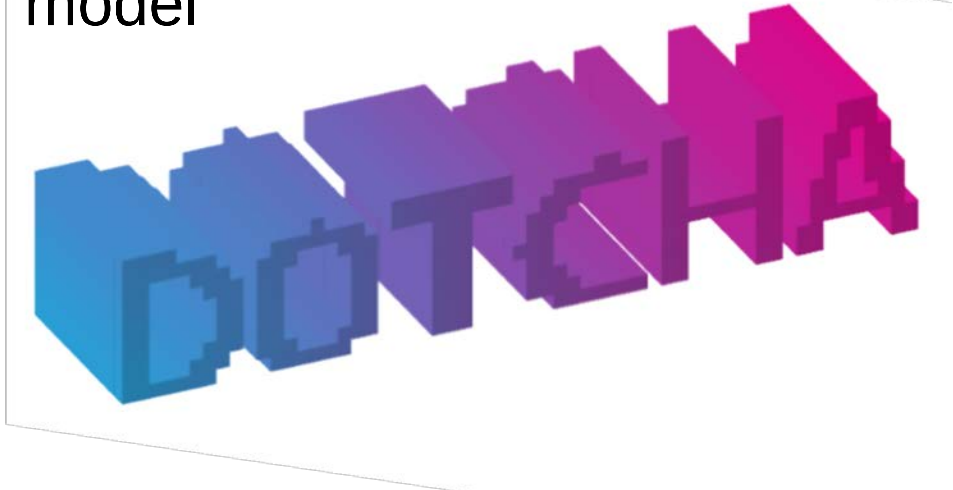
5) Noise Addition

Generation of DotCHA

(1) Extrusion

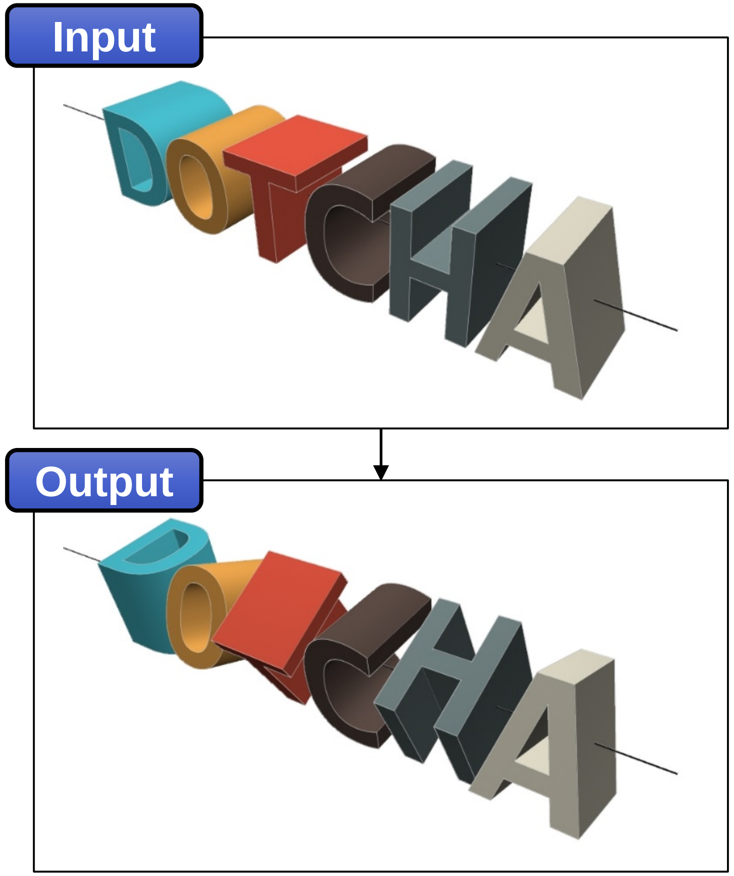


- Engraving the given letters on a **solid rectangular parallelepiped** model
- Cutting a solid cube model into **small unit blocks** (molecular construction)
  - Dividing a model into smaller units forming the larger model



Generation of DotCHA

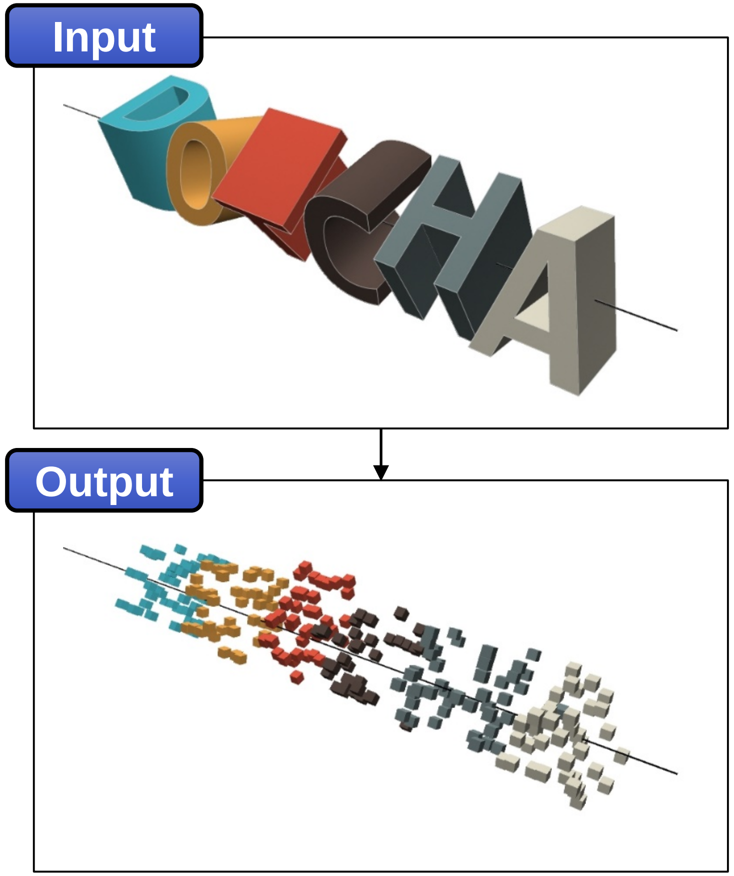
(2) Twist



- Rotating around the **center axis** of the rectangular parallelepiped model
- Rotating each letter at **unique angle** to ensure that the correct answers are not recognized **from a single direction**

Generation of DotCHA

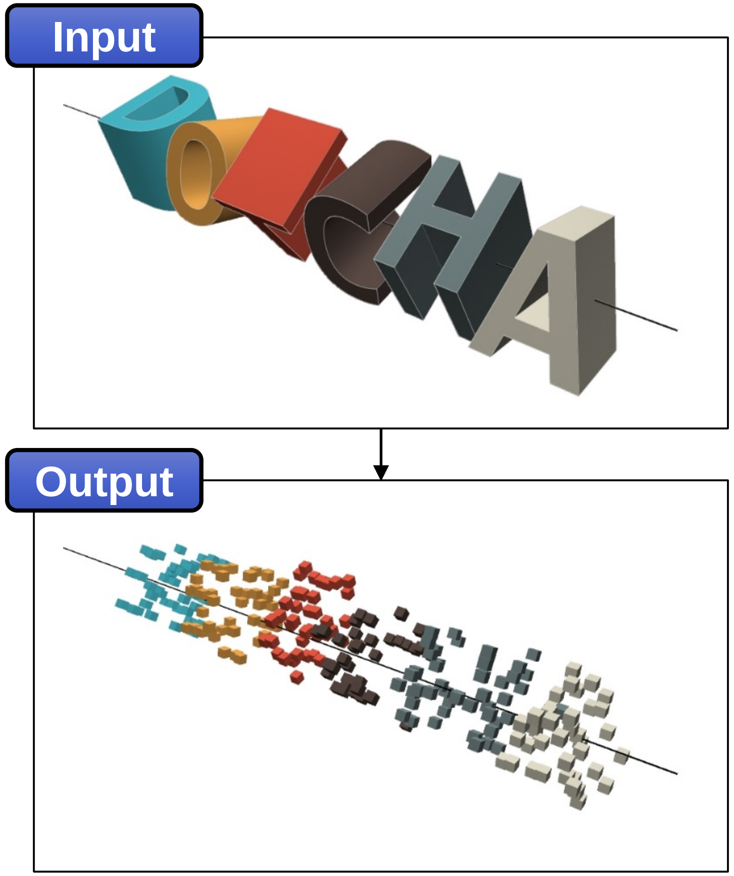
(3) Removal of Blocks



- Removing a set of unit blocks to make each letter recognized **only in one particular direction, not in any direction**
- Removal with **two conditions**:
    - Remove unnecessary blocks that do not affect the shape of the letters
    - Remove blocks evenly while preserving the balance between directions

Generation of DotCHA

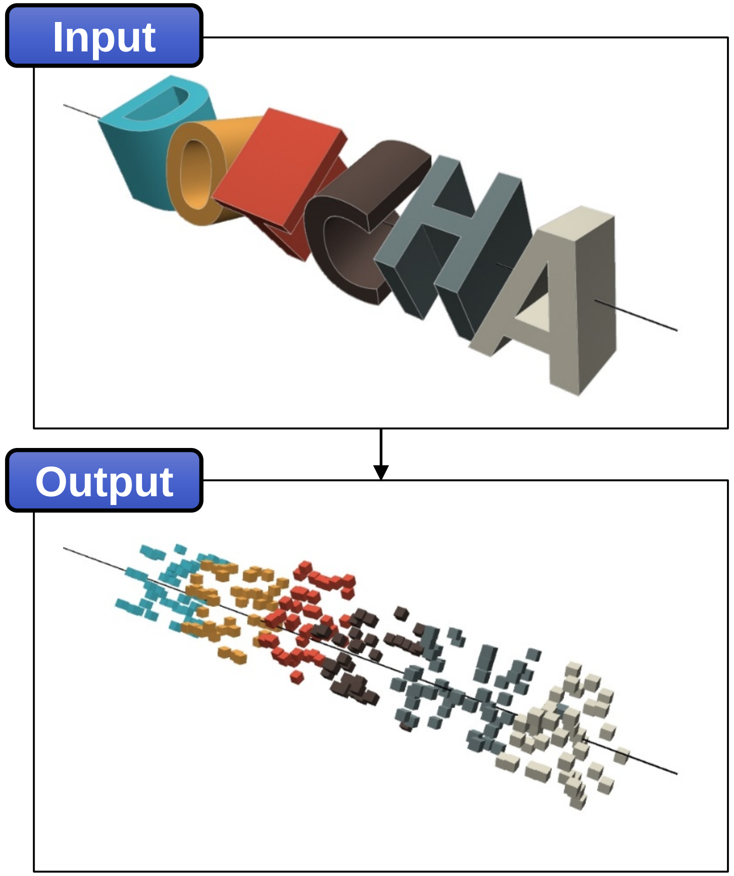
(3) Removal of Blocks



- Multigraph *G*
  - **Multiple edges** between a pair of vertices
  - **Vertices**: unit blocks
  - **Edges**: whether a pair of blocks are located on the same coordinates along the local *y* or *z* axes

Generation of DotCHA

(3) Removal of Blocks

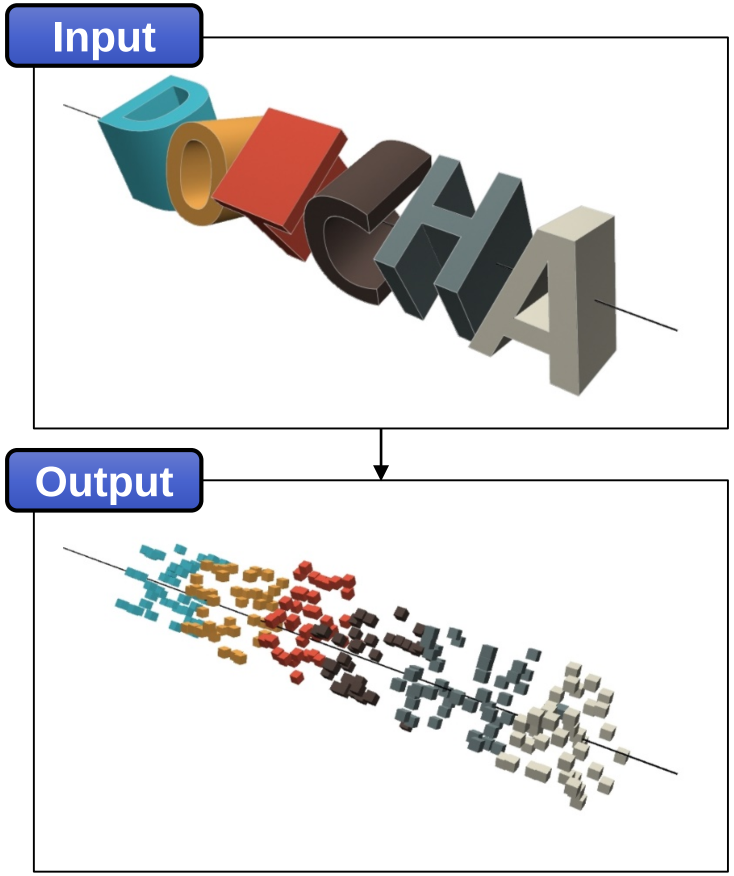


- Scoring function *S* of vertex *v*

$$S(v) = \alpha \cdot |N_R(v)| + |N_G(v)|$$

Generation of DotCHA

(3) Removal of Blocks



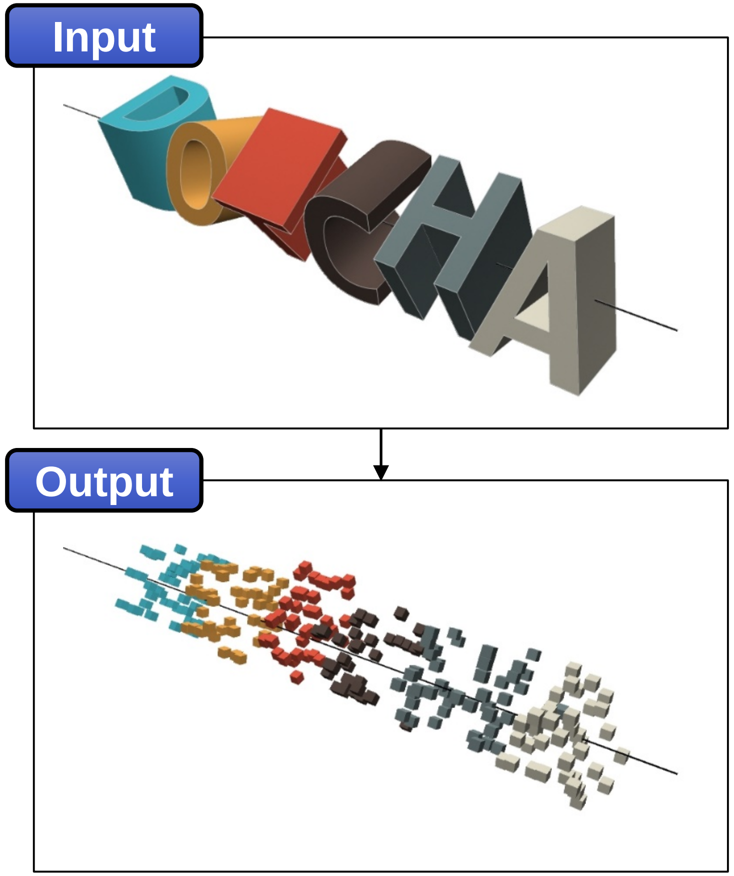
- Scoring function *S* of vertex *v*

$$S(v) = \alpha \cdot \underline{|N_R(v)|} + |N_G(v)|$$

- ***Dispersion of blocks in the cube***
- Counts of neighboring vertices
  - Neighbor: Euclidean distance from *v* is at most *k*
- The number of blocks existing around the block *v*

Generation of DotCHA

(3) Removal of Blocks



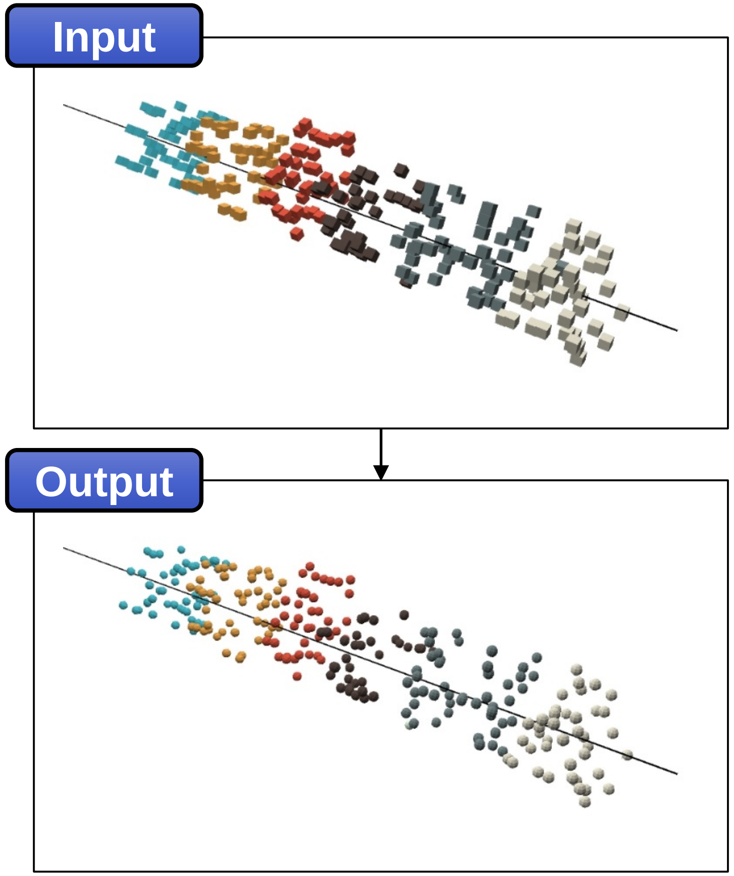
- Scoring function *S* of vertex *v*

$$S(v) = \alpha \cdot |N_R(v)| + \underline{|N_G(v)|}$$

- Counts of adjacent vertices of *v* in graph *G*
- The number of blocks placed along the local *y* and *z* axes

Generation of DotCHA

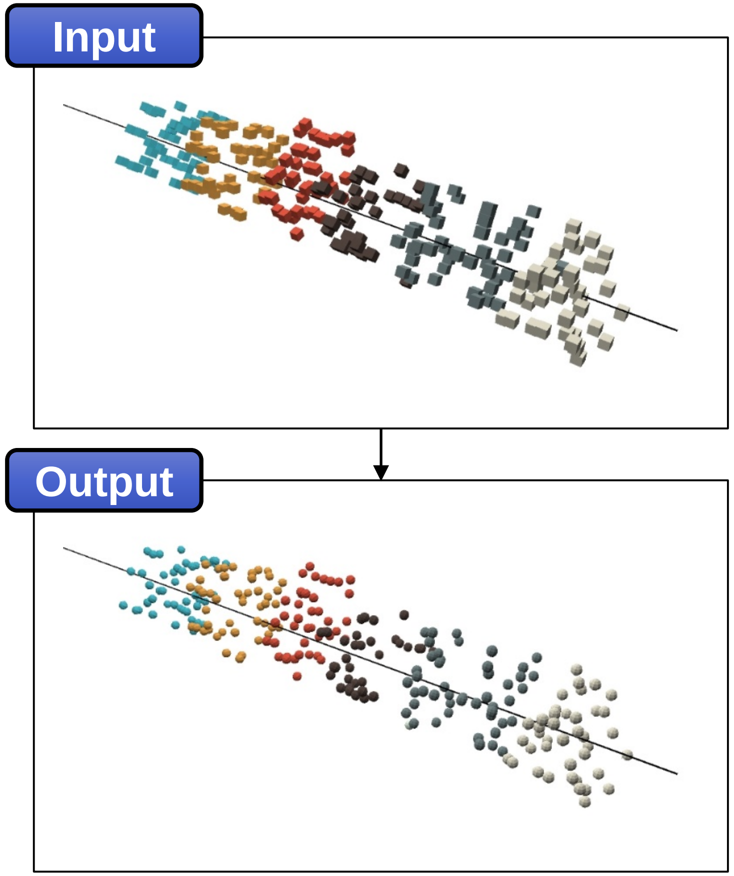
(4) Conversion to Spheres



- **Hiding the orientation** of the model inferred from the **edges** of the cube
- Conversion unit blocks into **unit spheres**
- Similar shape of **scatter-type** method

Generation of DotCHA

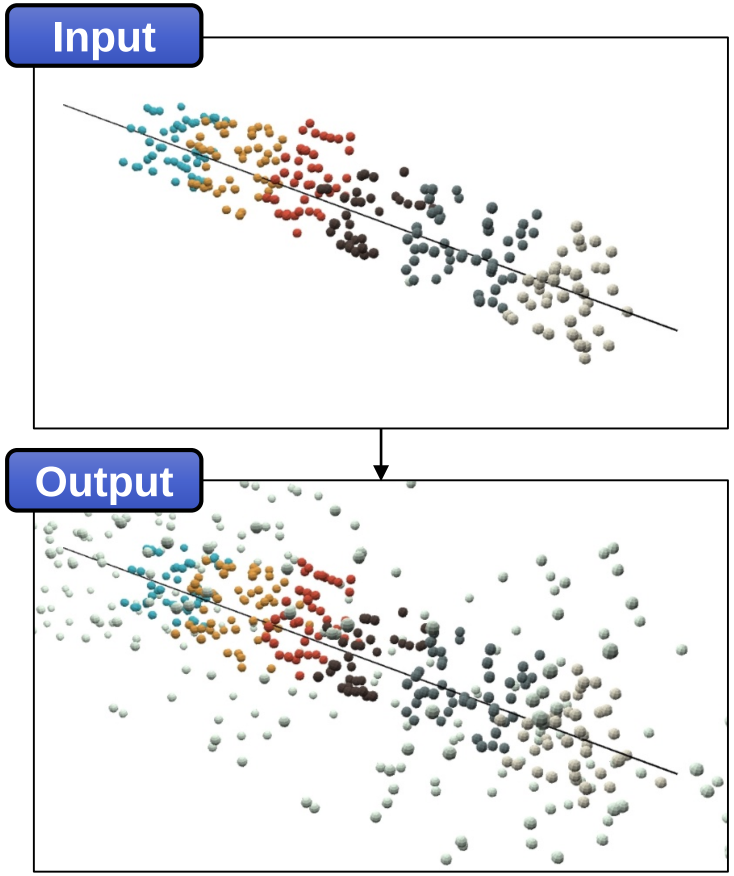
(4) Conversion to Spheres



- **Two parameters** to maximize the usability and security
    - Sphere radius (ρ)
        - The radius of unit sphere
    - Sphere offset (σ)
        - The location offset of the center of sphere from the center of the unit block

Generation of DotCHA

(5) Noise Addition



- Noise (δ)
  - The number of noise spheres

- Base on the **motion parallax** to give users the perception of **depth** from the relative motion between models

Experiments

Settings

- 6 letters
  - a combination of **random** alphabets to avoid **dictionary attack**
- Implemented using Three.js library on HTML5 Canvas
- k=10 alphabet pattern
- Considering several different attack scenarios
  1) Finding the correct view directions
  2) Reading the letters from the correct view directions

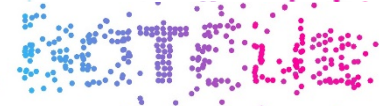
Experiments

(1) Finding the Correct View Directions

- Test whether the correct view directions can be identified
- Sampled 30 different views including 6 ground truth views
- Score the views through pixel counting and edge detection

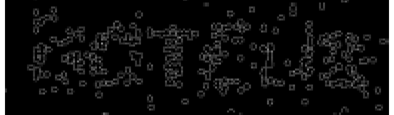
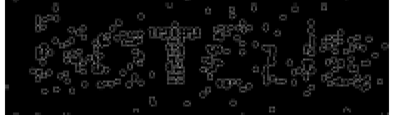
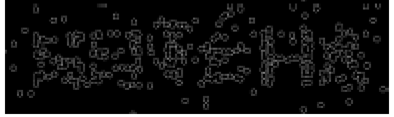
Experiments

(1) Finding the Correct View Directions

Pixel Counting Result					Ground Truth	
Largest		Smallest				
	241,608		238,191	A		240,544
	241,027		238,435	T		240,497
	240,956		238,518	C		240,276
	240,949		238,710	O		239,316
	240,806		238,855	H		239,180
	240,544		238,896	D		239,119

Experiments

(1) Finding the Correct View Directions

Edge Detection Result				Ground Truth	
Largest		Smallest			
	12,668		11,856 H		12,445
	12,564		11,788 O		12,335
	12,507		11,784 D		12,318
	12,459		11,742 C		11,981
	12,445		11,693 T		11,924
	12,404		11,685 A		11,784

Experiments

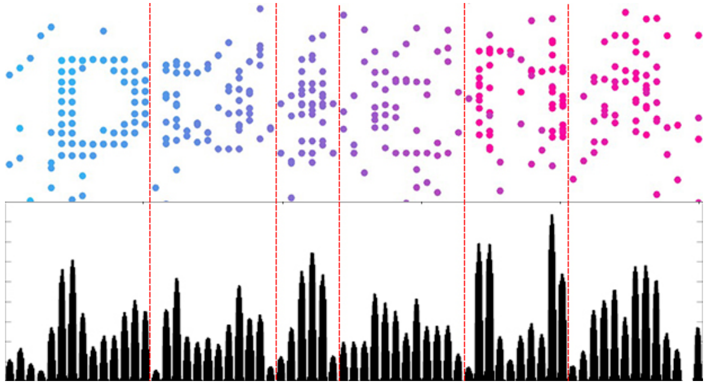
(2) Reading the Letters from the Correct View Directions

- Assuming the **correct view** directions **are given**
- Test whether letters **can be read**
- Recognition by using **OCR**
  - Using *Google Tesseract, ABBYY FineReader 14*
  - **Advantage of the scatter-type** CAPTCHA; OCR **engines could not completely recognize** any of the words, **even from the correct** view images.

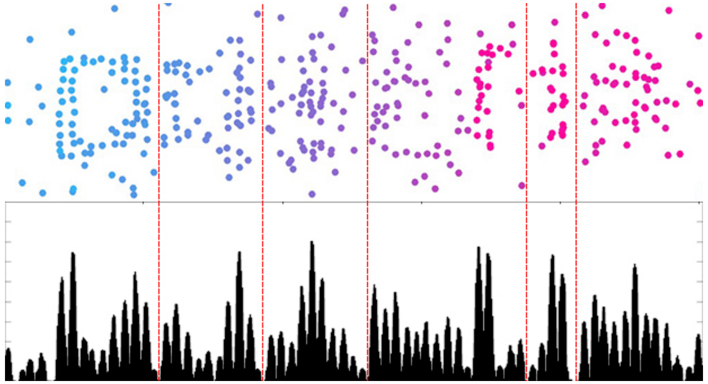
Experiments

(2) Reading the Letters from the Correct View Directions

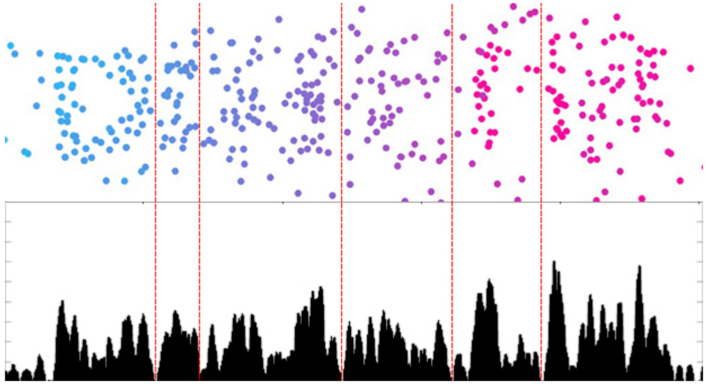
- Test robustness to the **segmentation attack**



$\sigma = 0$



$\sigma = 0.25$

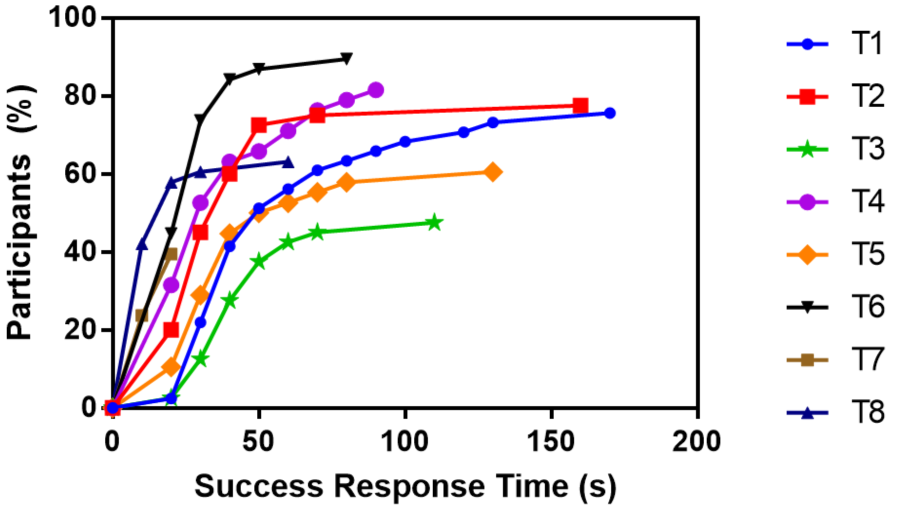


$\sigma = 0.5$

Experiments

(3) User Study

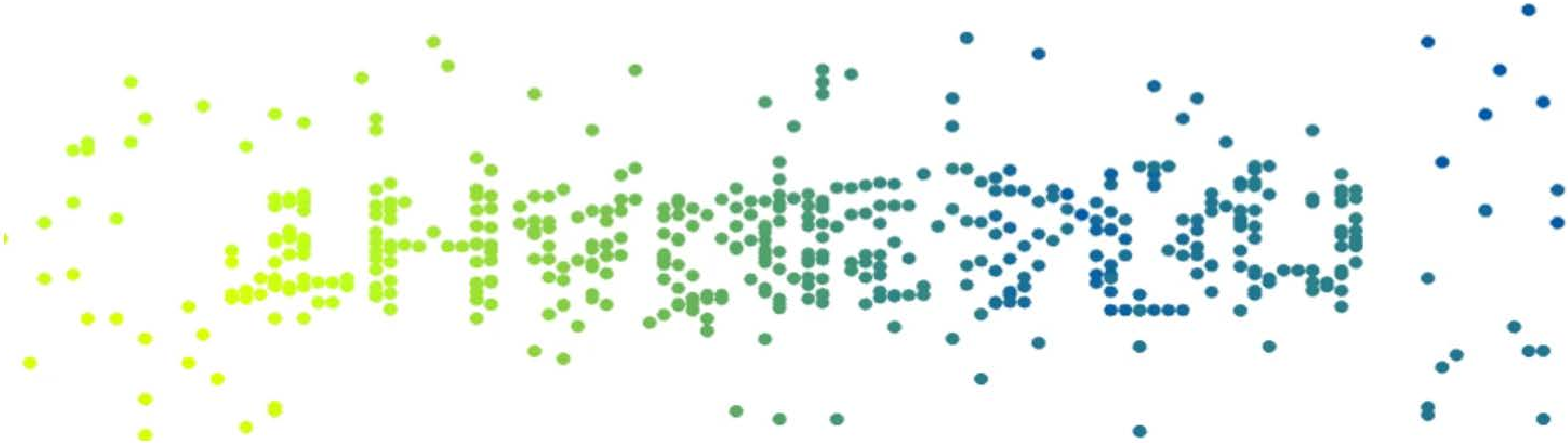
- 50 participants, recruited online
- 8 **unsupervised tests** using their own devices
  - 6 DotCHA challenges (T1-T6)
  - 2 reCAPTCHA (2D text-based CAPTCHA) (T7-T8)



Conclusion

- We introduce a new type of 3D text-based CAPTCHA, called DotCHA to overcome the limitations of existing 2D and 3D approaches.
- To improve the usability while preserving security, automated rotation and interactive systems are combined in our demo.
- A unit sphere can be replaced with a set of particles to improve the security of DotCHA.

- Code is available: https://github.com/SuziKim/DotCHA



QnA