

# Two Novel Image-Based CAPTCHA Schemes Based on Visual Effects

Ping Zhang, Haichang Gao<sup>(✉)</sup>, Zhouhang Cheng, and Fang Cao

Institute of Software Engineering, Xidian University,  
Xi'an 710071, Shaanxi, People's Republic of China  
hchgao@xidian.edu.cn

**Abstract.** CAPTCHA is a security mechanism designed to differentiate between computers and humans, and is used to defend against malicious bot programs. Text-based and image-based CAPTCHAs are two of the most widely deployed schemes, but most of the existing CAPTCHA schemes are either too difficult for humans to solve or not safe enough since a lot of researchers have attacked them successfully. These CAPTCHAs are also language-dependent and they cannot be automatically generated. So it is urgent to explore new possible schemes of CAPTCHA. In this paper, we mainly made two contributions to CAPTCHA. First, we propose two novel image-based CAPTCHA schemes based on visual effects. DeRection is one CAPTCHA scheme that takes advantage of human ability on capturing deformed regions of an image in the case of contrast. CONSCHEME is another one that capitalizes on human ability of understanding the content in a three-dimensional space. We conducted preliminary experiments over more than 110 users to verify the usability and security of two schemes. Second, by analyzing the characteristics of these two schemes proposed in this paper and comparing them with the existing image-based ones, we propose a set of new guidelines for the design of image-based CAPTCHA.

**Keywords:** CAPTCHA · Language independent · Web service  
Visual effects · Image-based

## 1 Introduction

CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart) [1] is a program which is used to tell computers and humans apart. It can protect the websites from cracking password, automated voting, receiving spam mails, automated registering accounts and attacking password systems.

Currently, most websites adopt text-based CAPTCHA that using sophisticated distortion, rotation or noise interferences to prevent from machines recognition. Users need to decipher characters within an image. However, the distorted characters really reduce human accuracy, and such schemes were broken by [2–8]. Therefore, image-based CAPTCHA becomes a promising alternative because

these schemes have many advantages, for example, being not vulnerable to segmentation attacks, and being user-friendly by mouse-based interaction. But it also suffers from various drawbacks, the images adopted by many image-based CAPTCHAs need to be manually tagged which means these schemes are not automatically generated, and most image-based CAPTCHAs are still dependent on language. Furthermore, with the rapid development of the deep learning algorithm, most text-based and image-based schemes that based on classification can be well attacked. In addition, audio and video-based CAPTCHAs have lower utilization rate and require higher bandwidth, many websites don't support these types of CAPTCHAs. Therefore, it is urgent and difficult to design a widely usable CAPTCHA.

In this paper, we present two schemes that can overcome the issues above. In the DeRection (Deformed Regions Detection in a GIF image) scheme, we present a GIF image which contains ten frames and every frame randomly has 2 to 6 deformed regions. Users need two steps to pass the challenge. First, click the GIF image to focus on one frame. Second, click all the deformed regions in this frame. In the CONSCHEME (Counting the Number of Stacking Cubes in a Three-dimensional scene) scheme, we use Unity to create a three-dimensional space with several cubes stacking together. Interacting with mouse and keyboard, users can get any perspectives of the scene. This challenge requires users to count the quantity of the cubes.

The two mechanisms have many features in common. Firstly, both of them have low language dependence since users only need to click on the deformed regions within the image in the DeRection and enter a number of cubes in the CONSCHEME. Besides, they can be generated online automatically. The images of DeRection can be obtained from websites and all we need from the CONSCHEME is a Unity plugin that can be embedded into web browser and reused once generated. In contrast, many existing image-based CAPTCHAs need a database of manually tagged images. By definition, they are not real CAPTCHAs which must be a completely automated public Turing test. We have demonstrated experiments to evaluate these two schemes and the results indicate that they both have good robustness and usability. The limitations of the proposed schemes are that they require higher bandwidth compared with traditional image-based CAPTCHAs. And users need to set up the Unity plugin when using CONSCHEME for the first time.

The rest of this paper is organized as follows. In Sect. 2, we introduce the related works. In Sect. 3, we provide the details of design and implement of the schemes. In Sect. 4, we present the experiments and analysis of the proposed scheme. And in Sect. 5, we propose a set of image-based CAPTCHA design guidelines. Finally, we make a brief conclusion in Sect. 6.

## 2 Related Works

CAPTCHA relies on the capability gap between humans and computers in solving Artificial Intelligence problems. Existing CAPTCHA systems can be broadly

grouped into three classes: (1) text-based, (2) image-based, and (3) audio or video-based.

Text-based CAPTCHAs are the most widely used schemes [2]. Designers usually use sophisticated distortion, rotation, noise and complex background on English letters and Arabic numerals to prevent automatic attack from achieving a high success rate.

Security is the most worrying factor of text-based CAPTCHAs. A large number of attack methods have been proposed. In 2003, Mori and Malik [2] broke EZ-Gimpy and Gimpy by analyzing the shape context with a success rate of 92% and 33% respectively. In 2008, Yan and El Ahmad [3] proposed a new segmentation method to attack Microsoft MSN with 92% segmentation and 60% overall success rate. In 2013, Gao et al. [4] broke several hollow CAPTCHAs with a generic method with the success rate range from 36% to 89%. In 2014, Goodfellow et al. [5] using deep CNN solving reCAPTCHA with 99.8% success rate. In 2015, Karthik and Recasens [6] applied CNN-based method on Microsoft CAPTCHAs with 57.05% CAPTCHA success rate of. More recently, in NDSS 2016, Gao [7] found a simple generic attack that firstly employed Gabor filter on attacking a wide range of CAPTCHAs.

With each failed CAPTCHA scheme, CAPTCHA designers accumulate experiences and then design better schemes, with increased friendly usability and improved security. Image-based CAPTCHAs have been proposed as an alternative for the text-based CAPTCHA systems. They can be broadly divided into two types.

First type is to find a corresponding word to describe images, such as Naming CAPTCHA [8] and IMAGINATION [9]. Naming CAPTCHA requires user to type a word to describe the common object in six images. IMAGINATION asks user to click an image’s geometry center from a synthetic image then select a correct label from a given list. However, typing a word may cause misspelling and polysemy, and the random guess rate of selecting from a list is relatively high. This type also has strong language dependence. IMAGINATION was attacked by Zhu et al. [10].

Second type is to classify the given images. For example, Asirra [11] shows 12 images of cats and dogs, asking user to select all the cats. SEMAGE [12] shows 8 real or cartoon images of animals, allowing user to choose images of same animals. 12306.cn scheme [13] shows 8 images, users need to recognize the text above the images and pick out all the targets to match the text. This type requires a database of correctly tagged images, which causes a waste of resources. And once the dataset is obtained, it will be much easier to break them. Zhu et al. [10] have broken Asirra with a success rate of 10.3%.

There are some other types of image-based CAPTCHAs. “What’s up” [14] asks user to adjust the orientation of three images. ARTiFACIAL [15] asks user to identify a face and click its six points from a synthetic image and it has been broken by Li [16]. FR-CAPTCHA [17] and FaceDCAPTCHA [18] are two face-based schemes, the former asks user to select two face images of the same person while the latter is to distinguish human faces from cartoon-face images. Our team has broken these two CAPTCHAs with 42% and 48% success rate respectively [19].

### 3 Design and Implement

#### 3.1 DeRection

In DeRection scheme, the users are given a GIF image, every frame of this image has 2 to 6 deformed regions. Users need to find out all of the deformed regions in the frame to pass the challenge. Users can click the “Change an Image” button to change the GIF image and click the “Change a Frame” button to change a frame of the GIF image.

Every GIF image is made up of 10 images with randomly 2 to 6 deformed regions. We download a picture from the Internet and judge if it is appropriate to be deformed by calculating its texture complexity. If it is, we use it to generate 10 images with different deformed regions and the position to be deformed should also be suitable. So the whole problem relies on finding a method to calculate texture complexity of an image.

There are many methods to extract features from an image and measure the texture complexity. Among these methods, GLCM (Gray-level co-occurrence matrix) [20] is a widely-used and well-performed one, we also adopted it in this paper. GLCM is the statistical method of examining the textures that considers the spatial relationship of the pixels. The GLCM characterizes the texture of an image by calculating how often pairs of pixel with specific values and in a specified spatial relationship occur in an image, creating a GLCM, and then extracting statistical measures from this matrix.

According to [20], we calculated the ASM (Angular Second Moment), Entropy, Contrast, IDM (Inverse Different Moment) and Correlation values and set weights to them based on the GLCM of three directions, they are  $0^\circ$ ,  $45^\circ$  and  $90^\circ$ . The texture complexity of a region in an image is measured by the formula:

$$R = \sum_{i=1}^3 (a_1 \cdot J_i + a_2 \cdot H_i + a_3 \cdot G_i + a_4 \cdot Q_i + a_5 \cdot Conv_i) \quad (1)$$

$J$ ,  $H$ ,  $G$ ,  $Q$  and  $Cov$  represents ASM, Entropy, Contrast, IDM and Correlation respectively.  $a_1$ ,  $a_2$ ,  $a_3$ ,  $a_4$  and  $a_5$  are constant parameters and their values are 0.2012, 0.2673, 0.2814, 0.1517 and 0.1126.

However, in our experiments, we cannot find a suitable threshold to accurately divide the regions. To solve this, we tested each of the feature parameters separately and there are many discoveries. For example, the complexity of the image is not strictly related to the value of Correlation. Through a series of experiments, finally we adopted Entropy and Contrast and introduced Variance, a new feature parameter to measure the texture complexity. Variance describes the discrete degree of the value of the variables to their mathematical expectation. We used Variance to measure the degree of discrete of the grayscale value of an image. Finally, we calculated according to the formula:

$$R' = \sum_{i=1}^3 (0.5 \cdot E_i + 0.5 \cdot Conv_i) + \frac{v}{10} \quad (2)$$

We calculated the GLCM of three directions, and we set the final threshold of  $R'$  to 14. This method is acceptable for classifying effect and time. Then, we compared the value of  $R$  and  $R'$  and we found that the images got with  $R$  has a large ambiguous area, which made it hard to divide the regions into two parts. We tested 50 different regions in more than 450 different images and got the  $R$  and  $R'$ , and we found that ambiguous area got from  $R$  is about 62.4%. So we finally choose  $R'$  to determine whether a region is suitable to be deformed and got a relatively satisfactory result. If the region is selected as deformed region, then we need to perform deformation. Deformation is all about moving pixels from one coordinate to another coordinate. We use the formulas below:

$$newX = \frac{(x - cenX)\sqrt{(x - cenX)^2 + (y - cenY)^2}}{radius} + cenX \quad (3)$$

$$newY = \frac{(y - cenY)\sqrt{(x - cenX)^2 + (y - cenY)^2}}{radius} + cenY \quad (4)$$

$(newX, newY)$  is the coordinate of the pixel in the original image and  $(x, y)$  is the coordinate in the processed image.  $(cenX, cenY)$  and radius are the coordinates of the center and the radius of the deformation region. Figure 2 shows the effect of convex lens deformation. The origin of the deformation is at the origin of all four different color concentric circles. While calculating mapping, the mapping point of the original point within the deformation area, which is a circle, moves along the direction away from the center of the circle.

The original version of DeRection is presenting users with a static image. But the results did not reach our expectations. Through analysis we found that the deformation regions are hard to find in a static image, but in GIF image, they can capture people's eyes. So we generated many GIF images and users need to click the GIF image to get one frame of the GIF image. The verification mode remains the same but it is much easier for users to pass the challenge. But the problem came that crackers can solve our scheme by comparing the pixels between frames to find out the deformation regions. To overcome this, we modified every pixel value after generating deformed regions. Figure 2 shows one frame of final version of DeRection.

### 3.2 CONSCHEME

CONSCHEME is another image-based CAPTCHA scheme we proposed, which is an interactive three-dimensional CAPTCHA system created by Unity.

We use Unity to create a three-dimensional room on the floor of which are a lot of cubes stacked. The walls, ceiling and floor are labeled the same stickers as the cubes. An example of CONSCHEME and its three views are shown in Fig. 3. The original version of this scheme had no floor or walls and the cubes just float in the air with no texture, and people felt dazzled when rotate the cubes. With texture and walls, people can focus on the cubes in proper perspective. Benefit from Unity's good interactivity, users can scroll the mouse wheel to zoom in or zoom out the cubes, click the arrow keys on the keyboard to rotate the object

and drag with the left mouse button to change the perspective of the cubes. When a unity-plugin is produced, we export it into web format and embed it into the web page. It can produce different number of cubes stacked together in different ways. Users are asked to count out the number of cubes and input it to pass the challenge.

## 4 Experiment and Analysis

To verify the design idea of our schemes, we carried out an experiment and asked human users to use our proposed CAPTCHAs and record relevant data such as accuracy and recognition time. Participants are more than 110, mostly sophomores in Xidian University. We built a website to present users with the challenge of our schemes to collect data. Users can submit their homework only if they have passed the challenge. It took us months to collect and analyze the data of all sets. We then improved our schemes according the data and the feedback information from the users and got the final versions of both schemes. The legal data of each scheme are shown in Table 1.

**Table 1.** The accuracy rate and recognition time of each set.

Scheme	The right number	Total number	Accuracy rate	Recognition time
DeRection	911	1054	86.43%	12.77s
CONSCHEME	832	978	85.07%	15.12s

As shown in Table 1, the accuracy rate of DeRection and CONSCHEME are roughly the same and higher than 85%. We can draw a conclusion from the data that for both of the schemes, users can pass approximately eight to nine challenges when they are given ten of our CAPTCHAs. The Table 1 also shows the recognition time of each scheme. It takes 12.77 s and 15.12 s to complete the DeRection and CONSCHEME. It is widely accepted that CAPTCHA should be completed no more than 30 s [1], so the two schemes are satisfying under this principle.

### 4.1 Design Analysis

Usability and robustness are the most important features that all the CAPTCHA schemes focused on. We tried but failed to find a CAPTCHA similar to ours from the existing CAPTCHAs. Finally, we decided to take the Asirra to compare with ours for its complete data. Asirra, as showed in Fig. 1, shows 12 images of cats and dogs and asks users to pick out all the cats. Similar to our first scheme, users need to click some regions in an image. And the solution space is finite for the limited pictures in the image, which makes it comparable to our CONSCHEME.

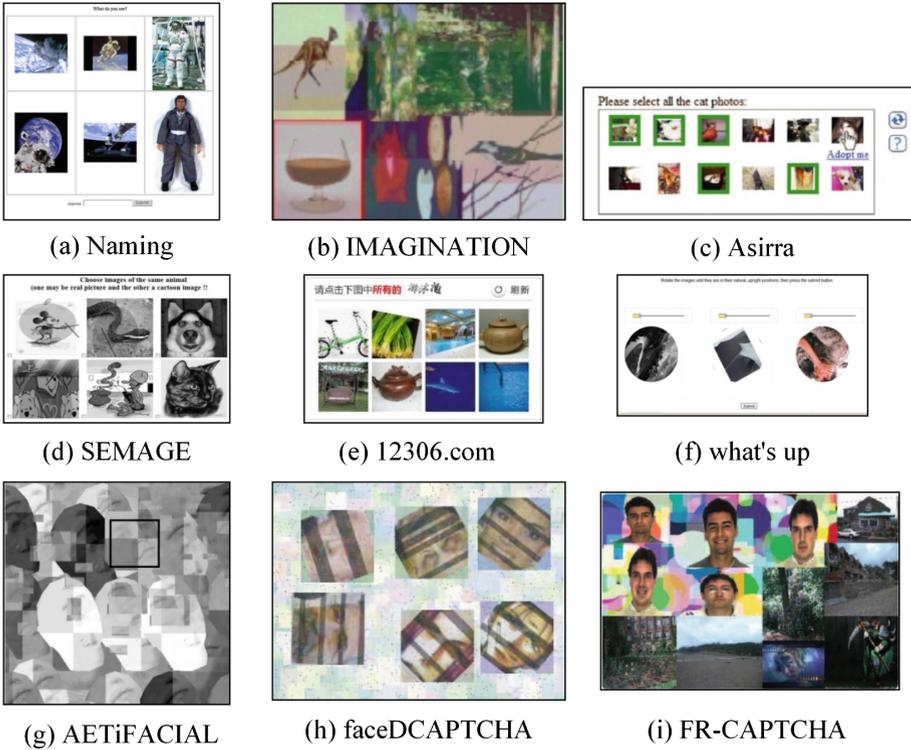


Fig. 1. Examples of existing image-based CAPTCHA schemes.

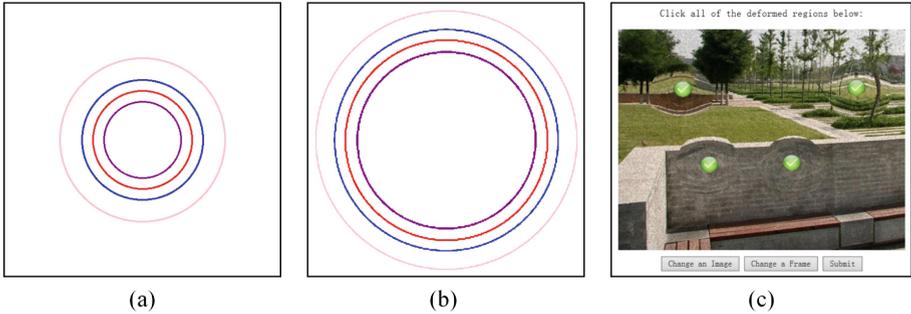
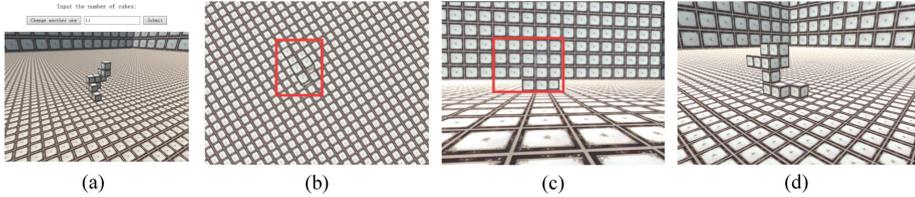


Fig. 2. The effect of convex deformation and an example of DeRection: (a) Original image; (b) After deformed; (c) An example of DeRection.

**Usability analysis.** Good usability indicates that users can quickly and accurately pass the test. However, in pursuit of robustness, many schemes usually sacrifice the usability too much. Some image-based CAPTCHA schemes carry out excessive deformation, noise and rotation. For example, the CAPTCHA scheme



**Fig. 3.** Three views of the stacked cubes in CONSCHEME: (a) Example of CONSCHEME; (b) Top view; (c) Main view and left view; (d) General view.

adopted by 12306.cn [13] mainly adopt 8 low-resolution images thus it is very difficult for users to recognize and distinguish all of the objects, so the usability of it is reduced. In fact, when firstly put into use, the 12306.cn scheme got a huge number of complaints. Asirra has good usability because the small images are clearly and orderly placed and users only need to click the cats. The schemes of ours are also user friendly. In DeRection, users can capture the different regions in GIF image easily. The CONSCHEME utilizes human’s ability to recognize objects in three-dimensional space, which is a capability that human have been trained and used since their birth. What’s more, the feedback messages say this scheme is attractive and interesting.

**Generation analysis.** According to the definition of CAPTCHA, can be automatically generated and evaluated is the basic criterion for CAPTCHA design. However, due to the difficulties in generation, most of existing image-based CAPTCHA schemes adopt a tagged images database that already existed or established during the generation process of the designers. While generating, they just use tagged images from the database and put them together in some ways. Asirra adopted a tagged images database to generate their challenges. On the contrary, both of our proposed CAPTCHAs can be generated online and do not adopt any tagged images database. In the DeRection scheme, all the original images we adopt can be downloaded from the Internet real time, so it does not adopt any image database. As with the CONSCHEME scheme, once the unity plugin is finished, it can generate different number of stacked cube while using.

**Language independence.** Our designs are language independent which can be used by people all over the world. As mentioned above, users using our schemes only need to count out how many cubes there are in the space or pick out all of the deformed regions in the image, thus it is not necessary for user to master English or other specific language. Asirra is also language independent. Unfortunately, some image-based CAPTCHA systems are strongly language dependent, such as IMAGINATION [9] asks users to select a correct label from a given list and Naming CAPTCHA [8] requires users to understand the objects and type a word that describe the object of images in the textbox.

## 4.2 Security Analysis

**Random guess attack.** In the DeRection, the size of every image is  $450 * 350$  pixels, and the maximum radius of the deformed regions is 50 pixels. Considering there are three deformed regions, the accuracy rate of random guess attack is about 0.0046%. Not to mention the rate of quantity of the deformed regions and the tolerance of the clicked zone is smaller than  $50 * 50$  pixels. As with CONSCHEME scheme, the number of stacked cubes is between 5 and 17 so the random success rate is  $1/13$ . Nevertheless, we can set multiple rounds of challenge. Such technique is already in use in Assira [11] and SEMAGE [12]. Furthermore, web service providers using our scheme can adopt CAPTCHA Token Buckets [11] to make it stronger and more secure that attackers cannot break our schemes without efforts.

**Other attacks.** During our design, we considered a lot of issues to achieve better robustness. In DeRection, after generating deformed regions, we modified every pixel value to resist attack through simple comparison between pixels. In CONSCHEME, the cubes' color and texture are the same as backgrounds, and the angles to get the three views are missed so bots cannot easily attack it by mathematical means. However, Asirra is broken at 10.3% by support vector machine classifiers trained on color and texture features extracted from images. Furthermore, some traditional image-based CAPTCHA schemes adopt a limited tagged image database and the labels and images are both limited, once we get the whole database by constantly refreshing and saving, we can solve out them without too much price. For example, the 12306.cn scheme contains about 580 categories and not more than one million images through a rough estimate. If the time is long enough, people can get the whole database thus break this scheme.

## 5 Image-Based CAPTCHA Design Guidelines

We analyzed the design process and the characteristics of our proposed CAPTCHA schemes and summarized a series of guidelines as follows:

- **Automate generate (AG):** The challenge should be generated automatically. But many existing schemes need human labor to gather raw material or tag images. For example, the images adopted “what’s up” [14] CAPTCHA should be preprocessed by removing bad images by human.
- **Machine attacks (MA):** The challenge should be strong enough and the best existing techniques are far from solving the problem or there is no existing specific ways to attack the test successfully.
- **Resistance to no-effort attacks (RTNA):** The challenge should survive no-effort attacks. No-effort attacks are those can solve a CAPTCHA challenge without solving the hard AI problem. For example, “What’s up” CAPTCHA was broken by random guess attack with success rate of 4.48% [10].
- **Secret database (SD):** Whether the adopted database is open or not.

- **Get the database (GTD):** Whether the attackers are able to access the secret database. Some databases are available while others are not. For example, Asirra’s image database is provided by a novel, mutually beneficial partnership with [Petfinder.com](#) and attackers may get the database from the website. And images adopted by 12306.cn are kept secret and crackers need to keep refreshing the login interface to get the whole database.
- **Easy to human (ETH):** The challenge should be quickly and easily used by a human user. Any test that requires longer than 30s becomes less useful in practice. And the average success rate of the user to pass this challenge should be as large as possible. The CAPTCHA adopted by the 12306.cn scheme got a huge number of complaints when firstly put into use.
- **Input modality (IM):** The challenge should be easy for users to interact with. The basic interactive forms include typing the keyboard, clicking the mouse and touching screen. The clicking is better than typing in to the specified textbox.
- **Language dependence (LD):** The challenge is related to a changing keyword. After users have acquired verification mode of the CAPTCHA, there are still some variations of the keywords in many tests, and users need to understand what these keywords mean to pass the challenge. For example, in every 12306.cn challenge, there is a changing distorted keyword that representing in the upper images for users to recognize.
- **Educational difference (ED):** Whether educational differences have an impact on passing this challenge. Some categories of objects need to be known by learning, and people may know something in different ways due to cultural differences.

According to the factors above, we can summarize guidelines about evaluating a CAPTCHA scheme. Firstly, a good CAPTCHA scheme should be generated automatically without a database of tagged images or with a database accessible to all of us. Besides, a good scheme can resist to common machine attacks and easy no-effort attacks such as random guess algorithm. What’s more, to reach a good usability, the higher the success rate and the faster the pass time, the better. Generally it is acceptable that the accuracy rate should not be lower than 80% and the recognition time should not be longer than 30s. It is better but not necessary to offer an interesting CAPTCHA. Then, the operation method should be as simple as possible, it is best for users to just use the mouse. But typing with keyboard is also generally accepted since nearly all of the text-based and many image-based CAPTCHAs require users to input something. Last but not least, the CAPTCHA scheme is best to be language independent and the users should not be required with any educational background.

As is shown in Table 2, the two schemes of ours are promising. They can be generated automatically with no database and there are no existing machine attacks or no-effort attacks against them. And users only need mouth and keyboard to use. The accuracy rate and recognition time are both acceptable but might be worse than the others in Table 2, but we have to take different

**Table 2.** The guidelines and evaluation of existing CAPTCHAs. (“—” means no data available or required, K&M means Keyboard and Mouse).

Schemes	AG	MA	RTNA	SD	GTD	ETH	IM	LD	ED
Naming	No	98.44% [21]	Yes	Yes	Easy	74%/24s	Keyboard	Yes	Yes
What’s up	No	—	No [10]	Yes	Hard	84%	Mouse	No	No
Asirra	No	10.3% [10]	Yes	Yes	Easy	83.4%/15s	Mouse	No	No
ARTiFACIAL	Yes	18%/1.47s [16]	Yes	No	—	99.7%/14s	Mouse	No	No
SEMAGE	No	—	Yes	Yes	Hard	94.8%/11.64s	Mouse	No	No
IMAGINATION	No	4.95%/0.96s [10]	Yes	Yes	Easy	85%	Mouse	Yes	Yes
12306.cn	No	—	Yes	Yes	Hard	—	Mouse	Yes	Yes
FaceDCAPTCHA	No	48%/6.2s [19]	Yes	Yes	Easy	92.47%	Mouse	No	No
FR-CAPTCHA	No	42%/<14s [19]	Yes	Yes	Easy	94%	Mouse	Yes	No
DeRection	Yes	—	Yes	No	—	86.43%/12.77s	Mouse	No	No
CONSCHEME	Yes	—	Yes	No	—	85.07%/15.12s	K&M	No	No

experimental backgrounds into consideration. For example, the results of the Naming CAPTCHA are not that convincing because they came from 20 users who were paid \$10–\$15 for completing 100 rounds challenge.

## 6 Conclusion

In this paper, we propose two image-based CAPTCHA schemes both of which are based on visual effects. The DeRection asks users to find all of the deformed regions in one frame of a GIF image. And the CONSCHEME CAPTCHA asks users to count the number of stacking cubes in a three-dimensional scene.

By analyzing the results obtained using demographically diverse group of volunteers, we can assert that both the DeRection scheme and the CONSCHEME scheme are convenient to use and easy to solve. Moreover, we summarized a series of guidelines for design good image-based CAPTCHAs. We believe that the proposed DeRection and CONSCHEME schemes facilitate security against bots in online services without compromising user convenience. And the guidelines can be widely adopted by researchers on further researches on image-based CAPTCHAs.

**Acknowledgments.** The authors thank the reviewers for their careful reading of this paper and for their helpful and constructive comments. This project is supported by the National Natural Science Foundation of China (61472311) and the Fundamental Research Funds for the Central Universities.

## References

1. Von Ahn, L., Blum, M., Langford, J.: Telling humans and computers apart automatically. *Commun. ACM* **47**(2), 56–60 (2004)
2. Mori, G., Malik, J.: Recognizing objects in adversarial clutter: breaking a visual CAPTCHA. In: *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, p. I. IEEE (2003)

3. Yan, J., El Ahmad, A.S.: A low-cost attack on a Microsoft CAPTCHA. In: Proceedings of the 15th ACM Conference on Computer and Communications Security, pp. 543–554. ACM (2008)
4. Gao, H., Wang, W., Qi, J., Wang, X., Liu, X., Yan, J.: The robustness of hollow CAPTCHAs. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, pp. 1075–1086. ACM (2013)
5. Goodfellow, I.J., Bulatov, Y., Ibarz, J., Arnoud, S., Shet, V.: Multi-digit number recognition from street view imagery using deep convolutional neural networks. arXiv preprint [arXiv:1312.6082](https://arxiv.org/abs/1312.6082) (2013)
6. Karthik, C.P., Recasens, R.A.: Breaking Microsoft’s CAPTCHA. Technical report (2015)
7. Yan, J.: A simple generic attack on text CAPTCHAs (2016)
8. Chew, M., Tygar, J.D.: Image recognition CAPTCHAs. In: Zhang, K., Zheng, Y. (eds.) ISC 2004. LNCS, vol. 3225, pp. 268–279. Springer, Heidelberg (2004). [https://doi.org/10.1007/978-3-540-30144-8\\_23](https://doi.org/10.1007/978-3-540-30144-8_23)
9. Datta, R., Li, J., Wang, J.Z.: Imagination: a robust image-based CAPTCHA generation system. In: Proceedings of the 13th Annual ACM International Conference on Multimedia, pp. 331–334. ACM (2005)
10. Zhu, B.B., Yan, J., Li, Q., Yang, C., Liu, J., Xu, N., Yi, M., Cai, K.: Attacks and design of image recognition CAPTCHAs. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, pp. 187–200. ACM (2010)
11. Elson, J., Douceur, J.R., Howell, J., Saul, J.: Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In: ACM Conference on Computer and Communications Security, vol. 7, pp. 366–374. Citeseer (2007)
12. Vikram, S., Fan, Y., Gu, G.: SEMAGE: a new image-based two-factor CAPTCHA. In: Proceedings of the 27th Annual Computer Security Applications Conference, pp. 237–246. ACM (2011)
13. China Railway Customer Service Website. 12306 CAPTCHA (2017). <http://www.12306.cn/mormhweb/>
14. Gossweiler, R., Kamvar, M., Baluja, S.: What’s up CAPTCHA? A CAPTCHA based on image orientation. In: Proceedings of the 18th International Conference on World Wide Web, pp. 841–850. ACM (2009)
15. Rui, Y., Liu, Z.: Artificial: automated reverse turing test using facial features. *Multimedia Syst.* **9**(6), 493–502 (2004)
16. Li, Q.: A computer vision attack on the artificial CAPTCHA. *Multimedia Tools Appl.* **74**(13), 4583–4597 (2015)
17. Goswami, G., Powell, B.M., Vatsa, M., Singh, R., Noore, A.: FR-CAPTCHA: CAPTCHA based on recognizing human faces. *PloS ONE* **9**(4), e91708 (2014)
18. Goswami, G., Powell, B.M., Vatsa, M., Singh, R., Noore, A.: FaceDCAPTCHA: face detection based color image CAPTCHA. *Future Gener. Comput. Syst.* **31**, 59–68 (2014)
19. Gao, H., Lei, L., Zhou, X., Li, J., Liu, X.: The robustness of face-based CAPTCHAs. In: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), pp. 2248–2255. IEEE (2015)
20. Zulpe, N., Pawar, V.: GLCM textural features for brain tumor classification. *IJCSI Int. J. Comput. Sci. Issues* **9**(3), 354–359 (2012)
21. Raj, A., Jain, A., Pahwa, T., Jain, A.: Picture CAPTCHAs with sequencing: their types and analysis. *Int. J. of Digit. Soc.* **1**(3), 208–220 (2010)