STE3D-CAP: Stereoscopic 3D CAPTCHA

Willy Susilo[1,⋆], Yang-Wai Chow[2], and Hua-Yu Zhou[2]

[1] Centre for Computer and Information Security Research
[2] Centre for Multimedia and Information Processing
School of Computer Science and Software Engineering
University of Wollongong, Australia
{wsusilo,caseyc,hz285}@uow.edu.au

Abstract. We present STE3D-CAP (pronounced as "steed-cap" /'stidkæp/)[1], a text-based CAPTCHA that is built from stereoscopic 3D images. This is a completely new direction in CAPTCHA techniques. Our idea is to incorporate stereoscopic 3D images in order to present the CAPTCHA challenge in 3D, which will be easy for humans to read (as the text stands out in the 3D scene) but hard for computers. The main idea is to produce a stereo pair, two images of the distorted 3D text objects generated from two different camera/eye viewpoints, that are presented to a human user's left and right eyes, respectively. When the two images are supplied to hardware capable of displaying stereoscopic 3D images, the resulting CAPTCHA can easily be solved by humans, as the text will appear to stand out from the rest of the scene, but computers will not be able to solve them easily. As per the usual practice, the text in the produced images will be distorted (e.g. translated, scaled, warped) and overlapped but additionally the depth of the 3D text objects in the stereoscopic images will add a degree of complexity to the CAPTCHA and make it harder for CAPTCHA attacks (due to positive and negative parallax in the stereo pair). We demonstrate that the existing attacks on STE3D-CAP will fail with an overwhelming probability and that we can increase our CAPTCHA's resistance to segmentation attacks whilst maintaining usability. We also note that our technique is applicable to other stereoscopic approaches, such as anaglyph.

1 Introduction

The invention of CAPTCHAs (Completely Automated Public Turing test to tell Computers and Humans Apart)[2] was put forth by von Ahn et al. in 2003 [26]. CAPTCHAs are designed to be simple problems that can be quickly solved by humans, but are difficult for computers to solve. After von Ahn et al.'s seminal work, hundreds of design variants have appeared either in practice or in the literature. CAPTCHAs have quickly gained popularity over the past few years, since they are used to prevent exploitations by bots and automated scripts in public web services, which are rapidly increasing. Essentially, CAPTCHAs are challenge response tests that have become almost ubiquitous

[⋆] This work is supported by ARC Future Fellowship FT0991397.

[1] The name is inspired by a working mount (horse) especially for warfare.

[2] The term CAPTCHAs have also been known as Human Interaction Proofs (HIPs) [6].

S.-H. Heng, R.N. Wright, and B.-M. Goi (Eds.): CANS 2010, LNCS 6467, pp. 221–240, 2010.

on the World Wide Web to determine whether a user is a human or a computer. Using CAPTCHAs, services can distinguish legitimate users from computer bots while requiring minimal effort by the human user. Many companies currently employ the use of CAPTCHAs to protect their services against email spam, as well as to prevent fraud and denial of service attacks in online registrations, ticket/event reservations, online voting, chat rooms, weblogs, etc. [8].

To date, there exist three main types of CAPTCHAs: [30]

- Text-based CAPTCHAs: typically obtained by selecting a sequence of letters, rendering them, distorting the image and adding some noise;
- Image-based CAPTCHAs: typically ask users to conduct an image recognition task; and
- Sound-based CAPTCHAs (or audio CAPTCHAs): typically require users to solve a speech recognition task.

Among the three families, text-based CAPTCHAs are the most popular since they are simple, small and easy to design and implement. Therefore, they have been widely used in major web sites such as Google, Yahoo and Microsoft, since they are very intuitive to users world-wide, in addition to having good potential to provide strong security [30]. Theoretically, challenges as short as five characters are robust against random guessing[3], namely $62^5 \approx 912$ million possible five-character challenges that comprising case-insensitive letters and digits. Nevertheless, computing efforts, such as Optical Character Recognitions (OCR) or segmentation techniques, have been found very successful to achieve human-like accuracy [10,29,6]. Generally, text-based CAPTCHAs have universally suffered from a property that making them hard for computers also implies making them hard for humans [10].

Image-based CAPTCHAs were first described using labelled photographs by Chew and Tygar [9], which rely on Google Image Search [15]. This work was known to be unsuccessful due to Google's method of inferring photo contents based on surrounding descriptive text [10]. A more recent example in this family is Asirra [10], which is an image-based CAPTCHAs proposed in ACM CCS 2007 that uses images from Petfinder.com database. The security of Asirra relies on the problem of distinguishing images of cats and dogs, which is a task that is trivial for humans. Unfortunately, Golle demonstrated an attack based on machine learning to produce a classifier with 82.7% accuracy in telling apart the images of cats and dogs used in Asirra [13].

Audio CAPTCHAs were introduced to provide an alternative for those who are unable to use visual CAPTCHAs. Nevertheless, a recent study by Bigham and Cavender demonstrated that existing audio CAPTCHAs are clearly more difficult and time-consuming to complete as compared to visual CAPTCHAs for both blind and sighted users [3]. They also questioned how audio CAPTCHAs could be created that are easier for humans to solve while still addressing the improved automatic techniques for defeating them and posed it as an open problem for future research [3].

Elson et al. [10] classified CAPTCHAs into two different classes. In a Class I CAPTCHA, a secret value, which is merely a random number, is fed into a publicly

[3] We assume that random guessing is taken over upper and lowercase letters plus the digits.

known algorithm to produce a challenge, which is analogous to a public-key cryptosystem. In a Class II CAPTCHA, two inputs are required namely a secret value and a secret high-entropy database, which is somewhat analogous to a one-time-pad cryptosystem. One of the challenges in building a secure Class II CAPTCHA is populating the database with a sufficiently large set of classified and high-entropy entries [10].

Combining the two classifications above, it is clear that the most desirable way to construct CAPTCHAs is to employ Class I text-based CAPTCHAs, as they are small and easy to design and implement. The key challenge is how to enlarge the gap between human and non-human success rates whereas the resulting CAPTCHAs will be tolerated by users. Elson et al. [10] criticized text-based CAPTCHAs and commented that they must be intolerable to users or else the resulting CAPTCHAs would be too easy to break. In addition, Jakobsson further argues that in its current incarnation, CAPTCHAs may be nearing the end of its useful life [16]. He also contends that the current trend is such that strengthening CAPTCHAs to withstand increasingly powerful automated attacks also results in them becoming increasingly difficult for human users to solve. This will hurt user tolerance and at some point this trend will simply make CAPTCHAs too hard for people to use [16]. Therefore, it is an interesting open question to produce a Class I text-based CAPTCHA.

Our Contributions. In this paper, we present STE3D-CAP, a text-based CAPTCHA that is built from stereoscopic 3D images, as shown in Figure 1.

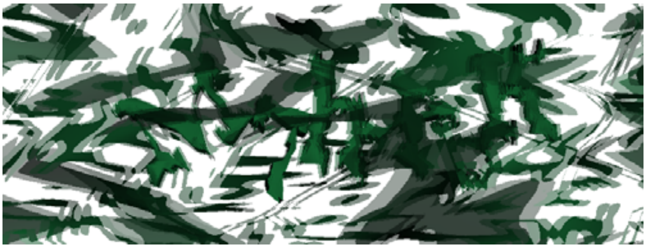


Fig. 1. STE3D-CAP without appropriate stereoscopic viewing equipment

STE3D-CAP is easy for human users, as it can be solved by humans who are equipped with an "appropriate" 3D visualization device (eg. Stereoscopic 3D display and glasses). The stereoscopic 3D images that represent left and right views are rendered by a graphics system on the server side and the resulting images are sent to the client system. The client system's hardware is therefore only provided with the two images and has to display these images to the user in stereo. We implemented our STE3D-CAP technique using an NVIDIA 3D Vision kit on a compatible NVIDIA graphics card and an Alienware^{TM}OptX AW2310, 3D capable, 120Hz monitor. We observe that all the known attacks, such as segmentation attacks [29] or pattern recognition analysis, will not be successful in analyzing STE3D-CAP.

The main drawback of STE3D-CAP is that it relies on the required stereoscopic display hardware that the user must own. Nevertheless, with the recent surge in the popularity of 3D movies and 3D games, LCD monitors and 3D TVs capable of displaying stereoscopic 3D images are becoming more and more common place nowadays (such as [12]). Further evidence of this can also be observed from the fact that the cost of 3D capable devices have become significantly cheaper in the past few years[4]. Our goal is to exploit advances in the lastest technology in order to breathe new life into current CAPTCHA techniques which are susceptible to novel attacks, by strengthening text-based CAPTCHAs against such attacks whilst maintaining human usability.

1.1 Related Work

After the seminal concept of a CAPTCHA was introduced by von Ahn [26], many design variations of CAPTCHAs have been proposed and used. The text-based CAPTCHAs are very popular due to its simplicity. Unfortunately, computer attacks have been found against most of the existing text-based CAPTCHAs. Simard et al. demonstrated the use of Optical Character Recognition (OCR) to obtain human-like accuracy, as long as the letters can be segmented reliably [24]. Mori and Malik showed that von Ahn et al.'s original GIMPY CAPTCHA [25] can be solved automatically 92% of the time [19].

Reading text-based CAPTCHA challenges typically consists of a segmentation challenge and a recognition challenge [6]. The segmentation challenge involves the identification of character locations in the right order, whereas the recognition challenge is in recognizing individual characters. The difficulty of these challenges can be increased using various techniques like cluttering the foreground and background, distorting individual characters, etc. Research has shown that computers are extremely successful in recognizing individual characters, even characters that are highly distorted [7,29]. This therefore suggests that the challenge for text-based CAPTCHAs is to design CAPTCHAs that are resistant to segmentation yet human readable.

In 2005, Microsoft published a text-based CAPTCHA, which was designed to be segmentation resistant [6]. The resulting CAPTCHA had been widely deployed in many Microsoft's online services, such as Hotmail, MSN and Windows Live for years. Unfortunately, a low-cost attack with success rate higher than 90% was developed by Yan and Ahmad [29], which demonstrated that this carefully designed CAPTCHA is vulnerable to novel, but simple attacks.

Usability issues of text-based CAPTCHAs have also been investigated by Yan and Ahmad [30]. The four common distortion on text-based CAPTCHAs that will not make them difficult for human users to recognize them are as follows [30].

- Translation: characters can be moved up or down and left or right by an amount;
- Rotation: characters can be turned either in the clockwise or counter clockwise direction;
- Scaling: characters can be stretched or compressed in either the x-direction or y-direction;
- Warp: elastic deformation of CAPTCHA images at different scales.

The length of text strings used in CAPTCHAs also plays an important role to their security. Some schemes choose to use a fixed length, and they turned out to be insecure.

[4] A set of NVIDIA [21] capable devices nowadays are around US$ 500, which is significantly lower compared to the cost last year.

For example, the Microsoft's CAPTCHAs use 8 characters in their challenge [6]. It turns out that the segmentation attack can be done easier knowing this fact [29]. On the other hand, Google's CAPTCHAs incorporate a different number of characters in each challenge, although their security has not been rigorously tested [30].

The use of color also plays an important role in CAPTCHA design. In terms of usability, color is good for drawing a user's visual attention. The incorporation of color can also make the CAPTCHA challenge more appealing, it can potentially aid in the recognition and comprehension of the text and it potentially makes the CAPTCHA seem less intrusive in the context of the application [6,30]. Color schemes can potentially increase a CAPTCHA's security against some attacks, e.g. OCR software attacks which are poor at recognising text in colored images. However as highlighted in [30], if used inappropriately color may add little or nothing to the security of a CAPTCHA, but at the same time can significantly reduce the usability of the CAPTCHA. The misuse of color can make the text in a CAPTCHA very difficult to read even for people with normal vision. In fact, color can have a negative effect on the security of a CAPTCHA, as it might make it easy for a computer to distinguish the important text from the background/foreground clutter, etc. As such, care must be taken when using color in CAPTCHAs.

The idea of representing text-based CAPTCHAs with 3D text objects has been used in [22]. Their CAPTCHA, which is called Teabag 3D, is obtained by making a picture in 3D with text objects. Unfortunately, it can be seen clearly that the location of the text objects can easily be distinguished due to the somewhat regular pattern in the surrounding regions, and therefore these CAPTCHAs would be breakable by using something like a simple segmentation technique [29].

There have also been a number of recent approaches to representing CAPTCHA challenge based on 3D models. A 3D object matching CAPTCHA challenge was introduced in [31]. This is an image-based CAPTCHA approach where users are presented with images of 3D models, which are rendered from different angles using Lambertian lighting, and are required to select matching 3D models from a set of images. However as pointed out in [23], this approach is susceptible to attacks using basic computer vision techniques. Ross et al. [23] introduced 'Sketcha', an image-based CAPTCHA approach based on images of line drawings which are rendered from 3D models. In Sketcha, users are presented with a set of randomly orientated line drawings, and are required to rotate each image until all the images are upright. Mitra et al. [18] proposed a technique of generating 'emergence images' by rendering extremely abstract representations of 3D objects models placed in a 3D environment. Their approach is based on 'emergence' which is the unique human ability to perceive objects from seemingly meaningless patches in an image. However when the image is viewed as a whole, a human can perceive the form of the main subject which pops out from the clutter [18].

The idea of CAPTCHAs has also been turned into another useful purpose, namely to help to digitize old printed material from books that computerized optical character recognition failed to recognize. This technique is known as reCAPTCHA [27]. Interestingly, it has reported that this method can transcribe text with a word accuracy exceeding 99.99% matching the guarantee of professional human transcriber [27].

2 CAPTCHA Revisited

Formally, CAPTCHAs have been defined by von Ahn et al. [26] as follows.

> *"A CAPTCHA is a cryptographic protocol whose underlying hardness assumption is based on an Artificial Intelligence problem."*

When the underlying Artificial Intelligence (AI) problem is useful, a CAPTCHA implies an important situation, namely either the CAPTCHA is broken and there is a way to differentiate humans from computers, or the CAPTCHA is broken and a useful AI problem is solved [26].

2.1 Definitions and Notation

The following definitions and notation are adapted and simplified from [26]. Intuitively, a CAPTCHA is a test V where most humans have success close to 1, while it is hard to write a computer program that has overwhelming probability of success over V. That means, any program that has high probability of success over V can be used to solve a hard AI problem. In the following, let $\mathcal{C}$ be a probability distribution. If $P(\cdot)$ is a probabilistic program, let $P_r(\cdot)$ denote the deterministic program that results when P uses random coins r.

Definition 1. [26] A test V is said to be (α, β)-human executable if at least an α portion of the human population has success probability greater than β over V.
Definition 2. [26] An *AI problem* is a triple $\mathcal{P} = (S, D, f)$ where S is a set of problem instances, D is a probability distribution over S and $f : S \rightarrow \{0,1\}^*$ answers the problem instances. Let $\delta \in (0,1]$. For $\alpha > 0$ fraction of the humans H, we require $Pr_{x \leftarrow D}\left[H(x) = f(x)\right] > \delta$.
Definition 3. [26] An AI problem $\mathcal{P}$ is said to be (ψ, τ)-solved if there exists a program $\mathcal{A}$ that runs in time for at most τ on any input from S, such that

$$Pr_{x \leftarrow D, r}\left[\mathcal{A}_r(x) = f(x)\right] \geq \psi.$$

Definition 4. [26] An (α, β, η)-CAPTCHA is a test V that is (α, β)-human executable and if there exists $\mathcal{B}$ that has success probability greater than η over V to solve a (ψ, τ)-hard AI problem $\mathcal{P}$, then $\mathcal{B}$ is a (ψ, τ) solution to $\mathcal{P}$.
Definition 5. An (α, β, η)-CAPTCHA is *secure* iff there exists no program $\mathcal{B}$ such that

$$Pr_{x \leftarrow D, r}\left[\mathcal{B}_r(x) = f(x)\right] \geq \eta$$

for the underlying AI problem $\mathcal{P}$.

3 Review on 3D Stereoscopy

Stereoscopy relates to the perception of depth in the human visual system that arises from the horizontal separation of our eyes by the interocular distance (distance between the eyes) [5]. In real life, this results in our visual cortex being presented with two slightly different views of the world. When viewing a 3D scene, binocular disparity refers to the difference in the images that are projected onto the left and right eye retinas, then onto the visual cortex [17]. The human visual system perceives the sensation of depth through a process known as stereopsis, by using binocular disparity to obtain depth cues from the 2D images that are projected onto the retinas.

Though a variety of 3D display devices have been developed over the years, in this study we only concern ourselves with stereo pair based technologies. A stereo pair is a set of two images, one created for the left eye and the other for the right eye. Stereo pair based technologies simulate binocular disparity based on the presentation of the different images to each of the viewer's eyes independently [17]. By synthetically creating and presenting two correctly generated images of the left and right views of a scene, the visual cortex will fuse the images as it does in normal viewing to give rise to the sense of depth [4].

However, it is important to note that there are a variety of other depth cues that the human brain can infer from a 2D image, e.g. perspective (objects further away from the viewer look smaller), occlusion (where closer objects block objects that are further away), shading and shadows, etc. Depth cues are generally additive, in other words the more the better. Therefore, it is important for the depth cues to be consistent and to avoid conflicting depth cues in the generation of the stereo pair [5,17].

3.1 Stereo Pair Generation

To generate the stereo pair, two camera/eye viewpoints are used to create the left and right images by horizontally displacing the cameras by an appropriate eye separation. Stereo pairs that are not created correctly will make viewing very uncomfortable or the brain might not even fuse the images at all resulting in the viewer seeing two separate images. Therefore a number of factors have to be considered in practice when attempting to generate the stereo pair, so as to not overwhelm the visual system. For example, eye separation that is set to be too large results in a condition known as hyperstereo, and although this exaggerates the stereo effect the brain might find it hard to fuse the images. Another consideration particularly relevant to our study where we attempt to clutter the CAPTCHA with noise, is that if the frequency of the noise is too high, there will essentially be little matching visual information for the brain to resolve between the images in the stereo pair [4].

Parallax refers to the signed distance on the projection plane between the projected positions of a point in the stereo pair. Parallax is a function of the depth of a point in eye space [11]. A point in space that is projected onto the projection plane can be classified as having one of three relationships: zero parallax, positive parallax and negative parallax. Note that these refer to the horizontal distance on the projection plane as the vertical parallax should always be zero, otherwise the user will generally suffer from uncomfortable physical symptoms from misaligned cameras. While the amount that can be tolerated will vary from viewer to viewer, adverse side effects include headaches, eye strain, and in severe cases even nausea [17].

Zero parallax occurs when the projected point is on the projection plane. The pixel position of the projected point is exactly the same position on both left and right images. This is depicted in Figure 2(a) from a top-down view. As illustrated in Figure 2(b), positive parallax occurs when the projected point is located behind the projection plane.

In this case, the pixel position of the projected point is located on the right in the right image and on the left in the left image. To the observer, the point appears at a depth ‘into’ the screen. The maximum possible positive parallax is equal to the eye separation and arises when the point is located at infinity. Figure 2(c) depicts negative parallax which occurs when the projected point is located in front of the projection plane. When this happens, the pixel position of the projected point is located of the left in the right image and on the right in the left image. The observer perceives the point as coming ‘out’ of the screen [11].

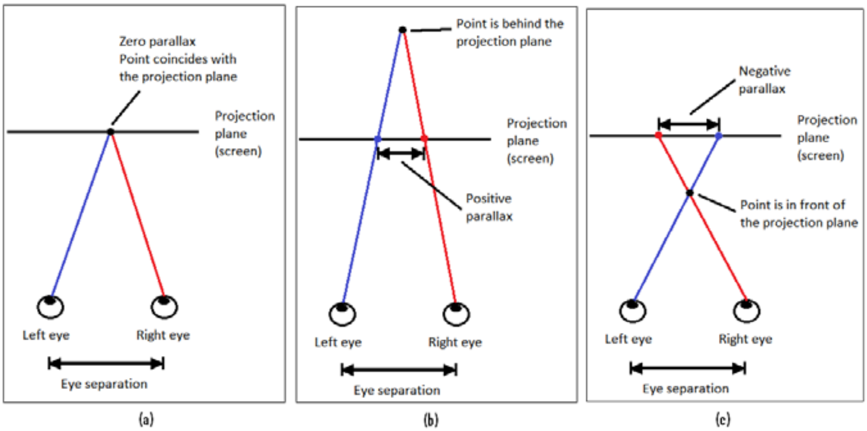

Zero parallax
Point coincides with the projection plane
Projection plane (screen)
Left eye
Right eye
Eye separation
(a)
Point is behind the projection plane
Projection plane (screen)
Positive parallax
Left eye
Right eye
Eye separation
(b)
Negative parallax
Projection plane (screen)
Point is in front of the projection plane
Left eye
Right eye
Eye separation
(c)

Fig. 2. Parallax

3.2 Stereoscopic 3D Display Technologies

Stereo pair based technologies require a method of ensuring that the left eye only sees the left eye’s image and the right eye only sees the image for the right eye. There are a variety of methods that have been developed to achieve this. Here we highlight a number of stereo pair based technologies of relevance to our study, a comprehensive overview of 3D display technologies can be found in [17].

A common technique used in a number of stereoscopic display devices is to alternate the display of left and right views on a single display. These techniques require the viewer to use equipment such as viewing glasses to prevent the left eye from seeing the right view and vice versa. These can either be active or passive viewing glasses. Active systems employ blocking lenses which synchronize with the display to alternately cause the left and right eye lenses to become opaque, thereby blocking the respective eye’s view. On passive systems, the display device produces polarized light where left and right eye images are polarized in orthogonal directions. The viewing glasses for these systems have similarly polarized lenses for each eye that only allows through light that is polarized along an axis parallel to the respective eye [17]. Either of these systems is suitable for STE3D-CAP.

Unlike the previous approaches, in the anaglyph method the viewer is not presented with alternate left and right views independently. Rather, both views are presented to the viewer simultaneously on a single image, where the left and right eye views are color encoded using two colors. The viewer has to wear glasses with red/green filters (or similar red/cyan, red/blue, etc.) which filter out colors with certain frequencies for each eye [5]. Some of the drawbacks of this approach include the lack of representation of the full range of color, and this approach typically suffers from a lot of cross-talk (this means that a portion of the view intended for one eye is visible to the other eye, resulting in what is known as ‘ghosting’). However, this presents a low-cost solution.

There are other stereoscopic display technologies that do not require special viewing glasses, these devices are called autostereoscopic. These devices use a variety of approaches such as lenticular sheets and parallax barriers, which are designed to focus and redirect light to different viewing regions causing the viewer to perceive a different image for each eye. Such display devices are increasingly gaining popularity and a number of companies have recently announced the launching of their glass-free 3D devices [20]. STE3D-CAP can be used on any of these systems as long as they are stereo pair based.

4 Design and Implementation of STE3D-CAP

In this section, we will describe the design of our new CAPTCHA, stereoscopic 3D CAPTCHA, or STE3D-CAP for short. To describe it precisely, we will proceed with presenting the underlying AI problem and then commence with the detail of our design and implementation.

STE3D-CAP is a CAPTCHA that are built using the stereoscopic 3D technology. The idea is to present CAPTCHA text as 3D objects that will be easily identified by humans (who are equipped with the proper equipment), but they are hard to be analyzed by machines. For the implementation of our idea, we use existing NVIDIA technology that requires us to supply two images (2D) that represent the 3D object for the left and right eyes, resp. Then, the NVIDIA card will render the two images and present the 3D objects.

STE3D-CAP have several attractive features:

- Humans can solve it quickly (§4.1).
- Computers cannot solve it easily (§4.1).
- STE3D-CAP is easy to generate as it is a text-based CAPTCHA.
- STE3D-CAP uses the latest technology.
- STE3D-CAP uses 3D, and hence, more noise can be added in the 3D scene while the resulting CAPTCHA is still usable.
- STE3D-CAP is a variable length CAPTCHA, which are more difficult to defeat.
- STE3D-CAP is built in a 3D environment, and therefore more distortion can be added to the CAPTCHA (eg. negative/positive parallax).

Nevertheless, STE3D-CAP has several disadvantages:

- STE3D-CAP requires special-type of equipment, namely equipment to display 3D[5].

[5] Even though specialized 3D displays are preferable, for practical applications, the anaglyph approach can be used. Moreover, the requirement for dedicated hardware in new CAPTCHA techniques has never been an issue (e.g. physical CAPTCHAs) [14].

- STE3D-CAP challenges may require more screen space than traditional text-based CAPTCHAs.
- Like virtually all other CAPTCHAs, STE3D-CAP is not accessible to those with visual impairments including those who are stereo-blind.

It should be noted that although STE3D-CAP is currently text-based, it can easily be extended to use models of other 3D objects instead of only 3D text. Nevertheless, we choose to focus on using a text-based CAPTCHA approach for reasons outlined in [8]; namely, that text characters were designed by humans for humans, humans have been trained to recognize characters since childhood, text-based CAPTCHA tasks are easily understood by users without much instruction and that each character has a corresponding key on the keyboard which gives rise to many possible input combinations.

An interesting approach to confuse segmentation attacks would be to randomly interleave 3D models among the 3D text. A human user would clearly be able to distinguish between the text and the random objects. However, it would make segmentation and recognition harder for a computer which cannot easily differentiate between the text and objects.

In addition, we use random character strings in STE3D-CAP rather than dictionary words. While the use of words from a dictionary will probably make the text in STE3D-CAP easier to perceive and has implications on the security, we avoid this as it unfairly disadvantages people unfamiliar with the chosen language.

4.1 New AI Problem Family

In this section, we introduce a family of AI problems that will be used to build our CAPTCHA, STE3D-CAP. An image is defined as an $h \times w$ matrix (where h stands for height and w stands for width), whose entries are pixels. A pixel is defined as a triplet (R, G, B), where $0 \leq R, G, B \leq M$, for a constant M.

Let $\mathcal{I}_{2d}$ be a distribution on images, $\mathcal{I}_{3d}$ be a distribution on stereoscopic 3D images and $\mathcal{T}$ be a distribution on stereoscopic image transformations, that include rotation, scaling, translation and warp. Let Ω be a distribution on noise frequency, and Υ be a distribution on erosion factors. Let $\mathcal{C} : \mathcal{I}_{3d} \times \Omega \times \Upsilon \rightarrow \mathcal{I}_{3d}$ be a distribution of clutter functions. A clutter function is a function that accepts a 3D image, a noise frequency $\in \Omega$ and an erosion factor $\in \Upsilon$ and outputs a cluttered 3D image. Let $|A|$ denote the cardinality of A.

Let $\Delta : |\mathcal{I}_{3d}| \rightarrow \mathcal{I}_{3d}$ be a lookup function that maps an index in $|\mathcal{I}_{3d}|$ and output a stereoscopic 3D image in $\mathcal{I}_{3d}$. Let $\vartheta : \mathcal{I}_{2d} \times \mathcal{I}_{2d} \rightarrow \mathcal{I}_{3d}$ be a function that maps two images (left and right images) to a stereoscopic image. Subsequently, let $\vartheta^{-1} : \mathcal{I}_{3d} \rightarrow \mathcal{I}_{2d} \times \mathcal{I}_{2d}$ be a function that given a stereoscopic image, outputs two images that represent left and right images, resp. Subsequently, we also assume that ϑ is publicly available. For simplicity, we denote the left and right images with superscript L and R, resp.

For clarify, for the rest of this paper, we will use **Roman boldface** characters to denote elements of $\mathcal{I}_{3d}$, while Sans Serif characters to denote elements of $\mathcal{I}_{2d}$.

Problem Family ($\mathcal{P}_{\textsf{STE3D-CAP}}$).
Consider the following experiment.

1. Randomly select $i \in |\mathcal{I}_{3d}|$.
2. Compute $\mathbf{i} \leftarrow \Delta(i)$.
3. Select a transformation $t \leftarrow \mathcal{T}$.
4. Compute $\bar{\mathbf{i}} \leftarrow t(\mathbf{i})$.
5. Select a clutter function $c \leftarrow \mathcal{C}$.
6. Compute $\mathbf{j} \leftarrow c(\bar{\mathbf{i}}, \omega, \upsilon)$, where $\omega \in \Omega$ and $\upsilon \in \Upsilon$ are selected randomly.
7. Output $\vartheta^{-1}(\mathbf{j})$.

The output of the experiment is $(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}}) \leftarrow \vartheta^{-1}(\mathbf{j})$, where $(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}}) \in \mathcal{I}_{2d} \times \mathcal{I}_{2d}$.

$\mathcal{P}_{\textsf{STE3D-CAP}}$ is to write a program that takes $(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}}) \in \mathcal{I}_{2d} \times \mathcal{I}_{2d}$ as input and outputs $i \in |\mathcal{I}_{3d}|$, assuming the program has precise knowledge of $\mathcal{T}$, $\mathcal{C}$ and $\mathcal{I}_{2d}$.

More formally, let

$$S_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}} = \{\vartheta^{-1}\left(c\left(t(\Delta(i)), \omega, \upsilon\right)\right) = (\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}}) : t \leftarrow \mathcal{T}, \\ c \leftarrow \mathcal{C}, \omega \in \Omega, \upsilon \in \Upsilon, (\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}}) \in \mathcal{I}_{2d} \times \mathcal{I}_{2d}\}$$

Let $D_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}}$ be the distribution of $S_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}}$ that are obtained from executing the above experiment, and

$$f_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}} : S_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}} \rightarrow |\mathcal{I}_{3d}|$$

such that $f_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}} = i$, $i \in |\mathcal{I}_{3d}|$. Then,

$$\mathcal{P}_{\textsf{STE3D-CAP}} = (S_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}}, D_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}}, f_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}}).$$

Hard Problem in $\mathcal{P}_{\textsf{STE3D-CAP}}$.

We believe that $\mathcal{P}_{\textsf{STE3D-CAP}}$ contains a hard problem. Given $\mathcal{P}_{\textsf{STE3D-CAP}} = (S_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}}, D_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}}, f_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}})$, for any program $\mathcal{B}$,

$$Pr_{x \leftarrow D_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}}, r}\left[\mathcal{B}_r(x) = f(x)\right] < \eta.$$

Based on this hard problem, we can construct a secure (α, β, η)-CAPTCHA.

Theorem 1. A secure (α, β, η)-CAPTCHA can be constructed from $\mathcal{P}_{\textsf{STE3D-CAP}}$ as defined above.

Proof. We will provide the proof in two stages. First, we show that (α, β, η)-CAPTCHA is (α, β)-human executable. Then, we show that (α, β, η)-CAPTCHA is hard for a computer to solve. We also show an instantiation of our proof.

Given $\mathcal{P}_{\textsf{STE3D-CAP}}$, humans can get the instance $(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}}) \leftarrow \vartheta^{-1}(\mathbf{j})$, where $(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}}) \in \mathcal{I}_{2d} \times \mathcal{I}_{2d}$. Then, by executing $\vartheta(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}})$ humans can easily see the instance of the problem and output i. We should justify that in practice, $\vartheta(\cdot)$ is implemented in a 3D hardware capable of displaying 3D, such as an NVIDIA card [21]. Executing $\vartheta(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}})$ means that humans will be able to see the objects provided clearly using the required equipments (such as 3D glasses), and humans can output i easily. Hence, (α, β, η)-CAPTCHA is (α, β)-human executable.

However, given $\mathcal{P}_{\mathsf{STE3D\text{-}CAP}}$, machines cannot output i. Although $\vartheta(\mathsf{j}^{\mathsf{L}},\mathsf{j}^{\mathsf{R}})$ is available publicly, machines cannot "view" the 3D representation of i and hence, cannot output i easily. The best way to analyze the problem is by processing $(\mathsf{j}^{\mathsf{L}},\mathsf{j}^{\mathsf{R}})$ directly, which will not help machines to identify i. Hence, $Pr_{x \leftarrow D_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}},r}\,[\mathcal{B}_r(x) = f(x)] < \eta$, for any $\mathcal{B}$.

An in-depth security analysis on $\mathcal{P}_{\mathsf{STE3D\text{-}CAP}}$ will be provided in §5.

4.2 Design Principles of STE3D-CAP

1. Differences between left and right images Intuitively, the first design principle requires that the "difference" between the left and the right images must be sufficiently noisy to ensure that segmentation attacks will fail. The idea is to ensure that **j** will be clearly visible for humans to identify i, while machines observing $(\mathsf{j}^{\mathsf{L}},\mathsf{j}^{\mathsf{R}})$ cannot deduce i. Note that j^{L} is the 2D version of the image from the left eye's perspective and j^{R} is from the right eye's perspective. While the 3D version of i in both j^{L} and j^{R} must exist, the noise in both images can be made vary since the noise visible from the left eye maybe blocked and invisible from the right eye, and vice versa. Suppose we define that $\mathsf{j}^{\mathsf{L}} = \bar{t}(\mathsf{j}^{\mathsf{R}}) + \delta$, where $\bar{\mathcal{T}}$ is a distribution on 2D image transformation (eg. translation) and $\bar{t} \leftarrow \bar{\mathcal{T}}$. We require δ to be sufficiently noisy to deter against segmentation attacks. Formally, we have the following theorem.

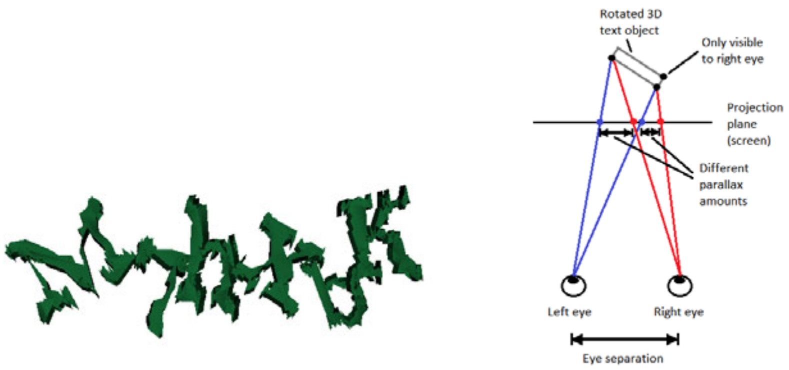
Theorem 2. An (α,β,η)-CAPTCHA constructed from $\mathcal{P}_{\mathsf{STE3D\text{-}CAP}}$ is secure against segmentation attacks, *iff* for $\delta = \mathsf{j}^{\mathsf{L}} - \bar{t}(\mathsf{j}^{\mathsf{R}})$, where $\bar{\mathcal{T}}$ is a distribution on 2D image transformation and $\bar{t} \leftarrow \bar{\mathcal{T}}$, we require that there exists no program $\mathcal{B}$ such that

$$Pr_{x \leftarrow D_{\mathsf{STE3D\text{-}CAP}_{\mathcal{I}_{2d},\mathcal{T},\mathcal{C}}},r}\,[\mathcal{B}_r(x) = i] \geq \eta.$$

2. Human Factors Although we vary a number of parameters in STE3D-CAP in order to make it less predictable and harder for computers to perform automated attacks, a number of human factors issues had to be kept in mind. This is because our aim is for a human user to be able to use STE3D-CAP comfortably.

In terms of stereoscopic viewing, it is generally easier on the eyes to view objects that are at screen depth or objects that appear into the screen, even though this does not result in the pop out of screen effect. Caution must be exercised for objects in front of the screen as parallax diverges quickly to negative infinity for objects closer to the eyes [11]. Focusing on objects that are positioned too close to the eyes forces the viewer's eyes to cross at a point in front of the screen, which can be very strenuous on the eyes. The focal length refers to the distance at which objects in the scene will appear to be at zero parallax. In general, objects should not be positioned closer than half the focal length, in other words negative parallax should not exceed the eye separation [4].

In addition, eye separation must also be kept within a reasonable range. Ideally, eye separation should be made as large as possible. This way the images for the left and right views will be rendered from very different angles and some information present in one image might be missing from the other and vice versa, as shown in Figure 3(b). Therefore, information will only be complete if both images are viewed together. However, it must be kept in mind that too large an eye separation will lead to hyperstereo


Rotated 3D text object
Only visible to right eye
Projection plane (screen)
Different parallax amounts
Left eye
Right eye
Eye separation

(a) Distorted 3D text

(b) Rotated 3D text object

Fig. 3.

which is uncomfortable for users, whereas a value which is too small results in hypostereo. In hypostereo, the user does not really perceive a 3D effect and furthermore the images in the stereo pair will not differ by much.

4.3 Implementation

We implemented STE3D-CAP by using a graphics system to render models of 3D text objects against foreground and background clutter. To develop and view STE3D-CAP, we used an NVIDIA 3D Vision kit with an Alienware^{TM}OptX AW2310 3D monitor. To avoid user eye strain, perspective camera properties, view frustums, etc. were set up using the general rules of thumb. For quantitative guidance, please refer to [4] and [11].

Customizable vertex and fragment shaders were used to give rise to random vertex perturbations and erosion effects. The vertices of the text models were randomly perturbed in 3D to distort the text. Figure 3(a) gives a depiction of this for the left eye. Please note that all STE3D-CAP images in this paper were generated using the same character string as shown in Figure 3(a) for comparison sake.

Individual 3D text objects also undergo random 3D transformations. Unlike conventional 2D CAPTCHAs, we can add more variation to the transformations as we are now dealing with 3D. Translation is not merely left/right and up/down but can also be varied with respect to depth from screen. Similarly rotation is not restricted to being clockwise and counter clockwise, as we can also rotate the 3D text objects about the vertical axis as well as the horizontal axis pointing to the right. Rotation must be limited to be within a certain range, otherwise the text may not be readable if slanted at angles which are too steep. We chose a conservative rotation range of between +/- 20 degrees for the rotation axes parallel to the projection plane and +/- 45 degrees clockwise/counter clockwise. By rotating the text objects the parallax of the projected points on the object will also be different in screen space. This is illustrated in Figure 3(b). We also randomly scale the 3D text objects to alter their size.

To increase the difficulty of segmentation attacks we adopted the "crowding characters together" method (letting characters touch or overlap with each other) which is suggested to be segmentation resistant [30], and we also attempted to clutter the scene with noise. Our implementation allows us to adjust the frequency of the noise. However, as noted in Section 3.1 high frequency noise makes it difficult for the brain to correlate matching visual information and subsequently makes it hard to fuse the stereo pair. Also, if the noise is too fine, it will be easy to differentiate the text from the noise by simply removing small individual clusters of noise. Furthermore, completely random noise that appears in one image but not the other will also be hard to fuse, and at the same time easy to filter out by simply finding the differences between the stereo pair.

Instead, in our implementation we use foreground and background surfaces with randomly perturbed vertices and eroded surface sections base on a 3D Perlin noise function. The surfaces' vertices were perturbed to avoid completely smooth surfaces which will be easy to filter out between stereo pair images. We can also adjust the scale and amount of erosion. In addition, we made the foreground clutter slightly translucent for usability reasons, so that the text behind it will not be completely obscured. In this manner, the clutter does not appear as random noise, but rather from the user's point of view looks like two eroded surfaces, and they can perceive the text amidst the surfaces. Figure 4[6] shows an example of such a stereo pair. It can be seen that one cannot use the images individually to complete the CAPTCHA challenge.

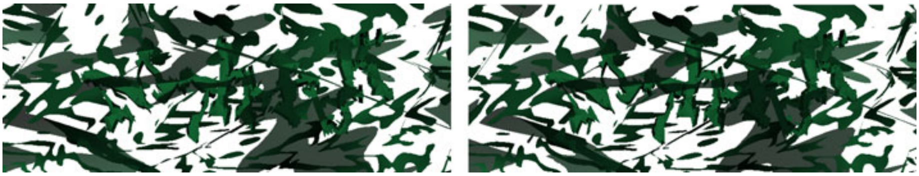


Fig. 4. Example of a STE3D-CAP stereo pair

At the moment, color is used in STE3D-CAP merely from a usability standpoint to improve the attractiveness of the CAPTCHA rather than for any particular security reason. This is because even though it will be much easier to see the text if it was highlighted with a different color from the clutter, this would also make it very easy to filter out the clutter by just separating the text based on color. Furthermore, if we introduced a lot of random colors to improve security, it would make the stereo pair very hard to fuse. Moreover, an automated attack could simply convert the images to greyscale and attempt to threshold the intensity levels rather than dealing with the color. The colors in STE3D-CAP were deliberately made to overlap with the clutter to make it harder for automated attacks whilst still being usable.

Despite STE3D-CAP being rendered in 3D, confusing character combinations still had to be avoided as highlighted in [30]. For example, a distorted 'vv' might look like a 'w', etc.

[6] Though we do not recommend this, it is possible to see the 3D text by crossing one's eyes à la magic eye images.

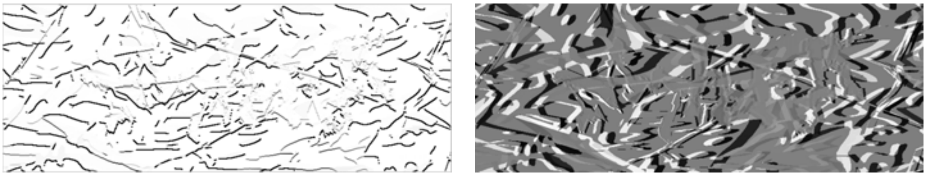
5 Security of STE3D-CAP

Before describing the security of STE3D-CAP, it is useful to review the threat model associated with CAPTCHAs, as CAPTCHAs are an unusual area of security where we are not trying to provide absolute assurances, but rather to merely slow down attackers. CAPTCHAs are considered to be "successful" if they force an automated attack to cost more than approximately 30 seconds worth of a human's time in part of the world where labor is cheap [10]. It is generally acceptable if CAPTCHAs can admit bots with less than 1/10,000 probability [25].

When considering the security of STE3D-CAP, we provide the adversary with $(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}})$ instead of $\mathbf{j}$, due to the following reason. First, although the input for humans is $\mathbf{j}$, machines cannot view 3D objects like humans. Therefore, it would be easier for machines to be provided with $(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}})$ instead. Second, the function φ is available publicly. This function will transform $\mathbf{j} \leftarrow \varphi(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}})$. This assumption is very reasonable as the implementation is on the client's system. Therefore, the machine adversary can also make use of this function whenever it is deemed necessary. Third, since the 3D view is generated by the client's machine, the two images $(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}})$ will need to be sent to the client's machine. Therefore, even though the view that is shown in the client's machine is $\mathbf{j}$, it is reasonable to assume that the adversary can capture both $(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}})$ by observing the TCP/IP packets.

5.1 Single Image Attacks

Any of the existing CAPTCHA attacks can be attempted on the left and right images of STE3D-CAP individually. However, unlike existing 2D text-based CAPTCHA approaches which cannot be overly cluttered in order to maintain usability (which makes them more susceptible to segmentation attacks), it is possible to increase the foreground and background clutter in STE3D-CAP. This will increase STE3D-CAP's security against segmentation attacks, while at the same time when STE3D-CAP is viewed in 3D, the viewer can still perceive the text and differentiate this against the clutter. Figure 5(a) was obtained by passing the left image through a Sobel edge detection filter. While it highlights the edges in the image, it does not give rise to much useful information.



(a) Edge detection image

(b) Difference image

Fig. 5.

5.2 2D Image Difference Attacks

Unlike the existing attack models in the literature, we introduce a new type of attack namely 2D image difference attacks. In this type of attack, an adversary who is given a pair of 2D images, $(\mathsf{j}^\mathsf{L}, \mathsf{j}^\mathsf{R})$ will first find the difference between these two. This relies on the fact that $\mathsf{j}^\mathsf{L} = \bar{t}(\mathsf{j}^\mathsf{R}) + \delta$, where $\bar{\mathcal{T}}$ is a distribution on 2D image transformation (eg. translation) and $\bar{t} \leftarrow \bar{\mathcal{T}}$. The adversary will need to find the appropriate $\bar{t}$[7]. Then, subsequently, the adversary will try to eliminate δ, which is $\delta = \mathsf{j}^\mathsf{L} - \bar{t}(\mathsf{j}^\mathsf{R})$. By eliminating δ, the leftover image will then be analyzed using the existing segmentation techniques, such as [29], to identify the segments and break the CAPTCHAs. To demonstrate this, Figure 5(b) depicts the difference image[8] between left and right views. Sections in white are in the left image but not in the right, whereas black represents sections in the right image but not in the left, and grey shows overlapping sections. It can be seen that little useful information can be gathered from the image. Figure 6 is the anaglyph version of Figure 1. For usability and ease of use of STE3D-CAP, one can view Figure 6 using a low cost red-cyan anaglyph glasses. Note however that the anaglyph version will look slightly different compared to when using appropriate stereoscopic devices, as one can see greater variation in the depth of the characters using the latter approach.

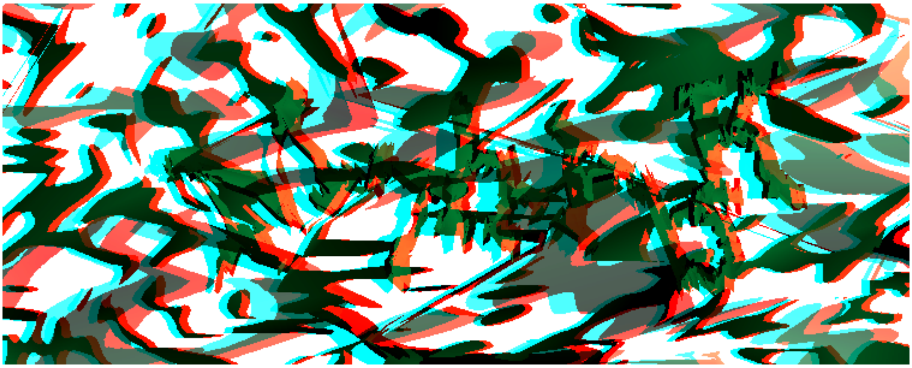


Fig. 6. Anaglyph

Theorem 3. An (α, β, η)-CAPTCHA constructed from $\mathcal{P}_{\mathsf{STE3D\text{-}CAP}}$ as defined above is secure against 2D image differences attacks.

Proof. Our (α, β, η)-CAPTCHA has been designed according to the first design principle in §4.2. This means δ has been chosen such that it will be sufficient to deter against segmentation attacks. Hence, an adversary launching 2D image differences attacks will end up with a noisy 2D image that will not represent mere the object i. Therefore, image segmentation attacks on the newly developed image will not be able to extract i.

[7] We note that this action can be done trivially by comparing the two images (j^L and j^R).

[8] This image is obtained by taking the left image minus the right image, and scaled between black and white.

Subsequently, we will also obtain the following theorem.

Theorem 4. An (α, β, η)-CAPTCHA constructed from $\mathcal{P}_{\mathsf{STE3D\text{-}CAP}}$ as defined above is secure against segmentation attacks.

5.3 Brute Force Attacks

A straightforward attack on STE3D-CAP is brute force. In this attack, the adversary will just provide a random solution to challenges until one succeeds. This means, given a STE3D-CAP challenge, $(\mathsf{j}^{\mathsf{L}}, \mathsf{j}^{\mathsf{R}})$, the adversary will find a random solution for it. Note that STE3D-CAP are variable length CAPTCHAs. Hence, the adversary has no knowledge on the length of the challenge. Suppose the length of the challenge is λ, and there are 62 possible characters comprising lower and upper case letters and digits. Then, the chance of successful brute force attacks is $\frac{1}{62^{\lambda}}$. In practice, this chance can be considered as negligible, especially when CAPTCHAs are combined with techniques such as token bucket algorithms [10] to combat denial-of-service attacks.

Theorem 5. An (α, β, η)-CAPTCHA constructed from $\mathcal{P}_{\mathsf{STE3D\text{-}CAP}}$ as defined above is secure against brute force attacks.

5.4 3D Reconstruction Attacks

While one may be able to approximate the reconstruction of the 3D scene from the stereo pair, this still leaves the problem of how to separate the 3D text from the 3D clutter which is difficult due to the different parallax. In addition, because of the characters are touching/overlapping this still leaves the problem of segmenting to individual characters. In short, even if one can successfully remove the clutter this will reduce to the difficulty of segmentation attacks which forms the basis of security for existing 2D CAPTCHA approaches. Furthermore, human visual perception of 3D scenes is still an open research problem that cannot easily be modeled.

6 Applications

The incorporation of STE3D-CAP in an application requires that the application be usable with a stereoscopic 3D display. Two current areas are increasingly moving toward the use of 3D displays; namely, 3D games and 3D movies. STE3D-CAP can be included seamlessly into applications like 3D Massively Multiplayer Online Games (MMOGs). These are online games that support multiple players who interact in the same shared virtual space. Many of these games are already being developed or modified to cater for stereoscopic 3D displays. An example is the popular World of Warcraft$^{\text{TM}}$[2]. The use of CAPTCHAs in these applications will help deter the use of bots to gain an unfairly advantage over other players and ruin the fun for other players [14].

The number of applications that use stereoscopic 3D displays will certainly increase with more and more companies currently developing and producing glass-free 3D display devices [20]. A number of web pages already contain anaglyph flash and java applets, while others provide 3D images and videos [1]. Web browser plugins are currently being developed to be able to display stereoscopic 3D images and videos on web pages [28], and this will certainly become more and more widespread. In that respect, stereoscopic 3D CAPTCHAs is anticipated to be the way of the future. While stereoscopic displays have yet to proliferate, the existing solution is to adopt the low-cost anaglyph approach as a drop-in replacement for current CAPTCHAs on web pages.

7 Conclusion and Further Work

In this paper we presented a new stereoscopic 3D CAPTCHA, called STE3D-CAP, which attempts to overcome limitations with existing 2D approaches. We demonstrated that STE3D-CAP is resistant against the existing 2D CAPTCHA attacks. Our approach has opened a new research direction to incorporate CAPTCHA challenges in 3D scenes.

Our approach also gives rise to the possibility of producing animated 3D CAPTCHAs where either the camera's viewpoint is translated in 3D or the scene is moved with respect to the camera. The differences between animated frames will give rise to 3D depth perception of the text in the scene via motion parallax, where objects at a distance appear to move slower compared to objects what are close to the viewer. Furthermore, as the camera moves from one position to another, the 3D text which might have been obscured in one frame will become visible in another frame. In order to break this CAPTCHA, one would have to somehow correlate the content between frames whilst attempting to separate the 3D text from the background and foreground clutter, which is not an easy task. This approach will work on standard displays and can be incorporated into web pages as animated Graphics Interchange Format (GIF) images.

References

1. Anaglyph flash gallery, http://www.3dmix.com/eng/flash-gallery/
2. Activision Blizzard. World of Warcraft, http://www.worldofwarcraft.com/
3. Bigham, J.P., Cavender, A.C.: Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use. In: Proceedings of the 27th International Conference on Human Factors in Computing Systems, pp. 1829–1838 (2009)
4. Bourke, P.: Calculating stereo pairs,
   http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/stereographics/stereorender/
5. Bourke, P., Morse, P.: Stereoscopy: Theory and Practice. In: Workshop at the 13th International Conference on Virtual Systems and Multimedia, VSMM 2007 (2007),
   http://local.wasp.uwa.edu.au/~pbourke/papers/vsmm2007/stereoscopy_workshop.pdf
6. Chellapilla, K., Larson, K., Simard, P., Czerwinski, M.: Building Segmentation Based Human-friendly Human Interaction Proofs. In: Baird, H.S., Lopresti, D.P. (eds.) HIP 2005. LNCS, vol. 3517, pp. 1–26. Springer, Heidelberg (2005)
7. Chellapilla, K., Larson, K., Simard, P., Czerwinski, M.: Computers beat humans at single character recognition in reading based human interaction proofs. In: 2nd Conference on Email and Anti-Spam (2005)

8. Chellapilla, K., Larson, K., Simard, P., Czerwinski, M.: Designing human friendly human interaction proofs (HIPs). In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 711–720 (2005)
9. 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)
10. Elson, J., Douceur, J.R., Howell, J., Saul, J.: Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization. In: Proceedings of the 14th ACM Conference on Computer and Communications Security (ACM CCS 2007), Conference on Computer and Communications Security, pp. 366–374 (2007)
11. Gateau, S.: Th. In: and Out: Making Games Play Right with Stereoscopic 3D Technologies. NVIDIA presentation, Game Developers Conference (2009)
12. Gizmodo: Sony plans to introduce 3D LCD television by the end (2010), `http://gizmodo.com/5350607/sony-plans-to-introduce-3d-lcd-television-by-end-of-2010`
13. Golle, P.: Machine Learning Attacks Against the Asirra CAPTCHA. In: Proceedings of the 14th ACM Conference on Computer and Communications Security (ACM CCS 2008), Conference on Computer and Communications Security, pp. 535–542 (2008)
14. Golle, P., Ducheneaut, N.: Preventing bots from playing online games. Computers in Entertainment (CIE) 3(3), 3 (2005)
15. Google Images, `http://images.google.com`
16. Jakobsson, M.: Captcha-free throttling. In: Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence, pp. 15–22 (2009)
17. McAllister, D.: 3D Displays. Wiley Encyclopedia on Imaging, pp. 1327–1344 (2002), `http://research.csc.ncsu.edu/stereographics/wiley.pdf`
18. Mitra, N.J., Chu, H.-K., Lee, T.-Y., Wolf, L., Yeshurun, H., Cohen-Or, D.: Emerging images. ACM Transactions on Graphics (TOG) 28(5) (December 2009)
19. Mori, G., Malik, J.: Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In: Conference on Computer Vision and Pattern Recognition (CVPR 2003), pp. 134–144 (2003)
20. Murph, D.: Intel shows off glasses-free 3D demo (2010), `http://www.engadget.com/2010/01/10/intel-shows-off-glasses-free-3d-demo-now-this-is-more-like-it/`
21. NVIDIA. NVIDIA 3D Vision, `http://www.nvidia.com/object/3D_Vision_Main.html`
22. OCR Research Team. Teabag 3D Revolution, `http://www.ocr-research.org.ua/teabag.html`
23. Ross, S., Chen, T.L.: The Effects of Promoting Patient Access To Medical Records. Journal of American Medical Informatics Association 10, 129–138 (2003)
24. Simard, P., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: International Conference on Document Analysis and Recognition, pp. 958–962 (2003)
25. von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: The CAPTCHA web page, `http://www.captcha.net`
26. von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: CAPTCHA: Using hard AI problems for security. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 294–311. Springer, Heidelberg (2003)
27. von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M.: reCAPTCHA: Human-Based Character Recognition via Web Security Measures. Science 321(5895), 1465–1468 (2008)

28. VREX. DepthCharge V3 Browser Plug-In, http://www.vrex.com/depthcharge/
29. Yan, J., Ahmad, A.S.E.: A Low-cost Attack on a Microsoft CAPTCHA. In: Proceedings of the 14th ACM Conference on Computer and Communications Security (ACM CCS 2008), Conference on Computer and Communications Security, pp. 543–554 (2008)
30. Yan, J., Ahmad, A.S.E.: Usability of CAPTCHAs - Or Usability issues in CAPTCHA design. In: Symposium on Usable Privacy and Security (SOUPS) 2008, pp. 44–52 (2008)
31. YUNiTi, http://www.yuniti.com