Multimed Tools Appl (2015) 74:5141–5156
DOI 10.1007/s11042-013-1666-7

Game-based image semantic CAPTCHA on handset devices

Tzu-I Yang · Chorng-Shiuh Koong · Chien-Chao Tseng

Published online: 28 September 2013

Abstract A completely automated public turing test to tell computer and human apart (CAPTCHA) is based on the Turing test, which aims to protect Internet services from automatic script attacks and spams. However, most proposed or deployed CAPTCHAs have been breached. It is possible to enhance the security of an existing CAPTCHA by adding noises systematically adding noises, but distortions would make characters recognition difficult for humans. On the other hand, most of the traditional CPATCHAs require complicated operations using keyboards and mice which may become limitations of modern handset devices. In this study, we propose a novel GISCHA using game-based image semantics with the contributions that 1) use simple keys, mouse, gesture, and accelerometer instead of complex alphabet inputs; 2) is language independent; 3) enhances the security level without annoying users; 4) is based on more advanced human cognitive abilities; and 5) make CAPTCHAs more interesting. The experiment results show that a single GISCHA challenge was completed in 9.06 s on average with a virtual keyboard and 10.25 s on average with accelerometers build in handset devices, and the pass rate of first time use is 94.8 %, which means that it is sufficiently easy for practical use.

Keywords CAPTCHA · Human interactive proofs · Game-based · Unambiguous image semantic · GISCHA

1 Introduction

With the rapid growth network technologies, Internet services including cloud computing, email services, online registration, online voting, chat rooms, weblogs, and online games are constantly being developed. Because most of the services are free of charge, they have become prone to attacks. Automatic scripts and bots have been established to create free accounts, send spam, cheat in online games, and vote remotely. Although some authentication studies [2, 14, 19] have

T.-I. Yang (✉) · C.-C. Tseng
Department of Computer Science, National Chiao Tung University, 1001 University Road, Hsinchu, Taiwan 300
e-mail: tiyang@cs.nctu.edu.tw

C.-C. Tseng
e-mail: cctseng@cs.nctu.edu.tw

C.-S. Koong
Department of Computer Science, National Taichung University of Education, 140 Minsheng Road, Taichung City, Taiwan 403
e-mail: csko@mail.ntcu.edu.tw

focused on helping to recognize users, privacy issue has become another burden. Human Interactive Proofs (HIPs) or completely automated public turing tests to tell computer and human apart (CAPTCHAs) [17, 21, 23] are widely used to resist these types of attacks. A CAPTCHA is a type of challenge-response test used in computing as an attempt to ensure that a response is generated by human beings. The process usually involves a computer asking a user to complete a simple test that the computer is able to grade. These tests are designed such that they are easy for a computer to generate but difficult to solve. However, owing to the limitations of the image sets and the types of response, they can be easily broken by recreating the entire database [8, 12, 17] or using image recognition [5, 18, 20]. Alternatively, text-based CAPTCHAs, require the user to translate an image or sound of words, but it still appears to be easily broken [5, 6, 11, 13, 24]. Although it is possible to systematically enhance the security of existing CAPTCHAs, the addition of noise distortion and obfuscation techniques would make characters recognition difficult for humans. Studies have shown that even computers can sometimes do much better than human beings [4, 7]. In addition, designs based on both text and speech recognition are language dependent.

On the other hand, traditional CPATCHAs, are designed for personal computers, which have a large screen and physical keyboard and mouse, as the input may not be suitable for handset devices. The examples show a traditional CAPTCHA on a handset device (i.e., iphone4s), which is inconvenient for users to operate because most of the submit buttons and challenge images are prohibitively small owing to the limitation of screen size and virtual keyboard (Fig. 1). In most cases, users need to zoom in/out and move up/down using different gestures to reveal and complete a challenge. Because the number of handset device users continues to undergo rapid growth, the CAPTCHA design for mobile users has become one of the most important features.

In this study, we focus on more advanced human cognitive process abilities and propose a new scheme based on game-based image semantics named GISCHA. The GISCHA may be as simple as a rolling ball or puzzle games, which can be easily operated using only simple arrow keys, mouse movements, gestures and accelerometer. We believe that GISCHA has an extremely high resistance to automated malware attacks because it is considered to be nearly impossible for

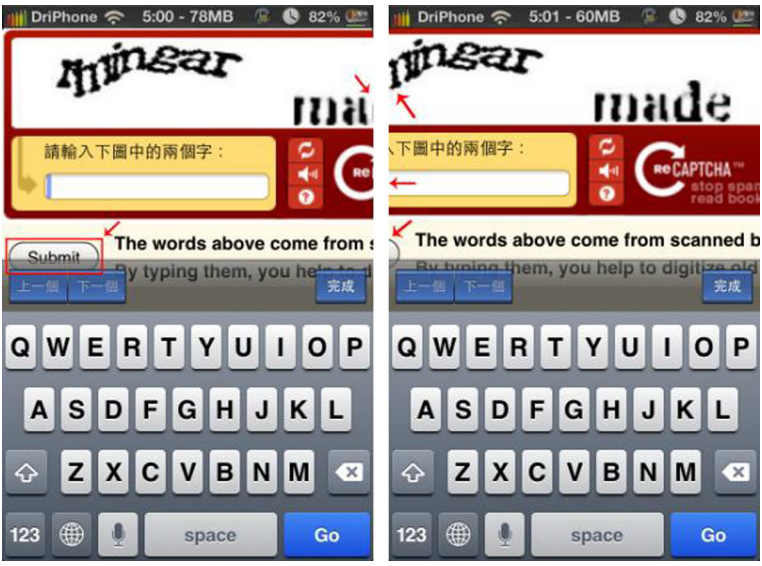

Fig. 1 Traditional CPATCHAs on iPhone4S

computers to reach such an advanced stage of artificial intelligence, regardless of how advanced the technology may be. Furthermore, it is language independent and background knowledge is not required.

We briefly introduce the related CAPTCHA works in Section 2 and the concept of GISCHA is described in Section 3. In Section 4, we confirm the usage of GISCHA, and discuss our design in section 5. We conclude our paper in Section 6.

2 Related works

A CAPTCHA is based on the Turing test concept, which was developed by Turing [21, 22]. The most common and successful CAPTCHAs are visually distorted images of letters and numbers that can ideally be identified by humans, but not by computers [8, 17]. CAPTCHAs can be grouped into three general categories: 1) Text-based: A string of characters is presented to the user, who is asked to recognize the combination of either words or random alphanumeric characters and punctuations [5, 12]. 2) Image-based: Images are presented to the user, who is normally challenged in the form of identifiable real-world objects, but which can also be presented in the form of shapes [24]. 3) Audio-based: The user is presented with an audio version of a CAPTCHA [1]. Current CAPTCHA implementations are primarily visual-based, and are therefore inaccessible to users who are unable to view the screen. On the other hand, audio CAPTCHAs ask users to interpret spoken audio sounds, thus posing other challenges (e.g., language dependency and dealing with the presence of annoying noises). To defeat automated speech recognition, these audio HIPs use a significant amount of background noise and a variety of speakers, making interpretation difficult. Many sites do not provide audio CAPTCHAs [1], and in this study, we therefore focus only on visual CAPTCHAs.

2.1 Text-based CAPTCHA

The ReCAPTCHA (Fig. 2) [25], which is character-based, recognition-based, and sound-based CAPTCHA system, is one of the most successful examples. However, with the rapid growth of computation power and development of advanced algorithms, most of the text-based CAPTCHAs have been reported to be breached. Studies have shown that more than 85 % of the commercial CAPTCHAs are vulnerable to automated attacks [5, 18]. Studies have also demonstrated that most CAPTCHA schemes are broken if they can reliably segmented [6, 11, 13]. Although the security levels can be enhanced by systematically adding distortion and noise to a text-based CAPTCHA, it becomes difficult for humans to recognize characters (Fig. 3).

Fig. 2 An example of ReCAPTCHA

Fig. 3 An example of CAPTCHAs with enhanced distortions and noises

2.2 Image-based CAPTCHA

Image-recognition-based CAPTCHAs have been considered as one of the best alternatives to text-recognition-based CAPTCHAs (Fig. 4). Images, which are intuitive to humans and are of a large variety, are rich in information. Based on the type of challenge presented, image-based CAPTCHAs can be categorized into two groups: anomaly-based and recognition-based. The recognition-based CAPTCHAs ask the user to identify the object in images [15] and the anomaly-based CAPTCHAs ask the user to determine the object that does not belong to the images [22]. Early image-based CAPTCHAs include Bongo and Pix [22], which use shapes and labeled images as challenges, and ask users to choose the tag related to all the images. Because of the limited number of shapes and labels, they may suffer from guessing and database reconstruction attacks [8]. Recently, image recognition techniques have also been able to help break the CAPTCHA system [5, 18, 20, 26]. In addition, labeling and objects may be ambiguous because different persons' perceptions differ (i.e., HotCAPTCHA asks users to select a person found to be most attractive).

On the other hand, Asirra [10], relies more on the human nature, because of the different capabilities of humans and bots to distinguish between cats and dogs. However, some images may not suitable for challenges, allowing the machine learning techniques to attack these types of CAPTCHAs [12]. Because of the limited number of image sets, potentially leading to database

Fig. 4 An example of an image-based CAPTCHA

Fig. 5 A four-panel cartoon CAPTCHA

reconstruction attacks [8], recent CAPTCHAs have been designed to use the image search function provided by Google. Unfortunately, the search results may not be completely reliable because the returned images may additional confusing objects (e.g., images containing both dogs and cats). In addition, hackers can also obtain the correct answers by sending the challenged images to Google's search engine.

Another example of CAPTCHAs is based on the image semantics. A randomly ordered four-panel cartoon is used, and the user is asked to reorder them as a human would (Fig. 5) [26]. It is almost impossible for the computer to understand the humor intended by the human beings. However, it is not yet understood how to properly make use of image semantics, called ambiguous semantics. Besides, Fig. 5 shows that it may fail to pass the challenge if the user is not familiar with Japanese language and culture. In addition, a four-panel cartoon CAPTCHA may suffer from brute force attacks because there are only 4! Combinations, and databases with four-panel comics may be limited in size.

So far, existing CAPTCHAs are also suffering from defects involving language dependency and annoyance. Most of the existing CAPTCHA systems require a short paragraph to explain how it works, and provide tough challenges that lack native language support. Hence, we have to find another more advanced human cognitive processing ability to tackle this challenge.

3 GISCHA

3.1 Concept

In this study, we focused on the ability to solve games, which is considered to be one of the most advanced human cognitive processing abilities. Through simple web-based games, we can easily identify human because it is almost impossible for computers (Artificial Intelligence, AI) to

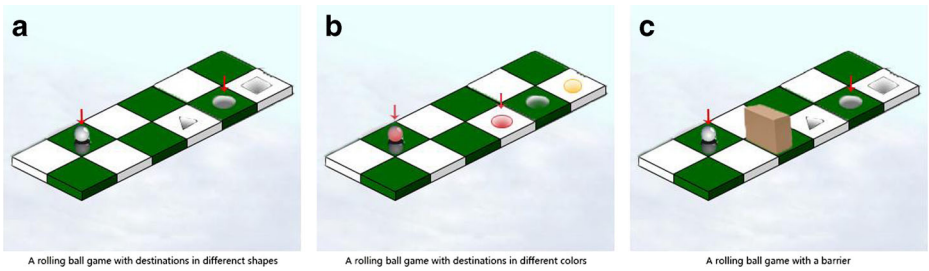

Fig. 6 Rolling ball game example

understand the meanings of all types of games. In addition, simple games are considered to be solvable by all age groups without additional background, and they are also language independent.

As a specific example, we have proposed a GISCHA using the simple rolling ball game (Fig. 6a). In this example, there is a rolling ball and destination holes with different shapes; the user who is able to move the ball to the destination hole shaped as a circle is identified as a human. However, for a computer, it would be difficult to understand the meaning of the rolling ball game and make corresponding movement. Moreover, even if image processing capabilities are developed to the level where the computer could recognize the meanings of the images and make correct responses, it would be almost impossible for computers to understand the real meaning of the rolling ball game by adding traps (Fig. 6b) and enticing the user to fake destinations (Fig. 6b), in which destinations have the same shape (circular), but different colors. Moreover, randomly changing the types of paths with barriers (Fig. 6c) gives another example. For human beings, it is easy to associate the ball with the same color if they recognize that all destinations have the same shape. Fig. 6c requires basic spatial ability, which is natural to human beings.

3.2 Control system

Performing traditional CAPTCHAs on a handset device may require more time because most CAPTCHAs were designed and displayed on a large screen. Another problem is that they are usually designed to be operated using physical keyboard and mouse, which is also a limitation for users of handset devices [9]. An accelerometer is a device that measures motion inputs and changes its orientation. As of early 2009, almost all new mobile phones and digital cameras were designed with at least a tilt sensor, and sometimes also had an accelerometer for the purpose of auto image rotation, motion-sensitive mini-games, and to compensate for shaking when taking photographs. In this study, we proposed two games, which can be operated by the accelerometer. Fig. 6 gives an example of GISCHA using HTML5 and JavaScript, which can be operated by a traditional keyword and mouse. In addition, we enhanced the control ability of handset devices by using a virtual keyboard and accelerometer. We use the information provided by accelerometers and gyroscopes which is built into modern handset devices. By turning handsets at different angles, one can control the direction of the rolling ball. Fig. 7a shows how to move the ball to the left side, Fig. 7b shows how to move the ball to the right, Fig. 7c shows how to move the ball to the top, and so on.

3.3 Definition

von Ahn et al. [23] broadly defined the CAPTCHA mathematically for transferring of AI problems into CAPTCHAs. In this study, on the basis of this mathematical model, we deduce a corollary to clarify that GISCHA can be automatically generated. The definitions are as follows.

Definition 1 A test V is said to be (α, β)-*human executable* if at least an α portion of the human population has success greater than β over V.

Definition 2 A game/problem is a triple $P=(S, D, f)$, where S is a set of problem instances, D is a probability distribution over S, and $f: S \rightarrow \{0,1\}^*$ answers the instances. Let $\delta \in (0,1]$. We require that for an $\alpha>0$ fraction of human H, $\Pr_{x \leftarrow D}[H(x)=f(x)] \geq \delta$ where Pr(.) is the deterministic programming that results when P uses random coins r.

Definition 3 A game/problem is said to be (δ, τ)–*solved* if there exists a program A, running in time at most τ on any input from S, such that $\Pr_{x \leftarrow D, r}[A_r(x) = f(x)] \geq \delta$.

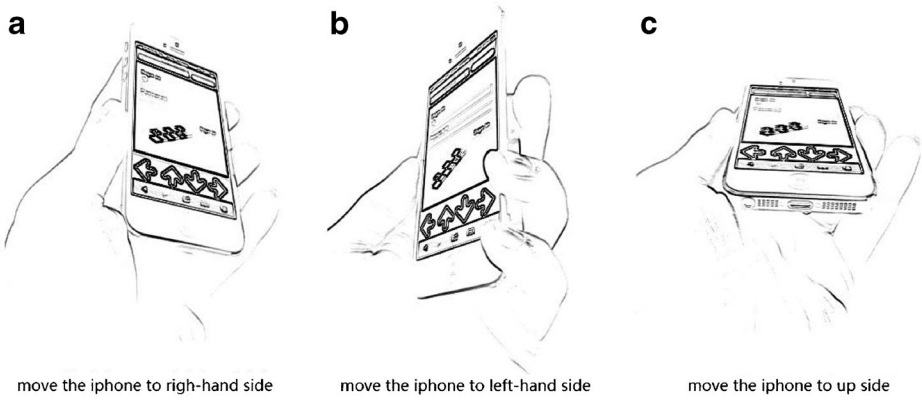

Fig. 7 Control system of GISCHA

3.4 Game problem

Let I be a distribution on images and T be a distribution of image transformations. We assume for simplicity that if $i,i' \in [I]$ and $i \neq i'$, then $T(i) \neq T'(i')$ for any $T,T' \in [T]$ where $[I]$ denotes the support of I. Let M be a finite set of movements. Let $\lambda:[I] \rightarrow M$ compute the movement of a game. The set of problem instances is $S_{I,T}=\{t(i):t \in [T] \text{ and } i \in [I]\}$, and the distribution on instances $D_{I,T}$ is the one induced by choosing $i \leftarrow I$ and $t \leftarrow T$. Define $g_{I,T,\lambda}$ so that $g_{I,T,\lambda}(t(i))=\lambda(i)$. Then the game problem $GP_{I,T,\lambda}=(S_{I,T}, D_{I,T}, g_{I,T,\lambda})$ involves writing a program that take $t(i)$ as the input and outputs $\lambda(i)$.

3.5 Definition of GISCHA

A GISCHA instance is described by a tuple $G=(I,T,M,\lambda,\tau)$, where I is a distribution of images and T is a distribution of image transformations that can be easily computed using current computer programs. GISCHA is a CAPTCHA with the following property: *any program that has high success over* $G=(I,T)$ *can be used to solve game problem (GP)*. The GISCHA verifier works as follows. First, V draws $i \leftarrow I$, and $t \leftarrow T$. Then, V sends to user U the message $\{t(i),M\}$, and sets a timer for τ. P responds with an operation $m \in M$. V accepts if $m=\lambda(i)$ and its timer has not expired, and rejects otherwise.

Corollary 1 If $G=(I,T,M,\lambda,\tau)$ is *(α,β)-human executable*, then G is a *(α,β)-CAPTCHA.*

3.6 Algorithms

The notations used in this paper are listed as follows:

Notation	Description
Info	The user's information.
OTPU	One time pass solution sends by the user.
r_j	The random factor.
EC(.)	The generator of GISCHA with input Random r.
A➔B: M	Entity A sends message M to entity B.

The proposed scheme is composed of two phases: registration and verification. The registration phase presents *EC* to the newly incoming user with random factor r_j, which contains the colors, arrows and type of the destinations. After the user fills *info* and completes $EC(r_j)$ with $OTPU_i$, the verification phase will check whether $EC(OTPU_i)$ has been accomplished. If the $EC(OTPU_i)$ output has the value "PASS," then the server will send the message to the user regarding the success of the login or registration. Otherwise, the server will generate another $EC(r_j+1)$ with newly generated r_{j+1}.

A. Registration Phase

If a user visits a web service, he/she may register a new account or login to the server.

A1: Server→User: $EC(r_j)$

a. Generate r_j
b. Generate $EC(r_j)$
c. Send $EC(r_j)$ to User *i*

A2: User→Server: (*Info*, $OTPU_i$)

a. The user i provides his/her information Info for logon or registration on the service.
b. Response $OTPU_i$ according to *EC*
c. Submit (*Info*, $OTPU_i$)

B. Verification Phase

Upon receiving the registration or login request from the user, the server verifies the GISCHA response and provides additional services:

B1: Server→User: $EC(r_{j+1})$ or PASS

a. verify $EC(OTPU_i)$
b. if $EC(OTPU_i)$ output fails then the server regenerates r_{j+1}
c. render $EC(r_{j+1})$ and sends to user for another retry
d. if $EC(OTPU_i)$ output passes then send user login success and redirect web pages

4 Experiment design

In this section, we conducted several experiments to evaluate the first-time pass rate, the average operation time with a different control system, and the preference of the user. We also asked the user to perform the traditional text-based CAPTCHA system for the comparison purposes.

4.1 Subjects

There are 42 participants in this experiment with ages from 15 to 60, including professors, students, engineers and housekeepers. They performed these selected GISCHAs from their own handset devices or using provided iPad minis, which included iphone4S, iphone5, Desire HD, and Nexus7. We also asked participants to perform reCAPTCHA [25] operations on handset devices as a control group.

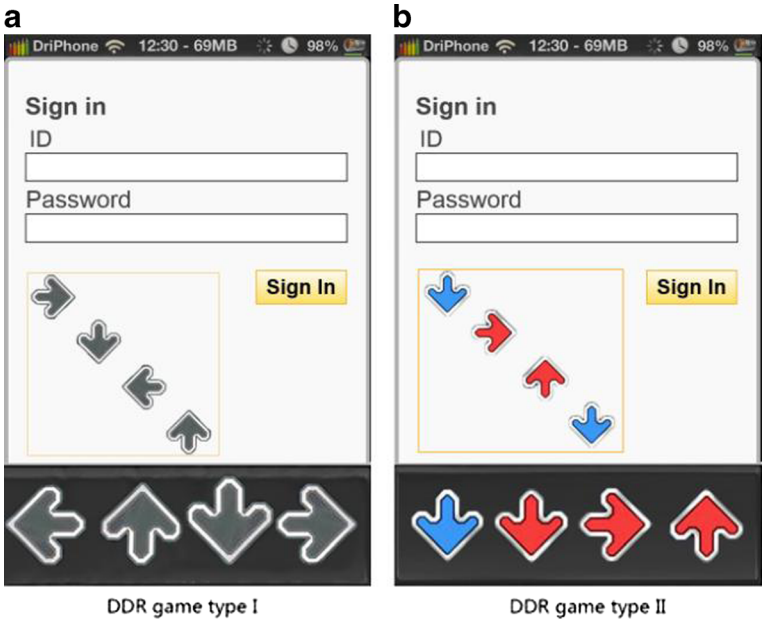

Fig. 8 A DDR GISCHA game

4.2 Experimental tools and procedures

In Section 3, we clearly defined the requirements, then we proposed two different types of GISCHAs as the experiment tool on the basis of the Corollary. There were three different types of rolling ball games: 1) destinations with different shapes, 2) destinations with different colors, and 3) paths with barriers (Fig. 6). The other game is the Dance Dance Revolution (DDR) game, which is based on the concept of the video game and has two types: 1) arrows with a sequence from top to bottom and left to right, and 2) arrows of type (1) with different colors added (Fig. 8). In Table 1, we summarize the types of experiments and control systems. Because there are different test platforms (e.g., iOS and Android) and different sizes of handset devices (e.g., screen sizes from 3.5 to 7 in.), we have to develop the experimental tools using HTML5, CSS3, and JavaScript technologies to enable cross-platform capabilities with a consistent operating environment. The other challenge is that there may be different sample rates on the platforms of iOS and Android, and so there may

Table 1 Experimental setup with different types

Controls / Games	Rolling I	Rolling II	Rolling III	DDR I	DDR II	reCAPTCHA
Virtual Keyboard	O	O	O	O	O	O
Gestures	O	O	O	O	O	X
Accelerometers	O	O	O	O	O	X

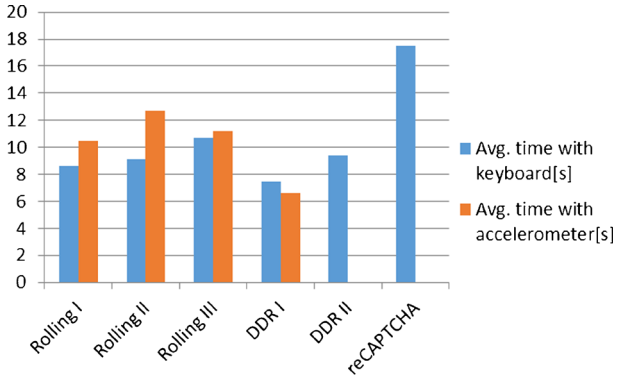

Fig. 9 The average challenge time with virtual keyboard as input

be inconsistent rolling speeds while operate the rolling ball games. In addition, the event firing models are also somewhat different for the different Internet browsers installed in these handset devices (e.g., Chrome, Firefox, and Safari).

There are six experiments in total: three for testing different types of rolling ball games, two for DDR games and one for reCAPTCHA. We did not explain how their operation, and so the participants played these GISCHAs for the first time. Then, before they passed the challenge, we revealed the first-time pass rate and the learning period. The participants were asked to strictly perform these six CAPTCHAs, and the operation time started when the user touched the canvas and stopped when they pressed the submit button.

4.3 Experimental results

In the experiment, all the GISCHAs had the pass rate of more than 90 %, which indicated that it was easy for human beings to operate. The repeat time indicates that most of the users passed the GISCHAs on the second trial, which implies that it is easy to learn through trial-and- error (Fig. 9). In comparison, when using reCAPTCHA, they need an average of 2.5 times to pass the challenge, which may imply that it is slightly more difficult for the users to recognize the distorted characters. From (Fig. 9), we observe that the average operating time of GISCHAs averages comes to 9.06 s and 17.5 s for reCAPTCHA, which means that the user needs more time to pass the challenge through virtual keyboards, and this may be possibly owing to the limitation of the screen size. The average operating time is not all positively correlated with the degree of operational difficulty because the average operating time of the different methods is not always the same, e.g., the keyboard operation time of DDR game type one is a little bit longer than when using the accelerometer. We also carried out a survey to find out whether users behind the screen are willing to use the GISCHA. The respective experimental results are shown in Table 2.

5 Discussions

5.1 Semantic ambiguities

One problem associated with the use of semantics is that the retrieved semantic information from an image tends to be subjective and user-dependent. For example, with the image-

Table 2 Experimental results

Games	Rolling I	Rolling II	Rolling III	DDR I	DDR II	reCAPTCHA
Pass rate of first time	0.97	0.95	0.97	0.95	0.90	0.69
Avg. time with KB[sec.]	8.60	9.10	10.70	7.50	9.40	17.50
Avg. time with Acc[sec.]	10.50	12.70	11.20	6.60	–	–
Avg. time of repeat[times.]	0	1.00	1.00	0	1.00	2.50
Preference [persons]		17		20		5

recognize-based CAPTCHA that asks the user to choose the most beautiful flowers, the result may be different for each person. This intrinsic ambiguity in semantics makes it difficult to generate CAPTCHA challenges using image semantics. Besides, the differences in users' intellectual growth and their knowledge may also lead to different results. Hence, we propose GISCHA which is based on unambiguous high-level semantics. We focused on the ability to solve games, which is considered to be one of the most advanced human cognitive processing abilities. In addition, simple games are considered solvable for all ages without the need for additional knowledge, and they are language independent. Fig. 10 shows that the user takes longer to finish the challenge when more semantics information is added to the picture, which is a sort of semantic ambiguity. However, the pass rates in Table 2 show that it only takes slightly more time to figure out the ambiguity.

5.2 Resisting the replay attack

Most of the CAPTCHAs are proposed and deployed with strong assumptions, such as that the communication channel must be secure and protected from any modification. The main reason is that they may suffer from the replay attack.

5.2.1 Scenario 1: Replay attack using eavesdropping

If the CAPTCHA scheme sends a packet that has a HTTP protocol, the challenges of the CAPTCHA and the corresponding password may be transferred in plain text. Because it is possible to eavesdrop on the transmission, the hacker can simply replace both the challenge and answer. For GISCHA, games can be generated on the user side by applying both FLASH and HTML5 technology which means that there is no specific pair of challenge and answer.

5.2.2 Scenario 2: Replay attack using database reconstruction

If the user uses automated programs that can record the corresponding answers, they can breach the CAPTCHA that appeared before by replaying the procedure, which is also named the database reconstruction attack [8]. Applications such as JDownloader and FreeRapid which are used to help download files from cloud stages protected by the CAPTCHA system, are used to construct the database of text-based CAPTCHA challenges. When the user is challenged by a text-based CAPTCHA that was not previously presented, the application will ask the user to input the answer. If the user passes the challenge, the application will record the challenge and answer pair for the next challenge, and so on. In our proposed scheme, it is difficult to realize the actual moves of solution, which are the solutions for different games; this means that it is difficult to determine if the challenge is the same as before and when to begin recording the corresponding answer. Besides, there exists more than one solution for the same game.

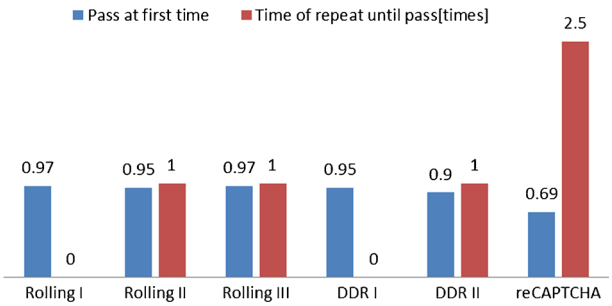

Fig. 10 The camparsion of pass rates and repeat times

5.3 Resisting the brute force attack

Because the *info*, e.g., password, of the traditional user authentication mechanism is usually fixed and rarely changes; it may suffer from brute force attacks. Some CAPTCHAs [8, 26] may suffer from the same attacks because the challenge and answer spaces are limited. Four-panel comics, for example, may suffer from this type of attack because there exist only 4! combinations. In the proposed scheme, the GISCHA, e.g., $EC(r_j)$ is an one-time permutation or one-time pad (OTP) and a human must be present to answer the $EC(r_j)$. If the user fails to pass the challenge, it will be changed in the next session with newly generated r_{j+1}, e.g., $EC(r_{j+1})$, which implies that the automated program cannot launch a so-called brute force attack because the correct answer will change every time. We analyzed the two proposed games as follows:

5.3.1 Scenario 1: Rolling ball games

For rolling ball games, the user can operate with only four arrow keys. However, the solution may be different because there exists more than one path to the destination. In our proposed scheme, using Fig. 6a as an example, the shortest path to the destination is going up and to the right using four arrow keys. However, the brute force attack will have to be tried at least 4^4 different keys before the best case solution is found. The situation may worsen if we add destinations have various colors and barriers.

5.3.2 Scenario 2: DDR games

For DDR games, the user can operate using only four arrow keys. In our proposed scheme, using Fig. 8a as an example, the correct answer is found by first pressing the right arrow, followed by the down arrow, then the left arrow time, and finally the up arrow, implying that they must be answered in sequence. A brute force attack will have to try at least 40 combinations before the correct sequence is found. In addition, the situation may become worse if we add the limitation of the wrong inputs. Besides, we may also add various colors as barriers for more advanced protection.

5.4 Resisting the machine learning

The use of machine learning to classify or recognize objects was an effective attack on Asirra [12], but it will not work on GISCHA because the objects used in a current challenge are uncorrelated with those used in other challenges. The reason why we added berries as the sample is that even though the computer can recognize the sample as a maze, it is almost

Table 3 Comparison of previously proposed CAPTCHA with GISCHA

Prevent from	Image-based		Audio-based	Text-based	GISCHA
	Recognition	Anomaly			
Mislableing	X	X	–	–	O
Misspelling	X	X	X	X	O
Synonymy	X	O	O	O	O
Polysemy	X	O	O	O	O
Brute Force/Guess attack	X	X	X	X	O
Replay attack	X	X	X	X	O
Language Dependent	X	X	X	X	O

impossible for a computer to know how to avoid the barriers without physical/spatial cognition; the barriers can be replaced by meanings that are only understood by human beings. This type of processing ability is innate and is common in our daily life. In other words, we can enhance the security by adding new and different types of obstacles by changing the position after every try.

5.5 Operation complexity and language independence

Traditional text-based CAPTCHAs usually ask the user to enter vocabularies and phrases, that may be difficult for foreigners to spell, and which therefore lead to high failure rates. However, image-based CAPTCHAs, may suffer from mislabeling, synonymy and polysemy troubles (Table 3). Because GISCHA was designed to use simple games in order to identify humans, they are designed to be operated by simple and repeatable keys. In other words, GISCHA is language independent both for the challenge and operation instructions, which are suitable for all aspects.

5.6 Entertainment and commercial aspects

Traditional CAPTCHAs are annoying to users, which may prevent the marketing expansion of commercial websites. Text-based CAPTCHAs with over distorted images may also increase the number of false results. In the proposed scheme, despite the fact that the proposed method may sometimes require more time for authentication compared to conventional CAPTCHAs using text-based ones, the level of usability experienced by the user is not expected to decrease significantly. In addition, users who use handset devices feel even better while operating GISCHA with an accelerator when walking or taking public transportation. Furthermore, there is a potential to realize commercial value. We demonstrated the rolling ball game on a flat surface, while indicating that these tiles can have other commercial purposes. For example, advertising posters can be used to decorate these tiles. In addition, all types of barriers can be big head dolls for commercial sausage.

6 Conclusion

Traditional CAPTCHAs may increase the sense of annoyance felt by users who have to prove that they are human every time they access the web. However, a CAPTCHA should bring a more

pleasant experience to the user. In this study, we proposed a GISCHA scheme that uses games to differentiate between humans and bots. We formally define the limitations and types of games that can be used to develop the GISCHA. We also introduced the concept of using accelerometers which are built into most of the handset devices to control the CAPTCHA for conveniences and to add more fun. We also developed three games to evaluate the usability of our proposed scheme. The results show that the GISCHA is easily operated and language independent. The experiments show that the average response time is 11.6 s and the correct attempt rate is above 90 %. Besides, most of the user can pass the challenge at the first try and the learning period is relatively short.

Currently, there remains scope for improvement in terms of both security and usability, and so we plan to make improvements to the proposed method using experiments to reveal the cognitive load of different games and GUIs. Furthermore, we need to evaluate the extent to which the correct response rate and the total response time for the GISCHA depends on the intelligence of each user, and we attempt to implement the GISCHA in other games that lead to better results.

Acknowledgments The authors would like to thank the National Taichung University of Education and Ministry of Education, Taiwan, for financially supporting this research under grants of the talent nurturing pioneer program for proactive SoC design - embedded systems and software engineering.

References

1. Bigham JP, Cavender AC (2009) Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use. In: CHI '09: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp 1829–1838
2. Birgale L, Kokare M (2012) Iris recognition using ridgelets. J Inf Process Syst 8:445–458. doi:10.3745/JIPS.2012.8.3.445
3. Bin B Zhu, Yan J, Li Q, et al. (2010) Attacks and design of image recognition CAPTCHAs. In: CCS '10: Proceedings of the 17th ACM conference on Computer and communications security. pp 187–200
4. Bursztein E, Bethard S, Fabry C, et al. (2010) How good are humans at solving CAPTCHAs? a large scale evaluation. In: Security and Privacy (SP), 2010 I.E. Symposium on. IEEE, pp 399–413
5. Bursztein E, Martin M, Mitchell J (2011) Text-based CAPTCHA strengths and weaknesses. In: CCS '11: Proceedings of the 18th ACM conference on Computer and communications security. ACM, pp 125–138
6. Chandavale AA, Sapkal A (2012) A New Approach towards Segmentation for Breaking CAPTCHA. In: International Conference on Security in Computer Networks and Distributed Systems. Springer, pp 323–335
7. Chellapilla K, Larson K, Simard P, Czerwinski M (2005) Computers beat humans at single character recognition in reading based human interaction proofs (HIPs). CEAS 2005—Second Conference on Email and Anti-Spam
8. Chew M, Tygar JD (2004) Image recognition captchas. In: Information Security, 7th International Conference. Springer, pp 268–279
9. Chow R, Golle P, Jakobsson M, et al. (2008) Making CAPTCHAs clickable. In: HotMobile '08 Proceedings of the 9th workshop on Mobile computing systems and applications. pp 91–94
10. Elson J, Douceur JR, Howell J, Saul J (2007) Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In: ACM Conference on Computer and Communications Security. pp 366–374
11. Fang K, Bu Z, Xia ZY (2012) Segmentation of CAPTCHAs based on complex networks. In: 2012 AICI Annual Conference. Springer, pp 735–743
12. Golle P (2008) Machine learning attacks against the Asirra CAPTCHA. In: CCS '08: Proceedings of the 15th ACM conference on Computer and communications security. ACM, pp 535–542
13. Huang S-Y, Lee Y-K, Bell G, Ou Z-H (2010) An efficient segmentation algorithm for CAPTCHAs with line cluttering and character warping. Multimedia Tool Appl 48:267–289. doi:10.1007/s11042-009-0341-5
14. Kim J-J, Hong S-P (2011) A method of risk assessment for multi-factor authentication. J Inf Process Syst 7:187–198. doi:10.3745/JIPS.2011.7.1.187
15. Kim J, Kim S, Yang J et al (2013) FaceCAPTCHA: a CAPTCHA that identifies the gender of face images unrecognized by existing gender classifiers. Multimedia Tool Appl. doi:10.1007/s11042-013-1422-z
16. Li S, Shah SAH, Khan MAU, et al. (2010) Breaking e-banking CAPTCHAs. In: ACSAC '10: Proceedings of the 26th Annual Computer Security Applications Conference. pp 171–180
17. Manuel B, Louis von A, John L, Nick H (2000) The CAPTCHA Project. In: The CAPTCHA. http://www.captcha.net/. Accessed 11 Feb 2013

18. Mori G, Malik J (2003) Recognizing objects in adversarial clutter: Breaking a visual CAPTCHA. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 I.E. Computer Society Conference on. IEEE, pp 134–141
19. Satone MP, Kharate GK (2012) Face Recognition Based on PCA on Wavelet Subband of Average-Half-Face. J Inf Process Syst 8:483–494. doi:10.3745/JIPS.2012.8.3.483
20. Simard PY (2004) Using machine learning to break visual human interaction proofs. Adv Neural Inf Process Syst 17:265–272
21. Turing A (1950) Computing Machinery and Intelligence. Mind 49:433–460
22. von Ahn L (2009) Human computation. In: Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE. pp 418–419
23. von Ahn L, Blum M, Hopper N, Langford J (2003) CAPTCHA: using hard AI problems for security. Adv Cryptol
24. von Ahn L, Blum M, Langford J (2004) Telling humans and computers apart automatically. Commun ACM 47:56–60
25. von Ahn L, Maurer B, McMillen C et al (2008) recaptcha: Human-based character recognition via web security measures. Science 321:1465–1468
26. Yamamoto T, Suzuki T, Nishigaki M (2010) A Proposal of Four-Panel Cartoon CAPTCHA: The Concept. In: Network-Based Information Systems (NBiS), 2010 13th International Conference on. pp 575–578

Tzu-I Yang received the B.S. and M.S. degree in Computer Science and Information Engineering from Tunghai University, Taichung Taiwan, in 2002 and 2004. He is currently pursuing the Ph.D. degree at the Department of Computer Science and Engineering, National Chiao Tung University, Hsinchu, Taiwan. His research interests include image processing, graphic user interface, software engineering and network security.

Chorng-Shiuh Koong received his MS and PhD degrees in computer science and information engineering from National Chaio-Tung University (Hsinchu, Taiwan) in 1995 and 2000, respectively. Currently, he is an associate professor at the department of Computer Science, National Taichung University of Education

(Taichung, Taiwan). His research interests include e-learning, multimedia authoring technology, interactive learning environments, visual language and software engineering.

Chien-Chao Tseng is currently a professor in the Department of Computer Science at National Chiao Tung University, Hsinchu, Taiwan. He received his B.S. degree in Industrial Engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1981; M.S. and Ph.D. degrees in Computer Science from the Southern Methodist University, Dallas, Texas, USA, in 1986 and 1989, respectively. His research interests include wireless internet, handover techniques for heterogeneous networks, and mobile computing.