Electronic Letters on Computer Vision and Image Analysis 16(3):1-13, 2017

Random Image Matching CAPTCHA System

Hassan Hajjdiab*, Mohammed Ghazal+ and Ashraf Khalil*;

* *Department of Computer Science and Information Technology, Abu dhabi University , Abu Dhabi, City, United Arab Emirates*
+ *Department of Electrical and Computer Engineering, Abu Dhabi, City, United Arab Emirates*

Received 19th Dec 2016; accepted 23rd Jul 2017

Abstract

Online security risks is an important issue and caught the attention of researchers in the area of networks, web development, human computer interaction and software engineering. One main challenge for online systems is to identify whether the users are humans or software robots (bots). While it is natural to provide service to human users, providing service for software robots (bots) comes with many security risks and challenges. Software robots are often used by spammers to create fake online accounts, affect search engine ranking, take part in on-line polls, send out spam or simply waste the resources of the server. In this paper we introduce a visual CAPTCHA technique that is based on generating random images by the computer, the user is then asked to match a feature point between two images (i.e. solve the correspondence problem as defined by the researchers in the computer vision area). The relationship between the two images is based on a randomly generated homography transformation function. The main advantage of our approach compared to other visual CAPTCHA techniques is that we eliminate the need for a database of images while retaining ease of use.

Key Words: CAPTCHA, Internet Security, Computer Vision, Image Matching.

1 Introduction

The question of distinguishing between communications initiated by humans as opposed to those that are initiated by software robots have been discussed from the early age of Artificial Intelligence (AI). Alan Turing who is considered by many as the father of AI was the first to devise a seminal test that was later called the "Turing Test" [1]. The Turing test is usually administered by a human targeting a computer. However, CAPTCHAs [2] are administered by computers targeting humans, which has resulted in CAPTCHA to be described as a Reverse Turing Test. The main purpose of CAPTCHA systems, often called Human Interactive Proofs (HIPS), is to distinguish humans from software robots by providing challenges that are easily solved by humans but are too hard for computers. An effective CAPTCHA does not suggest that no software can be built to solve it with a reasonable success rate but rather that the cost of building such a tool would be too expensive in terms of development and computational requirements to be practical. The goal is to make the cost of building and using software to break CAPTCHA higher than the cost of using a human. All CAPTCHA systems must satisfy three basic properties:

Correspondence to: $hassan.hajjdiab@adu.ac.ae$

Recommended for acceptance by Àngel Sanchez
http://dx.doi.org/10.5565/rev/elcvia.1036
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

1. Must be easy for humans to solve
2. Must be hard for computers and software robots to solve
3. Must be supported by a large and dynamic set of test cases that is not possible for a computer to know in advance

The set should be easy to generate and grade. The goal of the large and dynamic set is to prevent the risk of an attacker from generating all possible answers to all of the possible tests.

1.1 Text-based systems

The most common form of CAPTCHA is the text-based CAPTCHA (see Fig. 1). In the text-based CAPTCHA, the user is asked to transcribe an image of deformed ASCII characters. The letters are deformed using various techniques such as FreeType fonts, background grids and gradients, blurring, re-orientation, and additive pixel noise. Fig. 1 shows examples of different text-based CAPTCHAs used by Microsoft's $Hotmail$. Since CAPTCHA was first introduced by Von Ahn [3] in 2000, hundreds of different text-based variations have been introduced. Yahoo! was one of the first major sites to introduce the EZ-Gimpy CAPTCHA version which was prepared by Carnegie Mellon University. EZ-Gimpy was made of full words with simple character distortion. EZ-Gimpy proved to be easily solved automatically 92% of the time [4]. To overcome the weakness of selecting English words from the dictionary, Paypal and many other major websites have relied on CAPTCHAs that use a random string of Latin letters and digits. However, this was also proved to be vulnerable to automated attacks [5, 6]. Subsequently, more advanced text-based CAPTCHAs were introduced. Researchers at Microsoft showed that segmentation (distinguishing the area of the characters, i.e. the start and the end positions of each character) is a particularly hard problem for computers and thus used this feature in their CAPTCHA system [7]. Microsoft's Hotmail used this service, however due to usability concerns it was removed; capitalizing on the harder segmentation made it very difficult for humans to solve the challenge as well. Nowadays, most online services try to overcome the vulnerability of CAPTCHAs by adding more noise and more distortion to the point that they are also very hard for humans to solve, thus causing major usability concerns for the users. Making text-based CAPTCHAs hard for computers usually makes them difficult for humans as well. To overcome this disadvantage many companies like [8, 9, 10] offer customers a subscription-based service to solve this kind of CAPTCHA's. This undesirable feature has led many researchers to propose the use of image-based CAPTCHAs instead.

1.2 Image-based systems

Image-based CAPTCHAs usually capitalize on the computers' inherent weakness in the field of computer vision rather than in the field of Optical Character recognition, as is the case for text-based CAPTCHAs. Chew and Tygar were among the first to propose using images in distinguishing humans from computers [11]. They specifically proposed using Google image search [14] to generate images that the user is asked to label. If the user generates the same label as the one returned by Google, then the user passes the test. Their technique of populating the database (the test set) with images required human intervention to make sure the association between the image and the label was clear and intuitive. The human intervention is the main disadvantage of this approach as it leads to limiting the size of the database that can be reconstructed by the attackers. Another solution was proposed by von Ahn et al. in which they constructed a collaborative game that resulted in labeled images as an outcome of playing the game [12]. These labeled images are added to the image database and are subsequently used by their PIX CAPTCHA (see Fig.3) which asks the tester to write the appropriate label for the shown image in order to pass the test. Solving this CAPTCHA system using machine learning algorithms is hard; however this solution has experienced limited success due to its difficulty to humans and to the limited size of its test database.

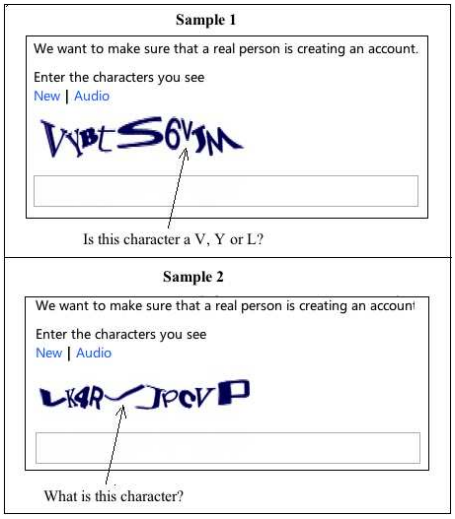


Figure 1: Hotmail CAPTCHA: hotmail uses text-based CAPTCHA. Human users commonly confuse letters such **V** with **L**, **cl** with **d**, **av** with **cw** and many others.

A more recent approach developed by Microsoft Research tries to address the problem of the limited size of the image database by using a public database of images of homeless cats and dogs from PetFinder.com [13]. In this Asirra (Animal Species Image Recognition for Restricting Access) CAPTCHA, users are asked to distinguish cat images from dog images given 12 images taken from the PetFinder database, which contains more than three million images of homeless animals [14]. The images in the database have been previously categorized by humans for PetFinder. The strength of Asirra stems from its significant advantage in usability compared to text-based CAPTCHAs and on its presumption of a computer's difficulty in classifying images of cats and dogs automatically. A brute force attack could solve the 12 image Asirra CAPTCHA problem with probability of $(\frac{1}{2})^{12}$ (i.e. 0.024%). However, according to [15], ASIRRA can be solved by machine vision attack with 10.3% success rate . This success rate is significant and could cause important security risks by a software robot. To block such attacks Asirra [14] adopted a token bucket scheme that penalizes an IP address for successive wrong answer.

Vikram et. al [16] proposed an image-based CAPTCHA called SEMAGE (**S**ematically **MA**tching ima**GE**). The idea of this apporach is to establish a semantic relation between a subset of images. SEMAGE approach starts by presenting a set of images with a subset of them sharing a certain relation with each other. The user is challenged by correctly identifying the subset of semantically related images.

Figure 4 shows a sample challenge for a user. The user is required to identify the two semantically related images. In this case the two encircled images are related. SEMAGE is very hard to be solve by a computer using machine learning techniques, however a brute force could solve the 6 image SEMAGE CAPTCHA with a probability of $\frac{1}{6C2} = \frac{1}{15}$ (i.e 6.66%) . The security level could be improved by increasing the number of displayed images, but this will results in reducing the usability level of the user interface.

Gmail uses a combination of text-based and image-based CAPTCHAs. A text is randomly generated in parallel with an image that contains a text. The user is challenged by typing all the text appearing on the screen (see Fig. 5).

Google inc [17] developed new type of captcha without solving complex problems. The user is challenged by selecting all images that correspond to the given question. In Figure 6, the user is asked to select all images with trees. Solving such captcha system is extremely difficult using machine learning techniques since it includes semantic image analysis for each of the 9 images. A brute force attack of 9 image recaptcha can succeed with probability of $\frac{1}{\sum_{i=1}^{9} 9Ci} = \frac{1}{511}$ (i.e. 0.19%). In addition to the above mentioned CAPTCHAs, many other forms

Figure 2: ASIRRA: The user is challenged by selecting all the cat images. A brute force attack can solve the problem with probability of $(\frac{1}{2})^{12}$.

Figure 3: PIX: The user is challenged selecting from a pull-down a word that relates all the images. In this case the relation is FISH

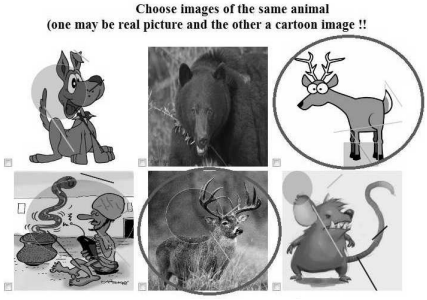


Figure 4: SEMAGE: The user is challenged by identifying the semantically related images. In this case the two encircled images are related

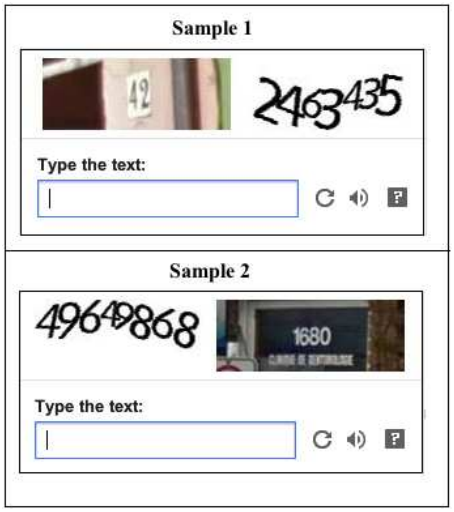


Figure 5: Gmail CAPTCHA: Gmail uses a combination of text-based and image-based CAPTCHAs. A text is randomly generated in parallel with an image that contains a text. The user is challenged by typing all the text appearing on the screen.

Figure 6: reCaptcha: Google Inc has developed a new type of CAPTCHA without solving complex problems. The captcha is solved by clicking on all images that correspond to the asked question. In this sample the use is asked to click on all images with trees.

of CAPTCHA designs have been proposed [18]. Examples of such designs are counting certain objects in a picture, segmenting facing, and identifying number of words in audio [6].

All of the above mentioned image-based CAPTCHA systems discussed in this section have one main disadvantage. This disadvantage is mainly the need of updating and maintaining a database of images. In this paper we propose a database-free image-based CAPTCHA system. The details of the proposed techniques is presented in section 4. The rest of the paper is organized as follows: section 2 presents the image correspondence problem, section 3 introduces the Möbius transformation on images, section 4 discuss our proposed captcha algorithm, section 5 provides a security analysis of our algorithm, section 6 is a usability analysis and finally section 7 is a conclusion.

2 The Correspondence Problem

The image matching problem (i.e. the correspondence problem) is defined as the establishment of the correspondence between features extracted from two or more images of the same scene. This problem is a well known ill-posed problem among researchers in the area of Computer Vision [19, 20, 21]. The main challenges of the correspondence problem are due to 1) occlusion: a point in one image may not have a corresponding match in the other image, and 2) scenes with repetitive patterns: the solution may not be unique, there may be more than one match to a feature point. For some cases the correspondence problem is too complex to solve in a reasonable amount of time and is categorized as an NP complete problem [22]. For a demonstration of the image matching problem see Figure. 7

3 The Möbius transform

Möbius transformation is any function of the form [23, 24]:

$$w = f(z) = \frac{az + b}{cz + d} \tag{1}$$

where a, b, c and d are complex or real constants and $ad - bc \neq 0$.

The Möbius transformation can be decomposed of a sequence of a dilation, translation, rotation and inversion.

Möbius transformation preserves the essential structure of the object it is defined as a projective transformation of the complex projective line and is an automorphism which map the object to itself and maps a line to a line or a circle and maps a circle to a line or a circle. The Möbius transformation is a bijective transformation and its inverse is defined as:

$$g = f^{-1}(z) = \frac{dz - b}{-cz + a} \tag{2}$$

Figure 8(a) shows a randomly generated image, Figures 8(b),(c) and (d) show different Möbius transformation using a pseudo-random function generated by selecting random constant a, b, c and d for Eq. 1.

4 Möbius CAPTCHA algorithm

Our approach is based in challenging the user using a computer generated random images. For this purpose we developed an algorithm that draws simple shapes such as circles, rectangles and lines. The attributes for each shape (circle radius, rectangle width and height or line length and orientation) are also randomly generated. The algorithm could be easily updated to enhance security and usability. In our proposed CAPTCHA approach we start by creating two empty images call them *Image* ***a*** and *Image* ***b***.

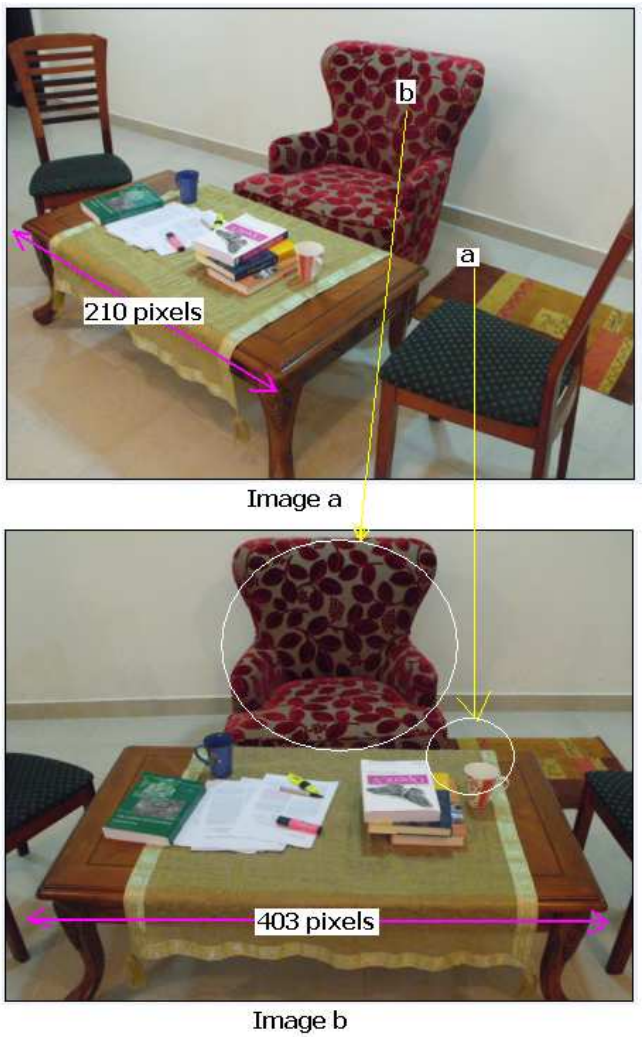
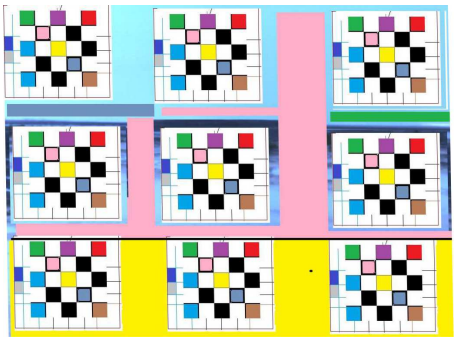
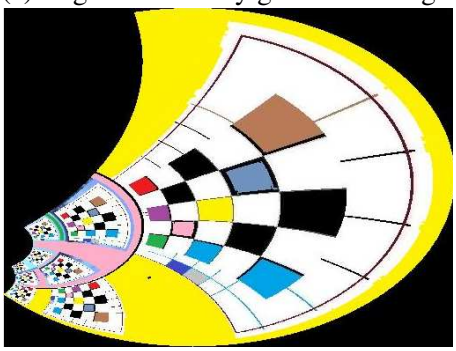


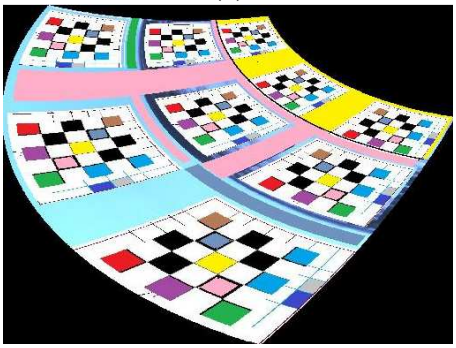
Figure 7: Image correspondence problem: matching two points in two images of the same scene is a trivial tasks for humans. For the computer it is a difficult problem, for example point **a** has no match due to occlusion, point **b** has many potential matches due to repetitive patterns. With additive noise and scaling this problem becomes even more complex for the computer.



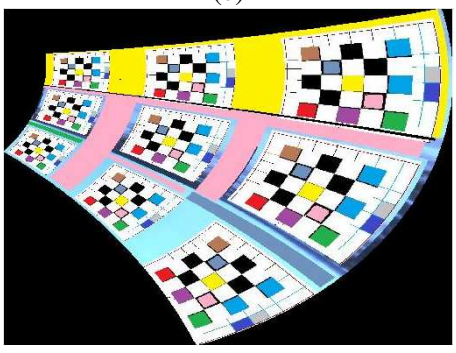
(a) original randomly generated image.



(b)

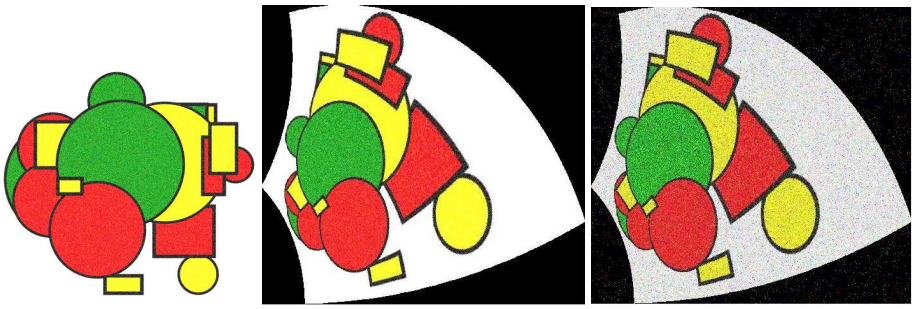


(c)



(d)

Figure 8: Möbius transformation: (a) randomly generated image (b), (c) and (d) show different Möbius transformations generated using different pseudo-random functions.



(a) (b) (c)

Figure 9: Möbius CAPTCHA: (a) randomly generated image (b) Möbius image using the function $f(z)$ as described in equation 1 (c) Möbius image with additive noise

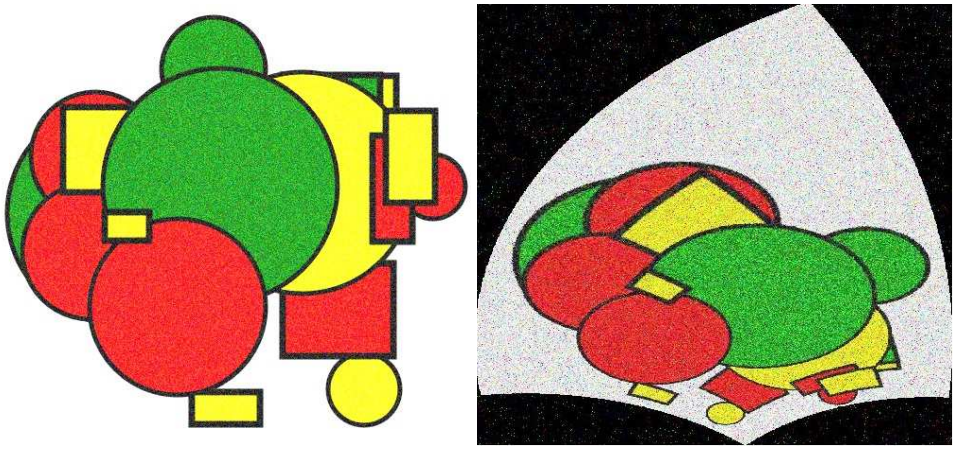
First, a random image is generated with random shapes and colors in *Image* ***a***. Then a random Möbius transformation is generated by selecting random constant a, b, c and d for Eq. 1 as shown in Figure. 9 (b). Finally noise and scaling are added to the Möbius image to produce the final image (*Image* ***b***) as shown in Fig 9 (c).

The proposed image matching CAPTCHA is performed by displaying to the user the initially generated image (*Image* ***a***) and the Möbius image (*Image* ***b***) as shown in Fig. 10. The user is then asked to match any point (call it $\mathbf{p}$) from Fig. 10(a) with its corresponding point (call it $\mathbf{p}^{'}$) in Fig. 10(b). Finding such correspondence is natural for humans as they are used to matching patterns. However this problem is extremely challenging for computer robots. This problem is know as the the correspondence problem, is one of the most challenging research tasks in the computer vision community. Image matching is the first step of a computer vision system. Many Computer Vision problems such as camera calibration [25, 26], 3D object reconstruction [27], obstacle detection [28, 29], motion estimation [30] and object tracking [31, 32] require solving the image matching problem as a fundamental step in the sequence of steps that need to be computed.

The two points are corresponding points if they satisfy Eq. 1, thus if $\mathbf{p} \approx \mathbf{f}(\mathbf{z})^{-1}\mathbf{p}'$ then the user is human otherwise the user is software robot. Our CAPTCHA approach can be summarized into the following steps:

MOBIUS CAPTCHA

- **Step 1**: Create two empty images call them *Image* ***a*** and *Image* ***b***
- **Step 2**: Generate on *Image* ***a*** random shapes and color
- **Step 3**: Select a pseudo-random function $f(z)$ and map the first image *Image* ***a*** to the second image *Image* ***b***.
- **Step 4**: Add random noise to the second image *Image* ***b***.
- **Step 5**: display Images a and b to the user and ask him/her to select any point $\mathbf{p}_1$ from *Image* ***a*** and match it with its corresponding point $\mathbf{p}_1^{'}$ in *Image* ***b***.
- **Step 6**: If $\mathbf{p_1} \approx \mathbf{f}(\mathbf{z})^{-1}\mathbf{p_1}^{'}$ then pass human user otherwise software robot.



(a) random image (b) Möbius image with additive noise

Figure 10: CAPTCHA test: the user is asked to selected one point from the original image and match it with its corresponding point in the Möbius image. This process is intuitive for humans but complex for software robots

5 Security Analysis

5.1 Computer and Machine Vision Attack

In our approach no underlying database is needed, the images are created randomly in real time. In addition the projective relation between the random image and the deformed image is a pseudo-random function H_{ab} with no known efficient algorithm to distinguish H_{ab} from all possible pseudo-random function family (PRF) that exit between the two images [33]. The total number of the PRF functions is finite but related with a polynomial relation with respect to the image resolution. For an n rows by n columns image resolution (i.e $N = n^2$ total number of pixels) if we select 4 points at random from the N image points, the possible combinations can be calculated by $\mathbf{C}_4^N = \frac{N!}{4!(N-4)!} \approx O(N^4)$. Thus the PRF family is $\approx O(N^4)$ functions.

To solve our proposed CAPTCHA problem by a software robot using a brute force attach that explores every possible function in the PRF family is computational expensive and cannot be solved in a reasonable amount of time. The probability of success is $\frac{1}{N^4}$ which is the probability of successfully selecting 4 points from *Image a* and their corresponding points in *Image b*.

Other than brute force attack approach, the other option for a software robot to solve the problem is to use feature-based matching techniques to find Möbius transformation between *Image a* and *Image b*. However, with the additive noise and scaling on the deformed images the solution becomes extremely difficult for feature matching algorithm such as [34], [35] and [36].

6 Usability Analysis

An empirical analysis was conducted in order to study the usability and measure the ease of use of our proposed captcha system. In the study ten volunteers were involved. The volunteers are mainly students from different majors and different seniority levels, however all of them are familiar with the computers and have been subjected to comparable captcha system such as re-captcha [17]. Each user was asked to solve ten captcha problems. For each test we recorder the time taken by the user to select one pixel in the random image and its corresponding match in the Möbius image. For captcha test the user can make up to three trials before displaying a new set of images. As a result of the empirical analysis the average response time was 2.76 seconds with with average success of 92%.

7 Conclusion and Future Work

In this paper we presented a novel CAPTCHA approach to overcome the main weaknesses of the CAPTCHA algorithms presented in literature [2, 6, 14, 11]. The main advantage of our approach is that it is based on image matching approach. Humans have developed an advanced matching and recognition skills compared to computer software. Matching two points of two images of the same scene (i.e the correspondence problem) is natural and intuitive for humans but extremely complex for software robots especially under additive noise and deformation.

In addition all image based CAPTCHAs with underlying database such [17, 14, 12, 11, 16] suffer from database attack. Each time a set of images is displayed a part of the image database is revealed to the attacker. With enough number of challenges a good portion of the database can be reconstructed and thus breaking the CAPTCHA becomes easier. In our approach the images are randomly generated and database reconstruction is hard and cannot be easily predicted by the attacker.

As part of future work, we propose to develop a database of captcha challenges and then apply state of the art image matching techniques on every set of images in order to assess the security level of our proposed system against potential software robot. At the same time we will conduct a comprehensive usability study of our system and compare the outcomes with other image based captcha systems. The usability study will include a large number of volunteers. Both empirical data and surveys will be collected from each volunteer in order to be used in the analysis.

References

[1] A. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, pp. 433–460, 1950.

[2] Luis von Ahn, Manuel Blum, and John Langford, "Telling humans and computers apart automatically," *Commun. ACM*, vol. 47, pp. 56–60, February 2004.

[3] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum, "reCAPTCHA: Human-Based Character Recognition via Web Security Measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, Sept. 2008.

[4] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: breaking a visual CAPTCHA," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2003, vol. 1, pp. I–134–I–141.

[5] Patrice Y. Simard, "Using machine learning to break visual human interaction proofs (hips," in *Advances in Neural Information Processing Systems 17, Neural Information Processing Systems (NIPS2004*. 2004, pp. 265–272, MIT Press.

[6] Joshua T. Goodman and Robert Rounthwaite, "Stopping outgoing spam," in *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, New York, NY, USA, 2004, pp. 30–39, ACM Press.

[7] Kumar Chellapilla, Kevin Larson, Patrice Simard, and Mary Czerwinski, "Designing human friendly human interaction proofs (hips)," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2005, CHI '05, pp. 711–720, ACM.

[8] "Death by captcha," *http://www.deathbycaptcha.com*, accessed April 2016.

[9] "decaptcher," *http://www.decaptcher.com*, accessed April 2016.

[10] "Bypass," *http://www.bypasscaptcha.com*, accessed April 2016.

[11] Monica Chew and J. D. Tygar, "Image recognition captchas," Tech. Rep. UCB/CSD-04-1333, EECS Department, University of California, Berkeley, Jun 2004.

[12] Luis von Ahn and Laura Dabbish, "Labeling images with a computer game," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2004, CHI '04, pp. 319–326, ACM.

[13] Jared Saul, "Petfinder," *http://www.petfinder.com*, accessed April 2016.

[14] Jeremy Elson, John R. Douceur, Jon Howell, and Jared Saul, "Asirra: a captcha that exploits interest-aligned manual image categorization," in *ACM CCS*, 2007, pp. 366–374.

[15] Philippe Golle, "Machine learning attacks against the asirra captcha categories and subject descriptors," *ReCALL*, p. 1, 2009.

[16] Shardul Vikram, Yinan Fan, and Guofei Gu, "Semage: a new image-based two-factor captcha.," in *ACSAC*, Robert H'obbes' Zakon, John P. McDermott, and Michael E. Locasto, Eds. 2011, pp. 237–246, ACM.

[17] Google inc, "recaptcha," *https://www.google.com/recaptcha/intro/index.html*, accessed Dec 2016.

[18] Manuel Blum Luis von Ahn and John Langford., "Petfinder," *The Captcha Project homepage: http://www.captcha.net*.

[19] O. Faugeras, *Three-Dimensional Computer Vision, A geometric Viewpoint*, MIT Press, Cambridge, MA, 1996.

[20] Desire Sidibe, Philippe Montesinos, and Stefan Janaqi, "Matching local invariant features with contextual information: An experimental evaluation," *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, vol. 7, no. 1, 2009.

[21] Kam Meng Goh, Syed Abu-Bakar, Musa Mokji, and Usman Sheikh, "Enhanced rotational feature points matching using orientation correction," *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, vol. 13, no. 1, 2014.

[22] Daniel Keysers and Walter Unger, "Elastic image matching is np-complete," *Pattern Recogn. Lett.*, vol. 24, no. 1-3, pp. 445–453, Jan. 2003.

[23] R. Schinzinger and P.A.A. Laura, *Conformal Mapping: Methods and Applications*, Dover books on mathematics. Dover Publications, 2003.

[24] Z. Nehari, *Conformal Mapping*, International series in pure and applied mathematics. McGraw-Hill, 1952.

[25] C. Matsunaga anf K. Kanatani, "Calibration of a moving camera using a planar pattern: optimal computation, reliability evaluation and stabilization by model selection," *Proc. 6th Euro. Conf. Computer Vision, Dublin, Ireland*, vol. 2, pp. 595–609, 2000.

[26] B. Triggs, "Autocalibration from planar scenes," in *In Proc. European Conference on Computer Vision*, 1998, pp. 89–105.

[27] E. Trucco and A. Verri, *Introductory Techniques for 3D Computer Vision*, Prentice-Hall, New Jersey, 1998.

[28] Z. Zhang, R. Weiss, and A.R. Hanson, "Obstacle detection based on qualitative and quantitative 3d reconstruction," *IEEE trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 15–26, Jan 1997.

[29] R.Elias H.Hajjdiab and R.Laganière, "Wide baseline obstacle detection and localization," in *in Proceedings of the IEEE Seventh International Symposium on Signal Processing and its Applications*, Paris-France, July 2003, vol. 1, pp. 21–24.

[30] P. Fornland, "Direct obstacle detection and motion from spatio-temporal derivatives," in *CAIP*, Prague,Czech Republic, Sept 1995.

[31] T. Williamson and C. Thorpe, "A specialized multibaseline stereo technique for obstacle detection," in *IEEE conf. on Computer Vision and Pattern Recognition*, 1998, pp. 238–244.

[32] S. Carlsson, "Recognizing walking people," in *European conf. on Computer Vision*, 2000.

[33] Oded Goldreich, Shafi Goldwasser, and Silvio Micali, "How to construct random functions," *J. ACM*, vol. 33, no. 4, pp. 792–807, Aug. 1986.

[34] P. Montesinos, V. Gouet, R. Deriche, and D. Pel, "Matching color uncalibrated images using differential invariants," *Image and Vision Computing*, vol. 18, no. 9, pp. 659–672, June 2000.

[35] J.Y. Zheng, "Acquiring 3d models from sequences of contours," *IEEE trans. on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 163–178, 1994.

[36] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.