Location Selection for Versatile Characters of Image Based CAPTCHA

Chen-Chiung Hsieh, Yu-Chen Lin, Kai-Miao Cha, and Chao-Wei Huang
Department of Computer Science and Engineering, Tatung University, Taipei City, Taiwan
Email: cchsieh@ttu.edu.tw, {f1285709, terryfeverpitch}@gmail.com, cammy62502005@yahoo.com.tw

***Abstract*—A CAPTCHA is a type of challenge-response test used to determine whether the user is human or not. This user identification procedure has received many criticisms from people who feel the distorted words are illegible even for users with no disabilities. In order to prevent website owner from suffering attacks and simultaneously maintain the readability of generated CATCHA, this paper proposed an improvement of image-based CAPTCHA which embeds versatile characters in the images for distinguishing human and computer. The improvement selects the locations for characters smartly by a salient detection of the used image. The greater salient location is picked to embed the versatile characters. By calculating the integral image of the salient image, we can decide the appropriateness of the selected location quickly. The proposed method makes the characters indiscernible by automated pattern recognition technologies like optical characters recognition while human can easily distinguish the location of the embedded characters. In experiments, 10 users were invited to test the system and the success rate is about 95.6%. Compare the average logging time with previous studies, the proposed method is also faster.**

***Index Terms*—CAPTCHA, internet verification code, image analysis, salient detection**

I. Introduction

Throughout the years, CAPTCHA has provided us with a number of ways to answer the same question. Why would a website want to verify this? To prevent spam emails. CAPTCHA originally developed by Carnegie Mellon researchers and professors in 2000 provided webmasters with a solution to the spam that plagued their sites. By generating an image containing a string of random, warped characters, CAPTCHA forms stopped automated spam bots in their tracks. Consequently, CAPTCHA [1] verification mechanism was applied to man-machine distinguishing. It is mainly with an automatically generated test mechanism where an individual person can easily pass through while automated program can hardly pass through on the contrary.

However, spammers quickly adapted their automation software to bypass this kind of test by pattern recognition technologies: remove a CAPTCHA's background, separate and identify individual characters, and ultimately type in the displayed character string. In this study, text based CHPTCHA is improved from a 1D distorted string to a 2D versatile characters by displaying individual character with varying size, color, style, and angle on a randomly selected background image. The most difficult part for automated process is to recognize each versatile character that is interfered by non-uniform background image. To ensure the robustness of selected locations for the characters, salient detection is deployed to expose the important parts of the background image. By integral image [2], the suitability of selected position could be verified quickly. In order not to affect the original image much, the displayed characters are of different color from the selected block image. To prove the robustness of proposed mechanism, pattern recognition technique SURF (Speeded Up Robust Features) [3], a speeded-up version of SIFT (Scale-invariant feature transform) [4] is tested and the system could successfully resist this kind of attack. Meanwhile, human users can easily complete the verification task through only mouse clicks.

II. Related Works

Rusu and Govindaraju [5] proposed handwritten CAPTCHA as shown in Fig. 1, where the alphabetic characters displayed are handwritten style. Users need to identify alphabetic letters in the image and enter the correct answer to pass through verification. The letters displayed in this simple background image can be easily analyzed by computers. On the other hand, some of the characters are distorted too much for user to identify and answer in a short time.

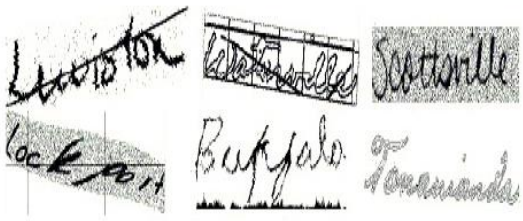


Figure 1. Verification mechanism using handwritten characters in a noisy background [5].

Baird and Bentley proposed implicit CAPTCHA method [6], where the verification mechanism only requires user an easy click. Firstly, the system shows a picture, an example as shown in Fig. 2, and informs user on basis of the program to click on the correct location.

Manuscript received April 23, 2015; revised July 14, 2015.

doi: 10.18178/ijsps.4.3.239-244

User must click on the correct location to pass through the verification process. However, such verification system is easy to be solved by semantic analysis so that safety protection could not be guaranteed.

Figure 2. Implicit CAPTCHA [6].

In Microsoft Hotmail website registration [7], a string of randomly selected letters from alphanumeric set is displayed with varying shape or color for individual character as shown in Fig. 3. That verification could not be easily passed through by automatic software, and certainly such interfering information could also create confusion to users. Thus, user needs to wait for the new CAPTCHA that can be easier identified to complete the verification.

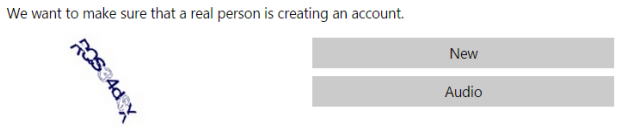


Figure 3. Verification mechanism for Hotmail registration [7].

Figure 4. Different orientations of the sub-images in Image Flip CAPTCHA [8].

Banday and Shah [8] proposed an image-based CAPTCHA named image flip CAPTCHA. It is safer than text-based CAPTCHA by deploying more than dozen of images displayed together. Due to the complexity of images, pattern recognition programs like content-based image retrieval (CBIR) [9] could not easily figure out the image contents. In addition, several vertical/horizontal lines are inserted as interferences as shown in Fig. 4. According to the given short guideline, user need to click the right images with indicated orientation such as portrait or landscape. The advantage is time saving by mouse clicks while doing verification. However, this system requires a great amount of images so that the images would not repeat frequently. Furthermore, time delay caused by transmitting the chosen images for combination.

Hsieh and Wu [10] proposed an innovative image-based CAPTCHA for distinguishing human and computer by embedding versatile characters in the images. However, the location selection for the versatile character is random based as shown in Fig. 5 which could not ensure not to be recognized by OCR attack. In this paper, the location is selected intelligently using gradient values and ensured by checking the reserved block is content abundant. Our designed mechanism is capable to elude the state-of-the-art OCR attacks since the characters are mixed independently in the image.

Figure 5. Some characters are placed on the uniform background [10].

III. Proposed Methods

The system flow of the verification mechanism, as shown in Fig. 6, firstly selects a background image randomly from image bank. Then, five verification codes from alphanumeric set are also selected randomly but to avoid repeating two times. The location selected must ensure the characters could be displayed completely within the image and avoid collision with other anchored verification characters.

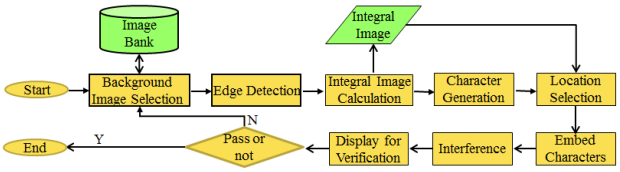


Figure 6. System flow diagram.

A. Background Image and Verification Codes Generation

Arbitrarily select a background image from a non-restricted image database. Then randomly generate five alphanumeric characters and embed these characters into the image. These five alphanumeric characters with varying size, color, style, and coordinates are displayed on web page for user to verify. In the verification process, user needs to select characters according to the sequence of the verification codes displayed in the hint box. When a character is clicked, its color will be changed. Verification is completed once all the verification codes are clicked without errors. If error occurred then a

verification fail message will be displayed and the system will restart a new verification process until the user passes through it.

The background image arbitrarily selected from the image database is brought into web page using JQuery Ajax. 26 uppercase and lowercase letters and 10 numeric digits are used as the character set. Five verification codes are generated in random fashion without duplication more than twice, wherein make variations on their size, color, and style. Each character is also rotated from the normal oriental direction.

B. Location Selection Using Edge Detection

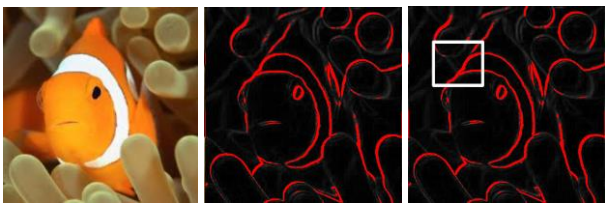
The prominent parts of an image are the better locations to anchor CAPTCHA. By way of the complex characteristic of that area, CAPTCHA could be protected from automation process while human can easily recognize it. To find out these areas, gradient value is adopted to measure the importance of each pixel. Here, Sobel [11] edge detection operator is used in this paper. It is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. The operator uses two 3×3 kernels as shown in Fig. 7(a) and Fig. 7(b) which are convolved with the original image to calculate approximations of the derivatives - G_x for horizontal changes, and G_y for vertical. The derivative at that pixel is calculated by (1).

$$G = \sqrt{(G_x^2 + G_y^2)} \tag{1}$$

−1	2	−1	−1	0	1
0	0	0	−2	0	2
1	2	1	−1	0	1

(a) (b)

Figure 7. Sobel edge detection. (a) Kernel for vertical gradient G_y. (b) Kernel for horizontal gradient G_x.



(a) (b) (c)

Figure 8. Gradient values. (a) Original image. (b) Gradient values of (a). (c) One of the selected CAPTCHA locations.

The background image is usually of color image so that we need to convert it to gray scale by (2) for edge detection. The larger gradient represents the more important of the locations. Pixels with gradients greater than a given threshold are reserved for the CAPTCHA location selection. Fig. 8 gives an example of the Sobel edge detection.

$$Gray = R \times 0.3 + G \times 0.59 + B \times 0.1 \tag{2}$$

C. Integral Image Generation

An integral image [2] is a data structure for quickly and efficiently generating the sum of values in a rectangular subset of a block. It was first prominently used within the Viola-Jones [2] object detection framework in 2001. As the name suggests, the value at any point (x, y) in the summed area table is just the sum of all the pixels above and to the left of (x, y) as in (3). Moreover, the summed area table can be computed efficiently in a single pass over the image, using the fact that the value in the summed area table at (x, y) is just as calculated in (3). Once the summed area table has been computed, the task of evaluating any rectangle (ABCD) as the shown in Fig. 9 can be accomplished in constant time by (4) with just four array references.

$$In(x, y) = \sum_{x' \le x, y' \le y} I(x', y') \tag{3}$$

$$\sum_{\substack{x_0 < x \le x_1 \\ y_0 < y \le y_1}} I(x, y) = In(D) - In(B) - In(C) + In(A) \tag{4}$$

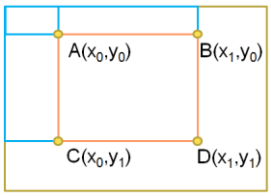


Figure 9. A sum of table called integral image that is used to calculate the sum of any block efficiently.

Since the location is picked randomly that needs to be checked by the integral image using the block of generated CAPTCHA character. As shown in Fig. 9, the middle red rectangle is the size of generated CAPTCHA character. The complexity of that rectangle could be calculated using (4). If the complexity value is greater than a given threshold, the CAPTCHA character is displayed at that location with background mixed using alpha setting as in (5).

$$CI(x, y) = \alpha \times C(x, y) + (1 - \alpha) \times BI(x, y) \tag{5}$$

where CI, C, and BI are the CAPTCHA image, CAPTCHA character, and background image, respectively. Note that α is the transparency for mixing two images. In addition, several straight lines are also generated to interfere the automation process.

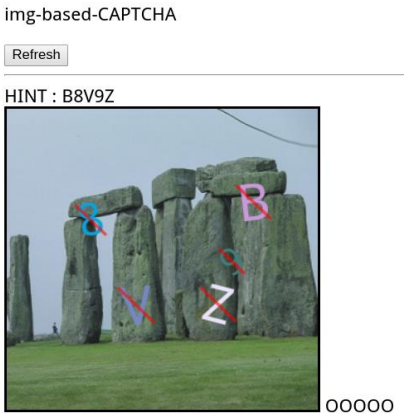


Figure 10. An example of generated CAPTCHAs.

Fig. 10 gives an example of generated image with verification code embedded. The verification code in normal format is hinted above the image and user needs to click these characters according to the sequence in the hint box.

D. SURF Character Matching

Scale-Invariant feature transform (SIFT) [4] is an algorithm in computer vision to detect and describe local features in images. Lowe's method for image feature generation transforms an image by gradient operator as in Fig. 11(a) into a large collection of key points with feature vectors as shown in Fig. 11(b). Each key point is invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric distortion. SIFT feature point comprises four steps: scale-space extreme value detection; feature point position optimization; calculating feature point directionality; feature point description.

Speeded Up Robust Features (SURF) [3] is a robust local feature detector proposed by Herbert Bay *et al*. It is partly inspired by the SIFT descriptor. The standard version of SURF is several times faster than SIFT and claimed to be more robust against different image transformations than SIFT. SURF is based on sums of 2D Haar wavelet responses and makes an efficient use of integral images.

It uses an integer approximation to the determinant of Hessian blob detector, which can be computed extremely quickly with an integral image (3 integer operations). For features, it uses the sum of the Haar wavelet response around the point of interest. Again, these can be computed with the aid of the integral image. This information is treated to perform operations such as locate and recognition of certain objects, people or faces, make 3D scenes, object tracking and extraction of points of interest. This algorithm is applied as the OCR engine to do CAPTCHA character recognition.

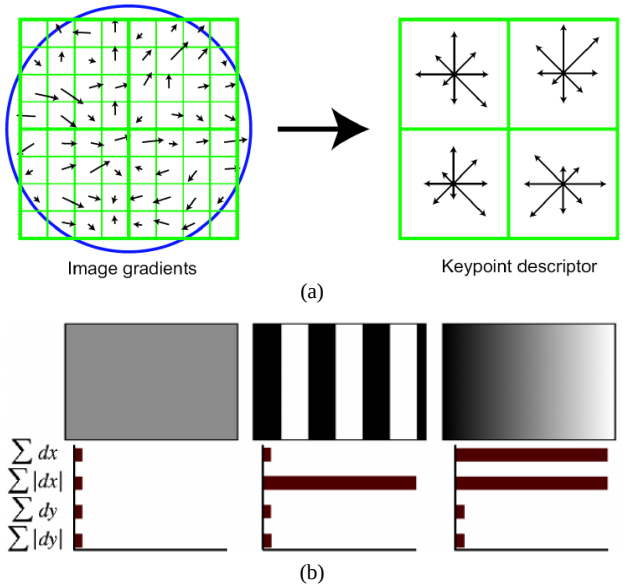


Figure 11. (a) SIFT feature point description diagram. (b) SURF feature point description diagram [3].

IV. Experimental Results

A. User Verification Test

For the test of the system, users need a computer connected to the developed web server. 10 subjects aged among 23~27, male and female were invited to conduct three experiments. In Experiment one, each subject conducted 10 times the verification. To avoid misunderstanding caused operation errors, each subject was told the standard verification procedure in advance. Table I summarized the results and it demonstrated that our method surpassed reCAPTCHA's [12] and HELLO CAPTCHA [13] by 80% and 88% [10], respectively.

TABLE I. Pass Rates for the Proposed Verification Mechanism.

User No.	Success	Failure
1	10	0
2	8	2
3	8	2
4	10	0
5	10	0
6	10	0
7	10	0
8	10	0
9	10	0
Total	86	4
%	95.56%	4.44%

Our verification time was 6.7 seconds which was lower than reCAPTCHA's 13.4 seconds and HELLO CAPTCHA's 21.7 seconds [10]. The proposed verification code mechanism is superior to reCAPTCHA due to the distorted characters in reCAPTCHA were too difficult for user to read. The situation was worse in HELLO CAPTCHA owing to occluded characters in dynamic presentation. In addition, downloading time was also time-consuming. Instead of requiring more time for user to recognize the text so as to input verification codes, the proposed mechanism only requires simple clicks to pass the verification. It is able to save time spending on web.

TABLE II. Success Rates Based on Characters.

User	Success	Failure
1	49	1
2	46	4
3	47	3
4	49	1
5	50	0
6	49	1
7	50	0
8	50	0
9	48	2
Total Characters	438	12
%	97.33%	2.67%

To investigate the factors which caused failure, Experiment three was conducted by asking each subject finding the five characters 10 times without aborting on error. Hence, each subject totally needed to find 50 characters from 10 images. The results were given in Table II and the failure reasons include: background of similar color, background of high illumination, background with duplicate characters, complex background, and characters happened to fit in background.

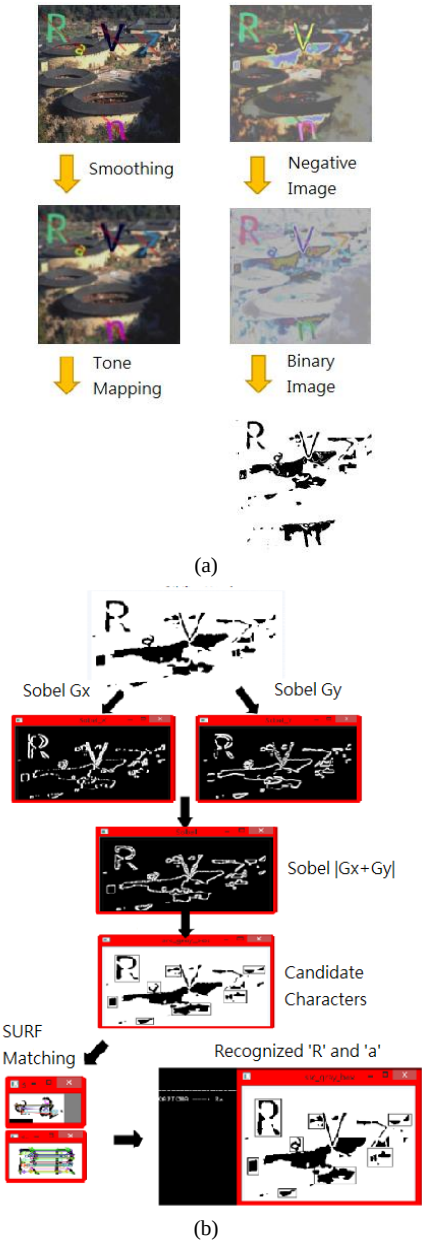


Figure 12. SURF attacks. (a) Character detection of the verification codes using standard OCR procedure. (b) SURF matching results.

B. SURF Attacks

This experiment was mainly to verify whether the proposed mechanism can be easily conquered or not. The more number of matched key points, the more possible be attacked. Fig. 12 gives an example showing SURF matching of the verification codes. It was found that only 'R' and 'a' could be detected due to the effects of selected complex parts of background. On the other hand, once the deployed background image was simple enough, then the verification codes of the proposed mechanism may not under protection.

V. Conclusions

Along with the widespread of internet usage, people rely on automatic mechanism to process massive data day by day. To avoid the automatic mechanism to be misused and reduce any improper use of operation, we need to distinguish between man and machine so as to prevent the risk brought by automation. Rights and interests of users could be effectively maintained only if man-machine difference could be differentiated.

In this paper, an improvement on image based CAPTCHA using versatile characters is proposed which could avoid placing characters on uniform regions while preserving human visibility. User only needs to click the codes according to the displayed text. From experimental results, the pass rate by human is higher than some existing approaches. Meanwhile, the human verification time of our method is quite shorter than the other approaches. To test the robustness of the proposed method, famous SURF matching algorithm was adopted to attack the system and the experimental results showed that the mechanism is capable of defending such kind of attacks.

Acknowledgment

The research work was supported by Ministry of Science and Technology, Taiwan, under Grant No. MOST 103-2221-E-036 -017, 2014.

References

[1] L. Ahn, M. Blum, N. J. Hopper, and J. Langford, "CAPTCHA: Telling humans and computers apart automatically," in *Proc. Eurocrypt '03 Advances in Cryptology, Lecture Notes in Computer Science*, 2003, vol. 2656, pp. 294-311.

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. 511-518.

[3] B. Herbert, E. Andreas, T. Tinne, and G. L. Van, "SURF: Speeded up robust features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346-359, 2008.

[4] D. G. Lowe, "Distinctive image features from scale-invariant key points," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[5] A. Rusu and V. Govindaraju, "Handwritten CAPTCHA: Using the difference in the abilities of humans and machines in reading handwritten words," in *Proc. 9th Int'l Workshop on Frontiers in Handwriting Recognition*, Sept. 2004, pp. 226-231.

[6] H. S. Baird and J. L. Bentley, "Implicit CAPTCHAs," in *Proc. SPIE on Document Recognition and Retrieval XII*, 2004, vol. 5676, pp. 191-196.

[7] Microsoft Hotmail website registration. [Online]. Available: https://signup.live.com/signup?uaid=4b1c4131c411477582695ddfa6b95555&lic=1

[8] M. T. Banday and N. A. Shah, "Image flip CAPTCHA," *ISeCure, The ISC International Journal of Information Security*, vol. 1, no. 2, pp. 105-123, Jul. 2009.

[9] F. Long, H. Zhang, and D. Feng, *Fundamentals of Content-Based Image Retrieval*, Springer, 2003, ch. 1.

[10] C. C. Hsieh and Z. Y. Wu, "Anti-SIFT images based CAPTCHA using versatile characters," in *Proc. International Conference on*

Information Science and Applications (ICISA), Suwon, Jun. 2013, pp. 1-4.
[11] R. Gonzalez and R. Woods, *Digital Image Processing*, Boston, USA: Addison Wesley, 1992, pp. 414-428.
[12] ReCAPTCHA. [Online]. Available: http//www.google.com/recaptcha
[13] HELLO CAPTCHA. [Online]. Available: http://hellocaptcha.com/index.php

Chen-Chiung Hsieh received his B.S., M.S., and Ph.D. degrees in the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 1986, 1988, and 1992, respectively. During Dec. 1992 to Jan. 2004, he was with the Institute for Information Industry (III) as a vice director. From Dec. 2004 to Jan. 2006, he joined Acer Inc. as a senior director. He is presently an Associate Professor in the Department of Computer Science and Engineering at Tatung University, Taipei, Taiwan. His research area is mainly focused in image and multimedia processing.

Yu-Chen Lin received his B.S. in the Department of Computer Science and Engineering, Tatung University, Taipei, Taiwan, in 2015. His research interests include image processing and character recognition.

Kai-Miao Cha received her B.S. in the Department of Computer Science and Engineering, Tatung University, Taipei, Taiwan, in 2015. Her research interests include image processing and multimedia processing.

Chao-Wei Huang received his B.S. in the Department of Computer Science and Engineering, Tatung University, Taipei, Taiwan, in 2015. His research interests include image processing and CAPTCHA.