Open access • Proceedings Article • DOI:10.1109/SIPS.2008.4671755

Bio-inspired unified model of visual segmentation system for CAPTCHA character recognition — Source link

Chi-Wei Lin, Yu-Han Chen, Liang-Gee Chen

Institutions: National Taiwan University

Published on: 17 Nov 2008 - Signal Processing Systems

Topics: Intelligent character recognition, Intelligent word recognition, Three-dimensional face recognition, Optical character recognition and CAPTCHA

BIO-INSPIRED UNIFIED MODEL OF VISUAL SEGMENTATION SYSTEM FOR CAPTCHA CHARACTER RECOGNITION

Chi-Wei Lin, Yu-Han Chen, and Liang-Gee Chen

DSP/IC Design Lab, Graduate Institute of Electronics Engineering and
Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan

ABSTRACT

In this paper, we present a bio-inspired unified model to improve the recognition accuracy of character recognition problems for CAPTCHA (Completely Automated Public Turing Test to Tell Computers and Humans Apart). Our study focused on segmenting different CAPTCHA characters to show the importance of visual preprocessing in recognition. Traditional character recognition systems show a low recognition rate for CAPTCHA characters due to their noisy backgrounds and distorted characters. We imitated the human visual attention system to let a recognition system know where to focus on despite the noise. The preprocessed characters were then recognized by an OCR system. For the CAPTHA characters we tested, the overall recognition rate increased from *16.63%* to *70.74%* after preprocessing. From our experimental results, we found out the importance of preprocessing for character recognition. Also, by imitating the human visual system, a more unified model can be built. The model presented is an instance for a certain type of visual recognition problem and can be generalized to cope with broader domains.

1. INTRODUCTION

Intelligent recognition has been an important part in the research of artificial intelligence. The goal for intelligent recognition is to complete a system that receives, processes, and analyzes signals like the way humans do. Signals for recognition can be categorized into different types, such as images or audios. Our main focus is to build up an intelligent system to improve character recognition, in hope of generalizing such methods to solve other visual recognition problems.

Character recognition has been a problem around for decades. In the past, there were many solutions focused on enhancing the efficiency of their algorithms to strike for high recognition rate. In our work, instead of searching for a more difficult algorithm, we turn to imitate the functions of the human early vision system. Our goal is to make this bio-inspired intelligent machine a unified model, just like the human visual system.

One of the most commonly used solutions for character recognition, the OCR system, has existed for more than 50 years [1]. OCR systems can solve a large amount of machine-printed characters with high recognition rate of more than 80%. They use algorithms that take advantage of order and direction of line segments of known inputs. On such premise, users may sometimes be restrained to only use letters with specific shapes or fonts. These OCR systems will have some problems when dealing with hand-printed documents. Meanwhile, there are other methods that aim at hand-printed or distorted characters such as [2] and [3]. While recognition of neat and clean hand-printed characters can be achieved, the accuracy for solving noisy images is still an open question. All these previous methods are sufficient for their own purposes, but they omitted other types of problems. For characters in a noisy background, directly using these methods may not produce expected results.

Fig. 1. An example of character CAPTCHA.

The reason causing this problem is that previous character recognition systems mainly focus on individual character identification. Such systems still possess the function of separating characters from the background, such as early methods proposed [4]. In this research, machine printed characters with regular intervals between them are dealt with. In general, however, characters do not have regular intervals, and sometimes the intervals are even difficult to distinguish. For a complete separation of characters and backgrounds, other features of characters like color contrasts, length ratios, and size proportions, should be considered. Character CAPTCHA is a combination of these features. CAPTCHA is a program built to separate machines from human beings [5]. Researchers of CAPTHCA try to construct patterns that are difficult for machines to break, but easy for human beings to solve [6]. Among all kinds of CAPTCHAs provided, a set of character CAPTCHA is built to block traditional character recognition machines. A character CAPTCHA may have complicated patterns with noisy backgrounds, as shown in Fig. 1. In this figure, we can easily find out where to focus, while the systems mentioned above cannot.

The human visual system is not much like those traditional machines. In the human visual system, there are two major stages for recognition. The first stage is early vision. At this stage, our visual system finds out where to focus to get the information we need. What it does is to segment out potential candidates from noisy backgrounds for high-level processing. The second stage of recognition is identification. After information is preprocessed in the first stage, much smaller amount of information is sent up for identification. When identification goes in process, knowledge from higher-level cortex feeds back to revise the early vision process. To recognize characters in noisy backgrounds, only using

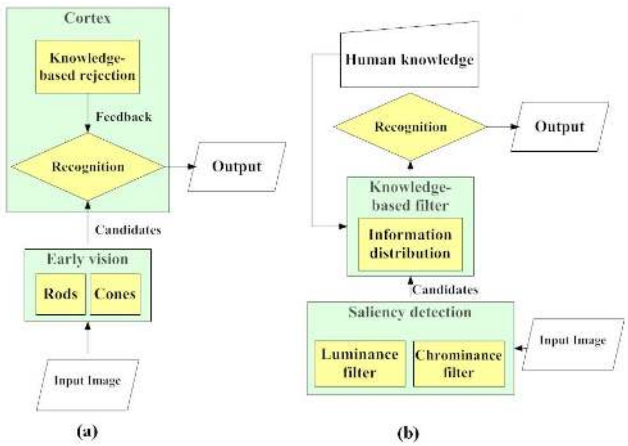


Fig. 2. The flow diagram of human visual system versus machine. (a) Process of the human visual system (b) Process of the intelligent system for recognition.

reliable identification techniques is not sufficient. The segmentation procedure is also an indispensable part. In our method, the image first passes two parallel paths, representing the early vision phase of a human visual system. Then they are connected to a high-level knowledge-based filter to complete the preprocessing. Identification results of preprocessed images will be shown in the last section to compare with those that are not preprocessed. The overall recognition rate increased from *16.63%* to *70.74%* after using our system for preprocessing.

2. PROBLEM DESCRIPTION

Traditional OCR systems or others systems could have poor performance when directly facing CAPTCHA characters due to segmentation problems. Over the past few years, there have been some researches on character recognition dealing with CAPTCHA characters. They emphasized on creating different techniques for different aspects of the problem. Some of the researches on CAPTHCA characters did not focus on segmentation. Also, most of the methods presented were specific to their own problem sets, but not general enough to solve others. In our research, we use different kinds of CAPTCHA characters as our experiment subject and try to solve the segmentation problem as a whole.

Mori and their team claimed to have broken some certain kind of CAPTCHA, EZ-Gimpy, with recognition rate more than 90% [7]. EZ-Gimpy is formed of lexical words with a black foreground and a colorful background. With such features, the characters of EZ-Gimpy can be easily segmented. Moy and others have also conducted a research on recognizing EZ-Gimpy [8]. Their purpose was to do character recognition by estimating distortion of characters. This research provides a useful technique for identification. However, these two projects focused mostly on the identification part rather than the segmentation part of character recognition.

Another research coping with CAPTCHA characters is presented by Microsoft [9]. In this research, various types of CAPTCHAs were broken with acceptable recognition rates. They dealt with each of their problems case by case and made different solutions for each one of them. Although they emphasized on segmentation, their purpose was not to break various types of CAPTCHAs with a unified method.

There are existed methods to break specific CAPTCHAs. However, breaking CAPTCHAs should be done in a more general way instead of using preprogrammed systems. The purpose of breaking CAPTCHAs is meant to improve intelligent AI systems to be more like humans. Our goal is to build a unified and efficient model to solve character segmentation problems. Also, concepts of some basic biological features are put into the system instead of creating more complicated algorithms for high recognition rate.

3. SYSTEM ARCHITECTURE

We propose three methods to build up our system for preprocessing a given CAPTCHA. The methods were inspired from the human visual perception system. With the combination of these three methods, we tried to build a unified algorithm that can adapt to various cases.

The first stage of the human visual recognition system contains photoreceptors sensitive to light. As shown in Fig. 2(a), photoreceptors of early vision are functionally classified to deal with the perception of luminance and chrominance [10]. Parallel paths behind these two kinds of photoreceptors are connected to the cortex [11]. At the recognition stage, knowledge from the cortex feeds back and meets in the middle with the received information. False information will be rejected and adjustment will be done at the early vision stage. The whole recognition process is completed with feed-forward information from early vision and feed-back knowledge from the cortex.

In our work, we built two direct receptors, luminance-based filter and chrominance-based filter, as shown in Fig. 2(b). These two filters are for saliency detection. The saliency detection part of recognition is a mapping of early vision. The candidates from saliency detection are sent to a knowledge-based filter for further processing. The human knowledge used here can be manually inputted or learned by training data. Knowledge-based filter uses given knowledge to determine potentially meaningful information.

4. IMPLEMENTATION

4.1. Luminance-based Filter

The goal of luminance-based filter is to separate the entire foreground from the background that are highly contrastive in luminance. For most CAPTCHAs, the entire foreground usually has a similar intensity of luminance. The foreground luminance often shows high contrast to the background. The obvious difference between the background and foreground is there for us to separate them apart. See Fig. 3. for examples of such CAPTCHAs. In this method, we utilize the feature of luminance difference for segmentation.

We first set a threshold for luminance to decide which part to filter out and which part to stay. The initial value of the threshold is 127, which is right in the middle of full luminance value (255) in YUV standard. Two types of luminance-based filters are used here. One is for filtering out light backgrounds, as in Fig. 3; the other is for filtering out dark backgrounds, leaving light foregrounds left. The pseudo code for filtering out a light-colored background images is shown in Fig. 4.

Fig. 3. A CAPTCHA with high luminance contrast between foreground and background.

```
Luminance-based filter for light background (original_image)
{
    Set threshold = 127;
    Set new_image as original_image with Y-values of pixels
        above threshold filtered out;
    //filter out pixels that are too light
    While ( background of new_image is not filtered out ) do
        {
        Set change_threshold = threshold/2;
        Set threshold = threshold - change_threshold;
        Set new_image as original_image with Y-values of pixels
            above threshold filtered out;
        }
}
```

Fig. 4. Pseudo code of luminance-based filter for filtering out light-colored backgrounds.

The criterion to determine which part is the background depends on the distribution of un-filtered pixels. Connected pixels distributed beyond a certain range should be regarded as the background. Therefore, the luminance-filter changes its threshold until objects of unacceptable sizes are filtered out.

4.2. Chrominance-based Filter

There are other types of CAPTCHAs that by depending on the contrast of luminance cannot solve. As shown in Fig. 5(a), a colorful image is shown. The characters inside can be easily recognized by their color contrast. Fig. 5(b) shows the same image in gray scale. The characters in Fig. 5(b) immerged inside the background. This means that the luminance for the whole Fig. 5(b) is almost the same. In this case, chrominance should be the criterion rather than luminance for segmentation.

The chrominance-based filter uses RGB information for segmenting CAPTCHA characters. Here we consider local information difference rather than the contrast of the foreground and the background as a whole. The reason is that colors can have a more complicated combination compared with mere gray-scale luminance. Therefore, colors can create various clusters that are separable, while for luminance alone, we only have to use the overall intensity difference to separate information.

(a)

(b)

Fig. 5. A color CAPTCHA and its gray-scale form. (a) Segmentation can be done by the contrast of colors. (b) The characters immerge into the background at gray-scale.

In this method, we use color differences of pixels to determine connected-components. The RGB distances of adjacent pixels are first calculated. Then pixels with distances below a given threshold are labeled as same connected-components. Sharp changes in color will be recognized as edges of clusters for segmentation. Salient parts in the image, like characters or noise clusters can be grouped together; backgrounds are prone to be connected as expanding areas.

4.3. Knowledge-based Filter

To imitate how human beings recognize things, both methods mentioned above have to be built on our own knowledge. There are various properties of characters that we use as knowledge here. Such knowledge can be inputted manually or trained by samples.

Characters should have certain shapes. Their width-length ratios should be in some range. The width-length ratio of a character cannot be too large, or else it may look like a long line. For example, a stretched 'w' might have maximal width-length ratio. On the other hand, character 'l' has the smallest width-length ratio.

The proportion for a connected-component in a frame should not be too large or too small to be a character. If the proportion of an object is too small, it will seem unimportant and will be regarded as noise; if it is too large, it should be regarded as part of the background.

Distribution of information is another feature to separate noise from characters. For example, a linear noise may be wide-spread, while its total information amount may not be too large. Such connected-components are noisy lines or curves from the background. They can be taken away by a simple filter traversing through the whole frame or the algorithm described in [12].

In the knowledge-based filter, connected-components that are invalid for the reasons above will be filtered out. With given knowledge mentioned above, the system is able to tell characters from noise and background. Luminance-based filter and chrominance-based filter work in parallel in the first stage. After that, both of them are connected to a high-level knowledge-based filter. At the second stage, the knowledge-based filter

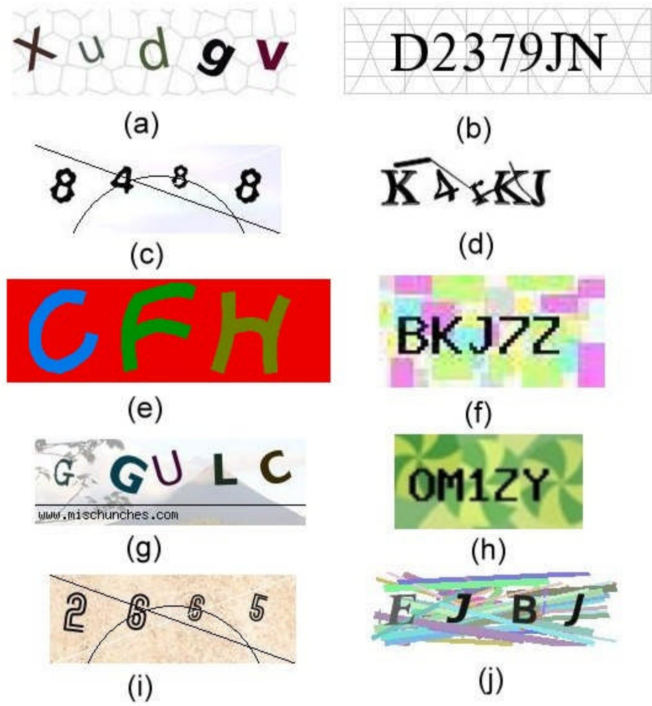


Fig. 6. Character CAPTCHA from different sources have different fonts and different background textures. Most of the characters can be segmented by our intelligent system.

determines which connected-components should be used. The chosen ones are sent to an OCR system or other intelligent systems for identification. Final results are fed back to determine whether the selection of the knowledge-based filter is correct.

5. RESULTS AND DISCUSSTIONS

We tested our methods on 10 types of CAPTCHAs from different sources, as shown in Fig. 6. The CAPTCHAs are different from one another in fonts or background textures. In this section, we will first present some preprocessed graphical results in binary output form to show the functions of each methods. After that, both original images and preprocessed images are scanned by a commonly used OCR system. Finally, numerical recognition results are compared.

5.1. Luminance-based Filter

Results of the luminance-based filter are shown in Fig. 7. Fig. 7(a) and Fig. 7(b) are the preprocessed results of Fig. 6(a) and Fig. 6(b), which have a light reticular background. Fig. 7(c) and Fig. 7(d) are from Fig. 6(f) and Fig. 6(h), which have colorful but light background. All of them have intense foregrounds making it easy to separate them apart. By filtering out the light-colored backgrounds, the salient foreground characters were able to be extracted individually.

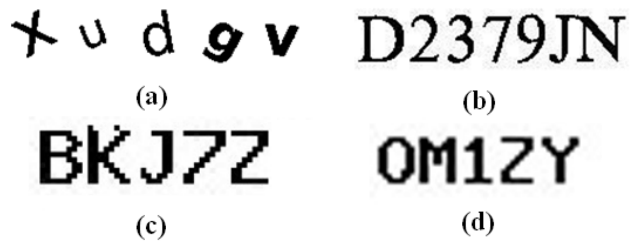


Fig. 7. Sample results of the luminance-based filter. (a-d) are preprocessed results of Fig. 6(a), Fig. 6(b), Fig. 6(f), and Fig. 6(h) correspondingly.

Fig. 8. Preprocessed result of Fig. 6(e) by using the chrominance-based filter.

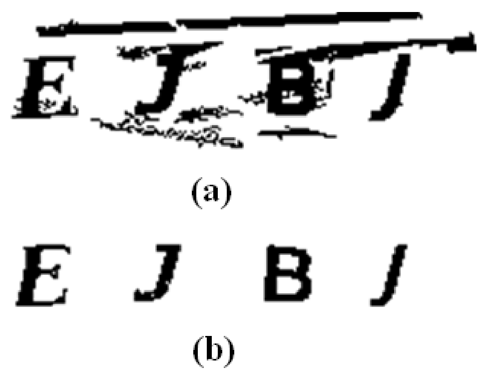
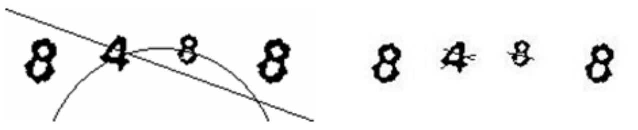


Fig. 9. Preprocessed result of Fig. 6(j). (a)Preprocessed by the luminance-based filter, which leaves noise unsegmented. (b) Proper segmentation done by the chrominance-based filter.

5.2. Chrominance-based Filter

Results of the chrominance-based filter are shown in Fig. 8. and Fig. 9. Fig. 8. is the preprocessed result from Fig. 6(e). There is little luminance difference between characters and the background in Fig. 6(e). By using the luminance-based filter, we got a blank image. However, chrominance of the background and of characters are different. The result of the chrominance-based filter is shown in Fig. 8, where characters are segmented out.

Fig. 9(a) and Fig. 9(b) are from Fig. 6(j). Some characters in Fig. 6(j) have low luminance contrast compared with the background. Fig. 9(a) is the result of Fig. 6(j) processed by the luminance-based filter. The result from luminance includes a great amount of noise, which shouldn't be a problem for us to tell apart. Therefore, by using the chrominance-based filter, the background was labeled as cluttered fragments and viewed as invalid parts. Remaining parts are shown in Fig. 9(b). Complete characters are segmented.



(a) (b)

Fig. 10. Preprocessed results of Fig. 6(c) right before and after the knowledge-based filter. (a) Only passing the luminance-based filter. (b)Applying the knowledge-based filter, crossing lines and curves are filtered out.

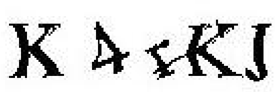
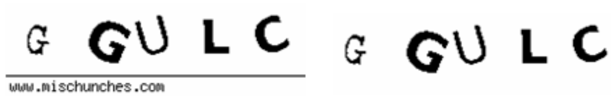


Fig.11. Preprocessed result of Fig. 6(d) by knowledge-based filter.



(a) (b)

Fig.12. Preprocessed result of Fig. 6(g) right before and after the knowledge-based filter. (a) Only passing the luminance-based filter. (b)Applying the knowledge-based filter, small characters and lines are filtered out.

5.3. Knowledge-based Filter

Results of the knowledge-based filter are shown in Fig. 10., Fig.11., and Fig.12. Fig.10. is the result from Fig. 6(c); Fig.11. is the result from Fig. 6(d). Both Fig. 6(c) and Fig. 6(d) have salient lines or curves crossing through characters. For human eyes, it shouldn't be a problem to recognize them as noise. This is because information of lines is scattered and sparse.

For Fig. 6(c), the luminance-based filter can easily filter out the background, leaving out the intense parts, as shown in Fig. 10(a). Fig. 10(b) shows the result after applying the knowledge-based filter, main parts of the line and curve are filtered out. Rough edges remained on characters '4' and '8', but it caused little trouble for recognition.

The knowledge filter can be directly applied to Fig. 6(d). The thin line above 'r' and the one connecting the right 'K' and 'J' were filtered out; the thick line above the left 'K' was filtered out due to its long bar shape. However, the thick line extended from the right 'K' to 'r' cannot be filtered out. This is because before recognizing characters 'r' and 'K', the line between them may be part of some character. It should be viewed as valid information in the first place.

Fig.12. is the preprocessed result from Fig. 6(g). The luminance-based filter or the chrominance-based filter could separate the salient parts from the background, as shown in Fig.12(a). With the help of the knowledge-based filter, the line under the characters and the URL beneath it were filtered out, as shown in Fig.12(b). In Fig.12(a), a thin line extends though the whole frame and the font size of the URL is too small. Both of them were recognized as noise and omitted.

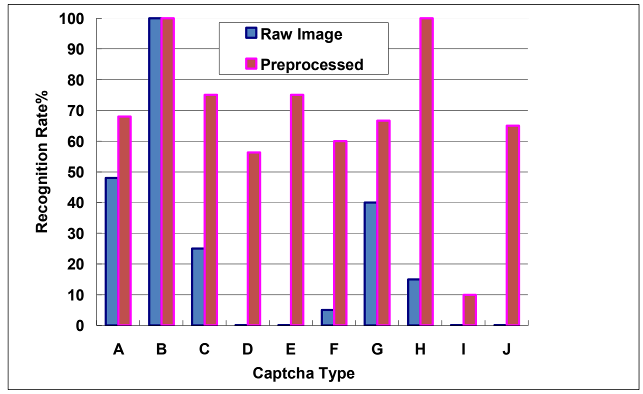


Fig.13. Recognition rate of raw CAPTCHA characters versus preprocessed CAPTCHA characters.

5.4. Numerical results from OCR

We used a common OCR system to test whether our segmentation system can improve the recognition rate. The OCR system we use was "SimpleOCR", which is a free software provided by Simple Software (http://www.simpleocr.com/). The purpose of this system is to recognize machine-printed characters that are scanned from printed documents. Characters of different fonts can be recognized as well as those with slight distortion or rotation. Written or drawn characters have lower recognition rates compared with regular-script characters. Nevertheless, this system is adequate to verify our results. We tested the types of CAPTCHAs mentioned above with a total number of 40 samples.

The experiment was divided into two parts. The first part was recognizing raw images, and the second part was recognizing preprocessed images. Fig.13. shows the average recognition for each kind of CAPTCHA. For most types of CAPTCHAs, the recognition rates for preprocessed images are much higher than those raw ones. In addition, two exceptions are to be noticed. CAPTCHA type(b) can be recognized perfectly without any preprocessing. This is because of its simple background and ordered characters. Characters in CAPTCHA type(i) fail at the identification procedure due to its hollow fonts even after preprocessing. For these two extreme examples, preprocessing shows very small effects on recognition. Therefore, we leave out these two data when calculating the overall recognition rate.

As a whole, the overall recognition rate for raw CAPTCHA images is *16.63%.* This means the OCR system has difficulty recognizing noisy and complicated images. For the preprocessed images, the overall recognition rate increased to *70.74%.* The effect of preprocessing is clear when we compare the two results. With good segmentation, the recognition process turns out to be much easier and more precise.

6. CONCLUSION

In this article, we used a bio-inspired unified model to attain the preprocessing of character recognition for CAPTCHA. Previous works are mostly specific to different cases. To improve such insufficiency, a general model that can do segmentation is presented. We proposed a model that combines three methods. Each of them is a mapping of some function in the human visual system. The system was tested with various types of CAPTCHA characters that have noisy backgrounds and are difficult for traditional character recognition systems. Most CAPTCHA characters were able to be extracted despite their complicated patterns. From our simulation results, character recognition accuracy is enhanced with the help of segmentation before identification. We showed that segmentation is a crucial part in the process of recognition. Also, by using a model that imitates the human visual system, a more general solution for recognition can be found. Our research on characters provides an alternative to improve the more general problem of visual recognition.

7. REFERENCES

[1] S. Mori, C.Y. Suen, and K. Yamamoto, "Historical Review of OCR Research and Development," *Proceedings of the IEEE,* vol. *80,* no. 7, pp. 1029-1057, 1992.

[2] C. Y. Suen, A. M. Berthold, and A. **S.** Mori, "Automatic recognition of handprinted characters-The state of the art," *Proceedings of the IEEE*, vol. 68, pp. 469-187, Apr. 1980.

[3] N. Kato, M. Suzuki, S. Omachi, H. Aso, and Y Nemoto, "A handwritten character recognition system using directionalelement feature and asymmetric Mahalanobis distance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 21, No.3, pp. 258-262, 1999.

[4] Y. Lu, "Machine Printed Character Segmentation-An Overview," *Pattern Recognition,* vol. 28, no. 1, pp. 67-80, 1995.

[5] L. von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart (automatically)," *CMU Tech Report CMUCS-02-117*, 2002.

[6] L.vonAhn,M.Blum,N.Hopper,andJ.Langford, "CAPTCHA: Using hard AI problems for security," *Proc.EUROCRYPT'03*, pages 294–311, 2003.

[7] G. Moriand, J. Malik, "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol.1, pp. 134-141,2003.

[8] G. Moy, N. Jones, C. Harkless, and R. Potter, "Distortion Estimation Techniques in Solving Visual CAPTCHAs," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, Vol. 2, pp. 23-28, 2004.

[9] K. Chellapilla, P. Y. Simard, "Using machine learning to break visual human interaction proofs (HIPs)," *Advances in Neural Information Processing Systems 17*, pp. 265–272, 2005.

[10] A. Duchowski, R. Vertegaal, "Eye-Based Interaction in Graphical Systems: Theory & Practice," *ACM SIGGRAPH 2000 Course Notes*, 2000.

[11] R. Shapley, "Visual Sensitivity and Parallel Retino-cortical Channels," *Annual Review of Psychology*, 1990.

[12] M. Chen, Z. Cheng, and Y. Liu, "A Robust Algorithm of Principal Curve Detection," *Proceedings of the International Conference on Pattern Recognition (ICPR 2004)*, Vol. I, pp. 429–432, 2004.