What's Up CAPTCHA?
A CAPTCHA Based On Image Orientation

Rich Gossweiler
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
rcg@google.com

Maryam Kamvar
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
mkamvar@google.com

Shumeet Baluja
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
shumeet@google.com

ABSTRACT

We present a new CAPTCHA which is based on identifying an image's upright orientation. This task requires analysis of the often complex contents of an image, a task which humans usually perform well and machines generally do not. Given a large repository of images, such as those from a web search result, we use a suite of automated orientation detectors to prune those images that can be automatically set upright easily. We then apply a social feedback mechanism to verify that the remaining images have a human-recognizable upright orientation. The main advantages of our CAPTCHA technique over the traditional text recognition techniques are that it is language-independent, does not require text-entry (*e.g.* for a mobile device), and employs another domain for CAPTCHA generation beyond character obfuscation. This CAPTCHA lends itself to rapid implementation and has an almost limitless supply of images. We conducted extensive experiments to measure the viability of this technique.

Categories and Subject Descriptors

D.4.6 [**Security and Protection**]: Access Control and Authentication

General Terms

Security, Human Factors, Experimentation.

Keywords

CAPTCHA, Spam, Automated Attacks, Image Processing, Orientation Detection, Visual Processing

1. INTRODUCTION

With an increasing number of free services on the internet, we find a pronounced need to protect these services from abuse. Automated programs (often referred to as bots) have been designed to attack a variety of services. For example, attacks are common on free email providers to acquire accounts. Nefarious bots use these accounts to send spam emails, to post spam and advertisements on discussion boards, and to skew results of on-line polls.

To thwart automated attacks, services often ask users to solve a puzzle before being given access to a service. These puzzles, first introduced by von Ahn et al. in 2003[2], were CAPTCHAs:

Completely Automated Public Turing test to tell Computers and Humans Apart. CAPTCHAs are designed to be simple problems that can be quickly solved by humans, but are difficult for computers to solve. Using CAPTCHAs, services can distinguish legitimate users from computer bots while requiring minimal effort by the human user.

We present a novel CAPTCHA which requires users to adjust randomly rotated images to their upright orientation. Previous research has shown that humans can achieve accuracy rates above 90% for rotating high resolution images to their upright orientation, and can achieve a success rate of approximately 84% for thumbnail images [27]. However, rotating images to their upright orientation is a difficult task for computers and can only be done successfully for a subset of images [15][19].

Figure 1 illustrates that some images are:

(A) easy for both computers and people to orient (because the image contains a face, which can be detected and oriented by computers)

(B) easy for humans to orient (because the image contains an object, *e.g.* a bird, that is easily recognized by humans) but difficult for computers to orient (because the image contains multiple objects with few guidelines for meaningful segmentation and the object in the foreground is of an
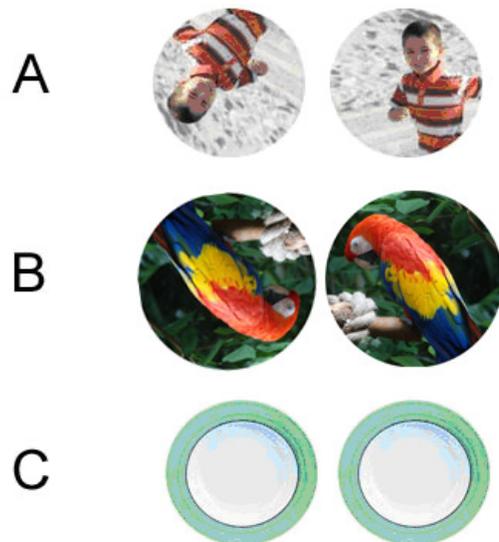

Figure 1: images with various orientation properties (left column: the image randomly rotated, right column: the image in its upright position).

irregular, deformable, shape).

(C) difficult for both people and computers to orient (because the image is ambiguous and there is no "correct" upright orientation)

To obtain candidate images for our CAPTCHA system, we start with a large repository and then remove images that a computer can successfully orient as well as those that are difficult for humans to orient.

For example, all of the images returned from an image-search start as potential candidates for our system. We then use a suite of automated orientation detectors to remove those that can be set upright by a computer. We discuss the system used to automatically determine upright orientation in Section 2. We then apply a social feedback mechanism to verify that the remaining images are easily oriented by humans. In order to identify images that people cannot orient, we compute the variance of users' submitted orientations and reject images which have a high variance. We discuss this social-feedback mechanism in detail in Section 3.

Our CAPTCHA technique achieves high success rates for humans and low success rates for bots, does not require text entry, and is more enjoyable for the user than text-based CAPTCHAs. We discuss two user studies we have performed to demonstrate both the viability and the user-experience of our system in Section 4. In Section 5, we present directions for future study.

1.1 BACKGROUND: CAPTCHAs

Traditional CAPTCHAs require the user to identify a series of letters that may be warped or obscured by distracting backgrounds and other noise in the image. Various amounts of warping and distractions can be used; examples are shown in Figure 2.

Recently, many character recognition CAPTCHAs have been deciphered using automated computer vision techniques. These methods have been custom designed to remove noise and to segment the images to make the characters amenable for optical-character recognition [3][4][5]. Because of the large pragmatic
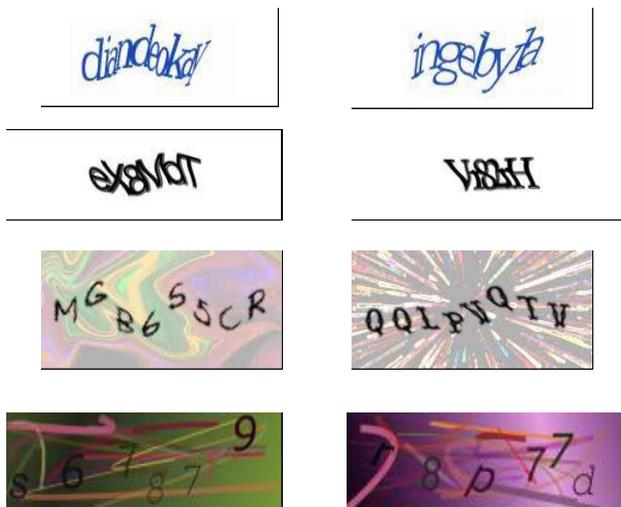

Figure 2: typical character recognition type CAPTCHAs (from Google's Gmail, Yahoo Mail, xdrive.com, forexhound.com)

and economic incentives for spammers to defeat CAPTCHAs, the techniques introduced in academia to defeat CAPTCHAs are soon likely to be in widespread use by spammers. To minimize the success of these automated methods, systems increase the noise and warping used in these CAPTCHAs. Unfortunately, this not only makes it harder for computers to solve, but it also makes it difficult for people to solve – leading to higher error rates [8][9] and higher associated frustration levels.

To address this, numerous alternate CAPTCHAs (including image based ones) have been proposed [1][6][7][8]. In designing a new CAPTCHA, the basic tenets for creating a CAPTCHA (from [10]) should be kept in mind:

1. Easy for most people to solve
2. Difficult for automated bots to solve
3. Easy to generate and evaluate

It is straightforward to create a system that fulfills the first two requirements. The first requirement suggests the need for usability evaluations, ensuring that people can solve the CAPTCHA in a reasonable amount of time and with reasonable success rates. The second requirement suggests that we test state-of-the-art automated methods against the CAPTCHA. In the CAPTCHA proposed here, we ensure the automated methods can not be used to defeat our CAPTCHA by using them to filter images which can be automatically recognized and oriented.

The third requirement is harder to fulfill; it is this requirement that presents the greatest challenge to image-based CAPTCHA systems. The early success of the text-CAPTCHAs was aided by the ease in which they could be generated – random sequences of letters could be chosen, distorted, and distracting pixels, noise, colors, etc. added. Subsequent image-based CAPTCHAs were proposed which required users to identify images with labels. The difficulty with these systems is that they require *a priori* knowledge of the image labels. Reliable labels are not available for most images on the web, so common techniques used to obtain labels included:

(1) using the label assigned to an image by a search engine,

(2) using the context of the page to determine a label,

(3) using images that were labeled when they were encountered in a different task, or

(4) using games to extract the labels from users (such as the ESP game [11]).

Unfortunately, many times the labels obtained by the former two methods are often noisy and unreliable in practice because people are needed to manually verify the labels. The latter two approaches provide less noisy labels. However, even in the cases in which labels can be obtained, it is necessary to be careful how they are used. Asking the user to come up with the label may be difficult unless many labels are assigned to each image. Furthermore, unless exact matches are entered, similarity distances between given and expected answers may be quite complex to compute (for example, a number of measurements can be used: edit distance, ad-hoc semantic distance, thesaurus distance, word-net distance, etc.).

Other, more interesting uses of labeled images, such as finding sets of images with recurring themes (or images that do not belong

to the same set) are possible [10]. However, it is likely that when a small set of N images is given, and the goal is to find which of the N-1 images does not pertain to the same set (i.e. the anomalous CAPTCHA, as described in [10]), automated methods may be able to make significant inroads. For example, if N-1 images are of a chair in several different orientations and the anomalous image is of a tree, the use of current computer-vision techniques will be able to narrow down the candidates rapidly (e.g. using local-feature detection [20] and the many variants [21]).

In the CAPTCHA we propose, we are careful not to provide the user with a small set of images to compare. Any similarity computation must be done against the entire set of images possible – without any *a priori* filtering clues given. The success of our CAPTCHA rests on the fact that orienting an image is an AI-hard problem. In the next section, we will review the many systems that attempt to determine an image's upright orientation. Although a few systems achieve success, their success is, when tested in realistic scenarios, limited to a small subset of image types [19].

2. DETECTING ORIENTATION

The interest in automated orientation detection rapidly arose with the advent of digital cameras and camera phones that did not have built-in physical orientation sensors. When images were taken, software systems needed a method to determine whether the image was portrait (upright) or landscape (horizontal). The problem is still relevant because of the large scale scanning and digitization of printed material.

The seemingly simple task of making an image upright is quite difficult to automate over a wide variety of photographic content. There are several classes of images which can be successfully oriented by computers. Some objects, such as faces, cars, pedestrians, sky, grass etc. [22][23], are easily recognizable by computers. It is important to note that computer-vision techniques have not yet been successful at unconstrained object detection; therefore, it is infeasible to recognize the vast majority of objects in typical images and use the knowledge of the object's shape to orient the image.

Instead of relying on object recognition, the majority of the techniques explored for upright detection do not attempt to understand the contents of the image. Rather, they rely on an assortment of high-level statistics about regions of the image (such as edges, colors, color gradients, textures), combined with a statistical or machine learning approach, to categorize the image orientation [12]. For example, many typical vacation images (such as sunsets, beaches, etc.) have an easily recognizable pattern of light and dark or consistent color patches that can be exploited to yield good results.

Many images, however, are difficult for computers to orient. For example, indoor scenes have variations in lighting sources, and abstract and close-up images provide the greatest challenge to both computers and people, often because no clear anchor points or lighting sources exist.

The classes of images that are easily oriented by computers are explicitly handled in our system. A detailed examination of a recent machine learning approach in [19] is given below. It is incorporated in our system to ensure that the chosen images are difficult for computers to solve.

2.1 LEARNING IMAGE ORIENTATION

In order to identify images that are easy for computers to orient, we pass the images through an automated orientation detection system, developed by Baluja [19]. Although the particular machine learning tools and features used make this orientation-detection system distinct, the overall architecture is typical of many current systems.

When the orientation detection system receives an image, it computes a number of simple transformations on the image, yielding 15 single-channel images:

- 1-3: Red, Green, Blue (R,G,B) Channels.
- 4-6: Y, I, Q (transformation of R,G,B) Channels.
- 7-9: Normalized version of R,G,B (linearly scaled to span 0-255).
- 10-12: Normalized versions of Y,I,Q (linearly scaled to span 0-255).
- 13: Intensity (simple average of R, G, B).
- 14: Horizontal edge image computed from intensity.
- 15: Vertical edge image computed from intensity.

For each of these single-band images, the system computes the mean and variance of the entire image as well as for square sub-regions of the image. The sub-regions cover (1/2)x(1/2) to (1/6)x(1/6) of the image (there are a total of 91=1+4+9+16+25+36 squares). The mean and variance of vertical and horizontal slices of the image that cover 1/2 to 1/6 of the image (there are a total of 20=2+3+4+5+6 vertical and 20 horizontal slices) are also computed. In sum, there are 1965 features representing averages (15*(91+20+20)) and 1965 features representing variances, for a total of 3930 features. These

Rotated Input Images

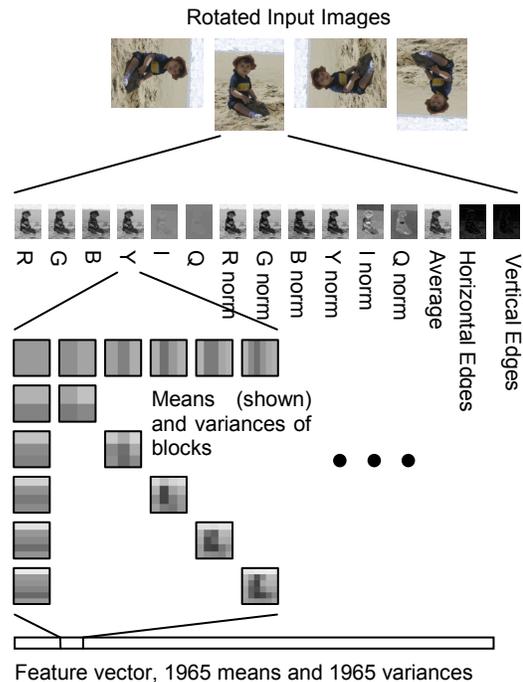Feature vector, 1965 means and 1965 variances

Figure 3: Features extracted from an image.

'retinal', or localized, features have been successfully employed for detection tasks in a variety of visual domains. Figure 3 shows the features in detail.

The image is then rotated by a set amount, and the process repeats. Each time, the feature vector is passed through a classifier (in this case a machine-learning based AdaBoost [25] classifier that is trained to give a +1 response if the image is upright and a -1 response otherwise). The classifier was previously trained using thousands of images for which the upright orientation was known (these were labeled with a +1), and were then rotated by random amounts (these rotations were labeled with -1). Although a description of AdaBoost and its training is beyond the scope of this paper, the classifiers found by AdaBoost are both simple to compute (are orders of magnitude faster than the somewhat worse-performing Support Vector Machine based classifiers for this task) and are memory efficient; both are important considerations for deployment.

As Figure 4 illustrates, when an image is given for classification, it is rotated to numerous orientations, depending on the accuracy needed, and features are extracted from the image at each orientation. Each set of these features is then passed through a classifier. The classifier is trained to output a real value between +1.0 for upright and -1.0 for not upright. The rotation with the maximal output (closest to +1.0) is chosen as the correct one. Figure 4 shows four orientations; however, any number can be used.

When tried on a variety of images to determine the correct upright orientation from only the four canonical 90° rotations, the system yielded wildly varying accuracies ranging from approximately 90% to random at 25%, *depending on the content of the image*. The average performance on outdoor photographs, architecture photographs and typical tourist type photographs was significantly higher than the performance on abstract photographs, close-ups and backgrounds. When an analysis of the features used to make the discriminations was done, it was found that the edge features play a significant role. This is important since they are not reliant on color information – so black and white images can be captured; albeit with less accuracy.

For our use, we use multiples of the classifiers described above. 180 Adaboost classifiers were trained to examine each image and determine the susceptibility of that image to automated attacks
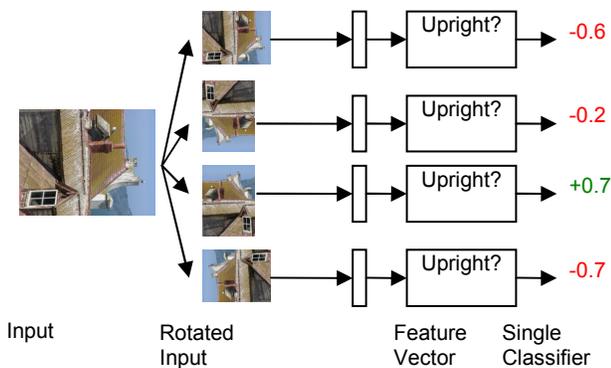

Figure 4: Overview of the system in its simplest form.

when used as a CAPTCHA. The details of the image selection process and how the Adaboost classifiers are used are given in the next section.

3. SELECTING IMAGES FOR THE ROTATIONAL CAPTCHA SYSTEM

As previously mentioned, a two-step process is needed to determine which images should be included in our CAPTCHA system. First, in Section 3.1, we describe the automated methods used to detect whether a candidate image should be excluded because it is easily oriented by a computer. In Section 3.2, we describe the social-feedback mechanism that can harness the power of users to further identify which images should be excluded from the dataset because they are too difficult for humans to orient.

3.1 Removing Computer-Detectable Images

It is important not to simply select random images for this task. There are many cues which can quickly reveal the upright orientation of an image to automated systems; these images must be filtered out. For example, if typical vacation or snapshot photos are used, automated rotation accuracies can be in the 90% range [14][15][19]. The existence of any of the cues in the presented images will severely limit the effectiveness of the approach. Three common cues are listed below:

1. Text: Usually the predominant orientation of text in an image reveals the upright orientation of an image.

2. Faces and People: Most photographs are taken with the face(s) / people upright in the image.

3. Blue skies, green grass, and beige sand: These are all revealing clues, and are present in many travel/tourist photographs found on the web. Extending this beyond color, in general, the sky often has few texture/edges in comparison to the ground. Additional cues found important in human tests include "grass", "trees", "cars", "water" and "clouds" [27][16].

Ideally, we would like to use only images that do not contain any of the elements listed above. All of the images chosen for presentation to a user were scanned automatically for faces and for the existence of large blocks of text. If either existed, the image was no longer a candidate.[1] Although accurate detectors do not exist for all the objects of interest listed in (3) above, the types of images containing the other objects (trees, cars, clouds) were often outdoors and were effectively eliminated through the use of the automated orientation classifiers described in Section 2.1.

If the image had neither text nor faces, it was passed through the set of 180 AdaBoost classifiers in order to further ensure that the candidate image was not too easy for automated systems. The output of these classifiers determined if the image was accepted into the final image pool. The following heuristics were used when analyzing the 180 outputs of the classifiers:

- If the majority of the classifiers oriented the image similarly with a high confidence score, it was rejected. The image was too easy.

---

[1] Additionally, all images were passed through an automated adult-content filter [24]. Any image with even marginal adult-content scores was discarded.

- If the predictions of the classifiers together had too large entropy, then the image was rejected. Because the classifiers are trained independently, they make different guesses on ambiguous images. Some images (such as simple textures, macro images, etc.) have no discernible upright orientation for humans or computers. Therefore, if the entropy of guesses was high, the image may not actually have a discernible correct orientation.

These two heuristics attempt to find images that were not too easy, but yet possible to orient correctly. The goal is to be conservative on both ends of the spectrum; the images need to neither be too easy nor too hard. The images were accepted when no single orientation dominated the results, while ensuring that there were still peaks in a histogram of the orientations returned.

There are many methods to make the selection even more amenable to people while remaining difficult for computers. It has been found in [15] that the correct orientation of images of indoor objects is more difficult than outdoor objects. This may be due to the larger variance of lighting directionality and larger amounts of texture throughout the image. Therefore, using a classifier to first select only indoor images may be useful. Second, due to sometimes warped objects, lack of shading and lighting cues, and often unrealistic colors, cartoons also make ideal candidates. Automated classifiers to determine whether an image is a cartoon also exist [26] and may be useful here to scan the web for such images. Finally, although we did not alter the content of the image, it may be possible to simply alter the color-mapping, overall lighting curves, and hue/saturation levels to reveal images that appear unnatural but remain recognizable to people.

3.2 Removing Images Difficult for Humans to Orient

Once we have pruned from our data set images that a computer can successfully orient, we identify images that are too difficult for a human to successfully rotate upright. To do this, we present several randomly rotated images to the user in the deployed system. One of the images presented is a "new" candidate image being considered to join the pool of valid images. As large numbers of users rotate the new image we examine the average and standard deviation of the human orientations.

We identify images that are difficult to rotate upright by analyzing the angle which multiple users submitted as upright for a given image. Images that have a high variation in their submitted orientations are those that are likely to have no clear upright orientation. Based on this simple analysis from users, we can identify and exclude difficult images from our dataset.

This social feedback mechanism also has the added advantage of being able to "correct" images whose default orientation is not originally upright – for example images where the photographer may not have held the camera exactly upright. Though the variance of the submitted orientation across users may be small, the average orientation may be different than the image's posted orientation. Users will correct this image to its natural upright position, compensating for the angle of the original image.

This social mechanism allows us to consistently correct or reject the images used in the CAPTCHA, which when combined with the automated techniques to exclude machine-recognizable images, produces a dataset for our rotational CAPTCHA system.

4. USER EXPERIMENTS

In this section, we describe two user studies. The first study was designed to determine whether this system would result in a viable CAPTCHA system in terms of user-success rates and bot-failure rates. The second study was designed to informally gauge user reactions to the system in comparison to existing CAPTCHAs. Since these were uncontrolled studies, we did not measure task-completion times.

4.1 Viability Study

The goal of this study was to understand if users would determine the same upright orientation for candidate images in the rotational CAPTCHA system. We found that after applying a social-correction heuristic (which can be applied in real time in a deployed system), our CAPTCHA system meets high human-success and high computer-failure standards.

4.1.1 Image Dataset

The set of images used for our rotational CAPTCHA experiment was collected from the top 1,000 search results for popular image-queries[2]. We rejected from the dataset any image which could be machine-recognizable, according to the process described in Section 3. From the remaining candidate images, we selected a set of approximately 500 images to be the final dataset which we used in our study. This ensures that our dataset meets the two requirements laid forth by [2]:

- First, that this CAPTCHA does not base its security in the secrecy of a database. The set of images used is the set of images on the WWW, and is thus is non-secretive. Further, it is possible to alter the images to produce ones that can be made arbitrarily more difficult.

- Second, that there is an automated way to generate problem instances, along with their solution. We generate the problem instances by issuing an image search query; their solution (the image's orientation) defaults to the posted orientation of the image on the web, but may be changed to incorporate the corrective offset found by the social-feedback mechanism.

To normalize the shape and size of the images, we scaled each image to a 180x180 pixel square and we then applied a circular mask to remove the image corners.

4.1.2 Experiment Setup

500 users were recruited through Google-internal company email groups used for miscellaneous communications. The users came from a wide cross-section of the company, and included engineers, sales associates, administrative assistants and product managers. Users participated in the study from their own computer and were not compensated for their participation. Since this study was done remotely at the participant's computer there was no human moderator present. Participants received an email

---

[2] Image queries are those which return a list of images, rather than a list of website URLs. For example, any query issued on images.google.com would be considered an image query.

with a link to the experiment website which included a brief introduction to the study:

> "This experiment will present a series of images one at a time. Each image will be rotated to a random angle. Use the provided slider to rotate the image until you believe it is in its natural, upright position, then press *submit* to go to the next image. This process will continue until you have adjusted ten images."

Figure 5 shows a screenshot of an example trial in the viability study written using cross-browser JavaScript and DHTML.
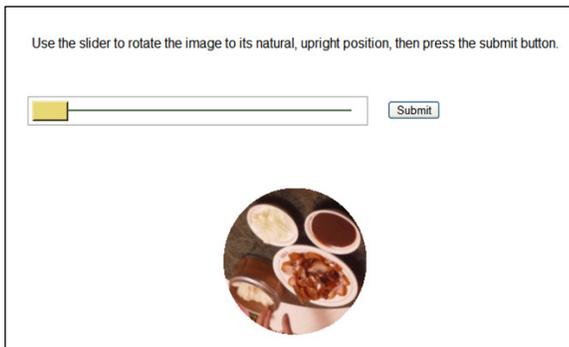

Figure 5: Screenshot of an example trial in the viability study

Each user was asked to rotate 10 images to their natural upright position. The first six images and their offset angles were the same for each user. We kept these trials constant to ensure that we would have a significant sample size for some of the problem instances. Figure 6 shows the first six images at the orientation that they were shown to the users. The last four images and their offset angles were randomly selected at runtime. We did this to evaluate our technique on a wide variety of images. For each trial, we recorded the image-ID, the image's offset angle (a number between ±180 which indicated the position the image was presented to the user), and the user's final rotation angle (a number between ±180 which indicated the angle at which the user submitted the image).

4.1.3 Results

We have created a system that has sufficiently high human-success rates and sufficiently low computer-success rates. When using three images, the rotational CAPTCHA system results in an 84% human success metric, and a .009% bot-success metric (assuming random guessing). These metrics are based on two variables: the number of images we require a user to rotate and the size of the acceptable error window (the degrees from upright which we still consider to be upright). Predictably, as the number of images shown becomes greater, the probability of correctly solving them decreases. However, as the error window increases, the probability of correctly solving them increases. The system which results in an 84% human success rate and .009% bot success rate asks the user to rotate three images, each within 16° of upright (8-degrees on either side of upright).
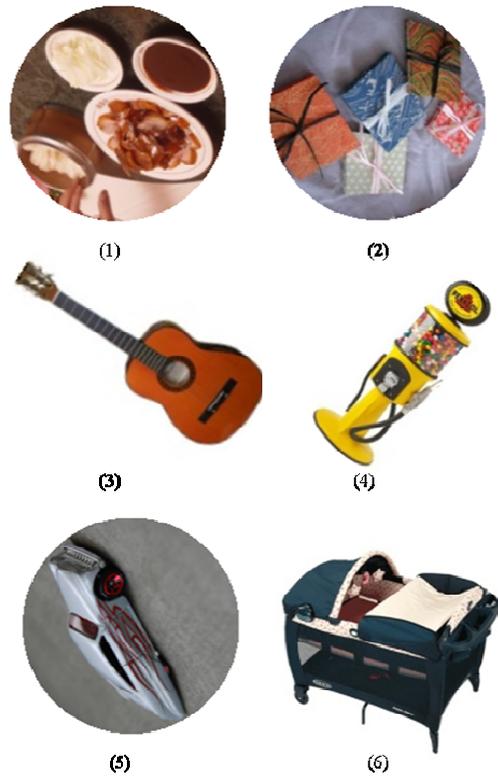

Figure 6: The first six images displayed to users

Figure 7 illustrates the average and standard deviation of users' final rotation angles for the first six images (the images which were shown to all of the users). There are some images for which users rotate very accurately (images 1, 5 and 6), and those which users do not seem to rotate accurately (images 2, 3 and 4). The images which have poor results can be attributed to by three factors, each of which can be addressed by our social feedback mechanism:
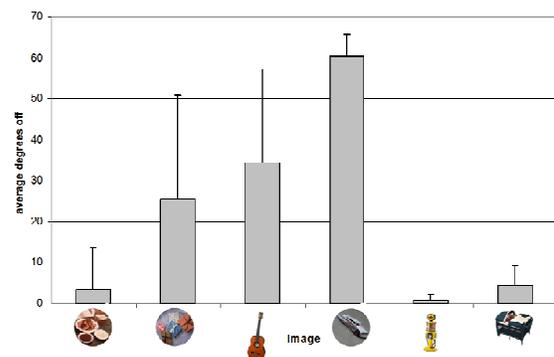

Figure 7: average degrees from the original orientation that each image was rotated.

1. Some images are simply difficult to determine which way is upright. Figure 8 shows one such image and plots the absolute number of degrees-from-upright which each user rotated the image. Based on the standard deviation in responses, this image is not a good candidate for social correction. We see that its

standard deviation was greater than the half of the error window; it was deemed not to have an identifiable upright position, and was rejected from the dataset.
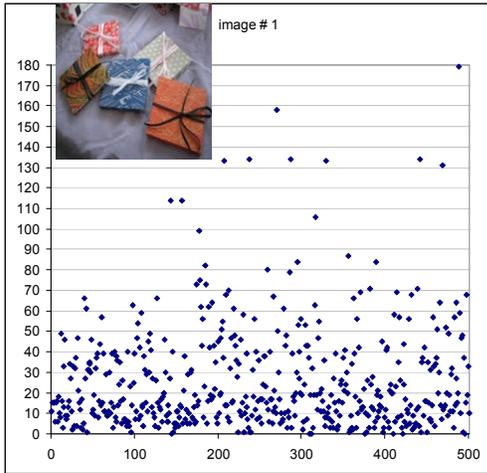

Figure 8: An image with large distribution of orientations.

2. Some images' default upright orientation may not correspond to the users' view of their natural upright orientation. We designate the default upright orientation as the angle at which the image was taken originally. This is illustrated in the picture of the toy car (image #3). Figure 9 shows the original orientation of the image, in contrast to the orientation of the image which most users thought was "natural", shown in the graph. Based on the low deviation in responses, this image is a good candidate for being "socially corrected". If this image was used after the social correction phase, the "upright" orientation would be changed to approximately 60° from the shown orientation.
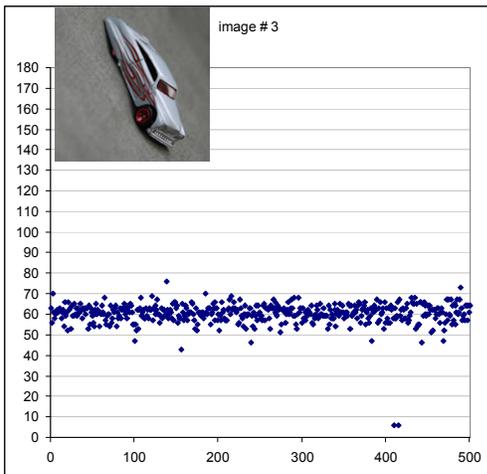

Figure 9: An image requiring social correction.

3. Some images have multiple "natural" upright positions. Figure 10 shows various orientations of the guitar image which could be considered upright. In our analysis we rejected any image which had multiple upright orientations (indicated by a large standard deviation in image rotation results). However, future versions of this CAPTCHA system may choose to allow for multiple orientations, if there is a multi-modal clustering around a small number of orientations.

Figure 10: Two possible natural orientations of the image.

It is important to note that the decisions about whether an image falls into one of the above categories can be made in real time by a system that presents a user a "candidate" image in addition to the CAPTCHA images. The "candidate" image need not be used to influence the user's success at solving the CAPTCHA, but is simply used to gather information. The user is not informed of which image is a candidate image.

In our analysis, the human success rate is determined by the average probability that a user can rotate an image correctly. However, we exclude any images which fall into case 1 or case 3 outlined above. Those images would be identified and subsequently rejected from the dataset by our social-feedback mechanism. If an image falls in case 2, we corrected the upright orientation based on the mode of the users' final rotation, as this could be similarly determined by the correction aspect of our social-feedback mechanism.

Human success rates are influenced by two factors: the size of the error window and the number of images needed to rotate. Table 1 shows the effect on human success, as the size of the error window and number of images we require a user to successfully orient vary. The configurations which have a success rate of greater than 80% are highlighted in green.

Table 1: Human-success rates (%), as number of images shown and size of acceptable error window varies.

| | | size of acceptable error window | | | | | |
		8 degrees	8 degrees	10 degrees	12 degrees	14 degrees	16 degrees
number of images	1	68.10	75.20	91.50	93.10	93.50	94.40
	2	43.89	58.55	83.72	86.68	87.42	89.11
	3	28.88	42.53	76.61	80.70	81.74	84.12
	4	19.09	31.98	70.09	75.13	76.43	79.41
	5	12.62	24.05	64.14	69.94	71.46	74.97
	6	8.34	18.08	58.68	65.12	66.81	70.77

The viability of a CAPTCHA is not only dependent on how easy it is to solve by humans, but it is also dependent on how difficult it is to solve by computers. Computer success rate is the probability that a machine can solve the CAPTCHA. No algorithm has yet been developed to successfully rotate the set of images used in our CAPTCHA system. A first pass at estimating

a computer's solution would be a random guess. Since our images have 360 degrees of freedom for rotation, computers would have a 1/360 chance at guessing the exact upright orientation (to within 1°). The computer success rate of our CAPTCHA is based on two factors: the window of error we would allow people to make when rotating the image upright, and the number of images that they would need to rotate. For example, if we allowed users to rotate images in a 6-degree window (3° on either side of upright) the machine success rate would be 6/360. If users were required to rotate 3 images to their upright position, the computer success rate would be decreased to $(6/360)^3$. A CAPTCHA system which displayed ≥ 3 images with a ≤ 16-degree error window would achieve a guess success rate of less than 1 in 10,000, a standard acceptable computer success rates for CAPTCHAs [8]. It should be noted, however, that these estimates are far too optimistic – intelligent orientation detection systems will be able to assign probabilities of upright orientations; thereby making more intelligent, although still perhaps imperfect, guesses. The caveats for intelligent guessing are also equally applicable to text-based CAPTCHA systems; for each character to be recognized, large numbers of incorrect characters can be reliably withdrawn from consideration. Therefore, in a manner similar to increasing the difficulty of text-based CAPTCHAs, we can simply modify/perturb/degrade the image, or increase the number of images to de-rotate.

In its simplest instantiation, we used an image-based CAPTCHA system that requires a user to rotate at least three images upright with a 16 degree error window (8-degrees on either side of upright). In order to generate data for the social-correction system, an additional image (or multiple images), the "candidate images", can be shown with the required CAPTCHA images.
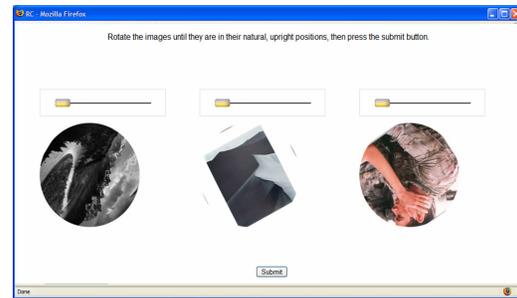
4.2 Happiness Study
The goal of this study was to informally determine what type of CAPTCHA users preferred to use.

4.2.1 Experiment Setup
Sixteen users were recruited to participate in the study through an email to an internal Google company email group and were compensated for their time[3]. The users were selected from a cross-section of departments within Google: the users consisted of 10 sales representatives and six employees from other departments including Engineering, Human Resources, Operations, and Finance. All users participated in the study from a Firefox Browser on a desktop computer located in a usability lab on the Google Campus.

This study asked users to do two tasks: to rotate a set of images into their natural, upright positions (Figure 11A) and to type distorted text into a textbox (Figure 11B). Each task, "rotate image" and "decipher text", had five trials, and the order of these tasks was counterbalanced across users.
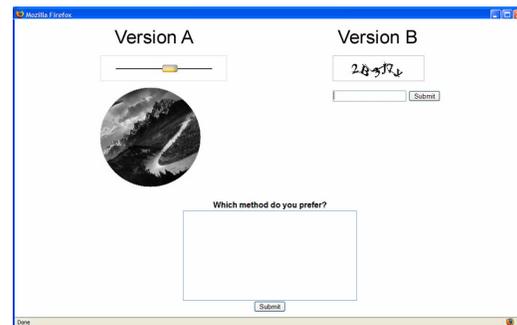
Before the "rotate image" task (Figure 11A), users were told we were measuring their ability to accurately rotate an image to its natural, upright position. Each trial consisted of three images shown to a user on the page. We chose three images because our previous study indicated that at least three images would be

---

[3] Users were compensated with their choice of a $15 Amazon.com gift certificate, a $15 iTunes gift certificate, or a 30 minute massage coupon for participating in the study.

A.

B.

C.

Figure 11: Snapshots of the user-happiness experiment.
11a) One trial in the "rotate image" task.
11b) One trial in the "decipher text" task.
11c) The side-by-side comparison presented to users after they had completed the "rotate image" and "decipher text" tasks.

needed for an effective system. The 15 images that users rotated were randomly selected from the web and processed as described earlier, and we generated a random angle to offset each image. These 15 images were always presented in the same order to users. Users were instructed to press the submit button after adjusting the images in each trial to their upright orientation.

Before the "decipher text" task (Figure 11B), users were told we were measuring their ability to accurately read distorted text. Each trial consisted of users deciphering text consisting of six letters. We randomly chose five CAPTCHAs from Yahoo's CAPTCHA base and presented them to users in the same order. Users were instructed to enter the text they believed to be in the image and press the submit button.

After users completed these 10 tasks, they were presented with a side-by-side comparison of the "rotate image" and "decipher text" (Figure 11 C).

4.2.2 Results

68.75% of users (11 users) preferred rotating images, and 31.25% of users (5 users) preferred deciphering text.

Among the comments from users who preferred the rotational approach indicated they thought that method was "easy", "cool", "fun" and "faster". One user stated that he preferred "visual cues over text", and many users referenced feeling like they were at an eye exam while deciphering the text.

Only two of the five users who stated their preference as "deciphering text" provided insight to their choice. One user pointed to an implementation flaw (that the slider should retain focus even when the mouse left its bounding box) as the reason he did not like the rotational approach, while the other user pointed to familiarity with the text CAPTCHA, and more absolute input mechanism as the rationale for her preference. "I prefer [deciphering text] since it requires simple keyboard inputs which are absolute. With rotating pictures I found myself continually making fine adjustments to make them perfectly upright, therefore taking a slight bit longer to accomplish. Also, I'm much more familiar with [deciphering text] since it's what most internet portals use for security purposes."

From these two studies, we conclude that not only is the rotational task a viable one, but compared to the standard deciphering text, users may prefer it.

5. VARIATIONS AND FUTURE WORK

There are a number of interesting extensions to this CAPTCHA system that we can investigate and deploy. The fundamental system presents n images, of which m of the images are new candidates ($m < n$). In alternative implementations, we can also present n images and ask the user to select p of them to rotate. We suspect that there will be useful trends from this system; difficult images will be chosen less frequently than other images. This gives us further evidence to identify images to exclude from our CAPTCHA system.

There are numerous sources for candidate objects to rotate. Beyond images, we can also introduce views from 3D models (or with advanced graphics capabilities, users can interact with the 3D models themselves). These models, being more austere, can remove many of the features such as lighting and horizons on which automated orientation mechanisms rely. Styles can be applied to remove strong edges. For example, we could use the Golden Gate bridge model without lighting effects and without the sky/ocean horizon. It also gives us another dimension of rotation, greatly increasing the number of possible answers, making it even harder for computers to randomly guess the correct angle. Furthermore, the difficulty in these tasks can be parameterized.

In our experiments, users moved a slider to rotate the image to its upright position. On small display devices such as a mobile phone, they could directly manipulate the image using a touch screen, as seen in Figure 12, or can rotate it via button presses. This may be particularly useful in cases in which there is no character keyboard or where keyboard entry is error prone. User interfaces that cycle through, scale, or otherwise engage the user based on the constraints of the display and input capabilities can be developed, measured, and compared for utility.

Figure 12: Example rotating an image on a mobile device.

Finally, another interesting aspect to this system is related to adoption and user perception. Most CAPTCHAs are viewed as intrusive and annoying. To alleviate user dissatisfaction with them, we can use images that keep the user within the overall experience of the website. For example, on a Disney sign-up page, Disney characters, movie stills, or cartoon sketches can be used as the images to rotate; eBay could use images of objects that are for sale; a Baseball Fantasy Group site could use baseball-related items when creating a user account.

6. CONCLUSIONS

We have presented a novel CAPTCHA system that requires users to adjust randomly rotated images to their upright orientation. This is a task that will be familiar to many people given the use of early digital cameras, cell phones with cameras, and even the simple act of sorting through physical photographs. We have preliminary evidence that shows users prefer rotating images to deciphering text as is required in traditional text based CAPTCHAs. Our system further improves traditional text-based CAPTCHAs in that it is language and written-script independent, and supports keyboard-difficult environments.

It is important that random images are not chosen for this task; they *must* be carefully selected. Many typical vacation and snapshots contain cues revealing upright orientation. We ensure that our CAPTCHA can not be defeated by state-of-the-art orientation detection systems by using those systems to filter images that can be automatically recognized and oriented. In contrast to traditional text based CAPTCHAs which introduce more noise and distortion as automated character recognition improves, we currently do not need to alter or distort the content of the images. As advances are made in orientation detection system, these advances will be incorporated in our filters so that those images that can be automatically oriented are not presented to the user. The use of distortions may eventually be required.

Some of the major pitfalls associated with other proposed image-based CAPTCHA systems do not apply to our CAPTCHA system. *A priori* knowledge of the image's label is not needed, which makes examples for our system easier to automatically generate than other image-based CAPTCHA systems. Furthermore, it is harder for bots to solve than the image-based CAPTCHAs that require a user to identify a common theme across a set of images, since the set of images to compare against is not closed.

Finally, our system provides opportunities for a number of interesting extensions. First, the set of images selected can be chosen to be more interesting or valuable to the end-user by displaying those that are related to the overall theme of the website. Second, more aggressive social-correction can be used through the presentation of multiple images of which only a few must be uprighted; this gives real, and immediate, insight into which images may be too hard for users. Third, the large number of 3D models being created for independent applications, such as Google's Sketch-Up, can be used as sources of new images as well as full-object rotations.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] Shahreza, A., Shahreza, S. (2008) "Advanced Collage CAPTCHA", Fifth International Conference on Information Technology, 1234- 1235

[2] von Ahn, L., Blum, M., Hopper, N. and Langford, J. CAPTCHA: Using Hard AI Problems for Security. Advances in Cryptology, Eurocrypt 2003. Pages 294-311.

[3] Huang, S.Y., Lee, Y.K., Bell, G. Ou, Z.h. (2008) "A Projection-based Segmentation Algorithm for Breaking MSN and YAHOO CAPTCHAs", The 2008 International Conference of Signal and Image Engineering

[4] Chellapilla K., Simard, P. "Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)," in L. K. Saul, Y. Weiss, and L. Bottou, editors, Advances in Neural Information Processing Systems 17, pp. 265–272. MIT Press

[5] Mori, G., Malik, J. (2003) "Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA", in Computer Vision and Pattern Recognition (CVPR-2003).

[6] Elson, J., Douceur, J. Howell, J., Saul, J., (2007) Asirra: A CAPTCHA that Exploits Interest-Aligned Manual Image Categorization, in Proceedings of the 14th ACM conference on Computer and communications security.

[7] Golle, P. (2008) Machine Learning Attacks against the Asirra CAPTCHA, to appear in in Proceedings of the 15th ACM conference on Computer and communications security.

[8] Chellapilla, K., Larson, K., Simard, P., Czerwinski, M., Designing Human Friendly Human Interaction Proofs (HIPs), CHI-2005.

[9] Yan, J., Ahmad, A.S.E., (2008) Usability of CAPTCHAs Or Usability issue in CAPTCHA design. In Symposium on Usable Privacy and Security (SOUPS) 2008.

[10] Chew, M., Tygar, D. (2004) Image Recognition CAPTCHAs, in Proceedings of the 7th International Information Security Conference (ISC 2004)

[11] Ahn, L.V., Dabbish, L. (2004) Labeling Images with a Computer Game, CHI-2004.

[12] Vailaya, A., Zhang, H., Yang, C., Liu, F., Jain, A. (2002) "Automatic Image Orientation Detection", IEEE Transactions on Image Processing. 11,7.

[13] Wang, Y. & Zhang, H. (2001), "Content-Based Image Orientation Detection with Support Vector Machines" in IEEE Workshop on Content-Based Access of Image and Video Libraries. pp 17-23.

[14] Wang, Y., & Zhang, H. (2004) "Detecting Image Orientation based on low level visual content" Computer Vision and Image Understanding, 2004.

[15] Zhang, L, Li, M., Zhang, H (2002) "Boosting Image Orientation Detection with Indoor vs. Outdoor Classification", Workshop on Application of Computer Vision, 2002.

[16] Luo, J. & Boutell, M. (2005) A probabilistic approach to image orientation detection via confidence-based integration of low level and semantic cues, IEEE Transactions on Pattern Analysis and Machine Intelligence, v27,5 pp.715-726.

[17] Lyu, S. (2005) Automatic Image Orientation Determination with Natural Image Statistics, Proceedings of the 13th annual ACM international conference on Multimedia, pp 491-494

[18] Wang, L., Liu, X., Xia, L, Xu, G., Bruckstein, A., (2003) "Image Orientation Detection with Integrated Human Perception Cues (or which way is up)", ICIP-2003.

[19] Baluja, S. (2007) Automated image-orientation detection: a scalable boosting approach, Pattern Analysis & Applications, V10, 3.

[20] Lowe, D.G., Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, v.60 n.2, p.91-110, November 2004

[21] Tuytelaars, Tinne, Mikolajczyk, K., A Survey on Local Invariant Features, preprint, Foundations and Trends in Computer Graphics and Vision 1:1, 1-106.

[22] Yang, M.H., Kriegman, D.J., Ahuja, N. (2002) "Detecting Faces in Images", IEEE-PAMI 24:1

[23] Bileschi, S., Leung, B. Rifkin, R., Towards Component-based Car Detection, 2004 ECCV Workshop on Statistical Learning and Computer Vision.

[24] Rowley, H., Jing, Y., Baluja, S. (2006), Large-Scale Image-Based Adult-Content Filtering, International Conference on Computer Vision Theory and Applications.

[25] Freund, Y., R. Schapire, "Experiments with a New Boosting Algorithm" (1996), in Machine Learning, Proceedings of the Thirteenth International Conference – 1996.

[26] Lienhart, R., Hartmann, A. (2002) Classifying images on the web automatically, J. Electron. Imaging Vol 11, 445

[27] Luo, J., Crandall, D., Singhal, A., Boutell, M. Gray,R., "Psychophysical Study of Image Orientation Perception", Spatial Vision, V16:5.