Multimedia Systems 9: 493–502 (2004)
Digital Object Identifier (DOI) 10.1007/s00530-003-0122-3

Multimedia Systems

ARTiFACIAL: Automated Reverse Turing test using FACIAL features

Yong Rui, Zicheng Liu

Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
e-mail: {yongrui,zliu}@microsoft.com

Abstract. Web services designed for human users are being abused by computer programs (bots). The bots steal thousands of free e-mail accounts in a minute, participate in online polls to skew results, and irritate people by joining online chat rooms. These real-world issues have recently generated a new research area called human interactive proofs (HIP), whose goal is to defend services from malicious attacks by differentiating bots from human users. In this paper, we make two major contributions to HIP. First, based on both theoretical and practical considerations, we propose a set of HIP design guidelines that ensure a HIP system to be secure and usable. Second, we propose a new HIP algorithm based on detecting human face and facial features. Human faces are the most familiar object to humans, rendering it possibly the best candidate for HIP. We conducted user studies and showed the ease of use of our system to human users. We designed attacks using the best existing face detectors and demonstrated the challenge they presented to bots.

Keywords: Human interactive proof (HIP) – Web services security – CAPTCHA – Face detection – Facial feature detection

1 Introduction

Web services are increasingly becoming part of people's everyday life. For example, we use free e-mail accounts to send and receive e-mails, online polls to gather people's opinion, and chat rooms to socialize with others. But all these Web services designed for human use are being abused by computer programs (bots).

- Free e-mail services

For people's convenience, Hotmail, Yahoo, and others provide free e-mail services. But malicious programmers have designed bots to register thousands of free e-mail accounts every minute. These bots-created e-mail accounts not only waste large amounts of disk space of the service providers, but they are also being used to send thousands of junk e-mails [1,3,10].

- Online polls and recommendation systems

Online polling is a convenient and cost-effective way to obtain people's opinions. But if the polls are abused by bots, their credibility is ruined. In 1998, http://www.slashdot.com released an online poll asking for the best computer science program in the US [1]. This poll turned into a bots-voting competition between MIT and CMU. Clearly, in this case the online poll has lost its intended objectives. A similar situation arises in online recommendation systems. For example, at Amazon.com, people write reviews for books, recommending that others buy or not buy a particular book. But if malicious bots start to write book reviews, this online recommendation system becomes useless.

- Chat rooms

In the information age, people use online chat rooms to socialize with others. But bots start to join chat rooms and point people to advertisement sites. Chat room providers such as Yahoo and MSN do not like the bots because they irritate human users and decrease human users' visits to their sites.

- Meta services and shopping agents

Meta service is unwelcome among E-commerce sites and search engines [15]. In the case of E-commerce, a malicious programmer can design a bot whose task is to aggregate prices from other E-commerce sites. Based on the collected prices, the malicious programmer can make his/her price a little cheaper, thus stealing away other sites' customers. Meta services are good for consumers, but E-commerce owners hate them because they consume a site's resources without bringing in any revenue. Similar situations arise with search engine sites.

The above real-world issues have recently generated a new research area called human interactive proofs (HIP) whose goal is to defend services from malicious attacks by differentiating bots from human users. The design of HIP systems turns out to have a significant relationship with the famous Turing test.

In 1950, Turing proposed a test whose goal was to determine if a machine had achieved artificial intelligence (AI) [12]. The test involves a human judge who poses questions

to a human and a machine and decides which of them is human based on their answers. So far, no machine has passed the Turing test in a generic sense, even after decades of active research in AI. This fact implies that there still exists a considerable intelligence gap between humans and machines. We can therefore use this gap to design tests to distinguish bots from human users. HIP is a unique research area in that it creates a win-win situation. If attackers cannot defeat a HIP algorithm, that algorithm can be used to defend Web services. On the other hand, if attackers defeat a HIP algorithm, then they have solved a hard AI problem, thus advancing AI research.

So far, there exist several HIP algorithms, but most of them suffer from one or more deficiencies in ease of use, resistance to attack, dependency on labeled databases, and lack of universality (see Sect. 3 for details). This paper makes two major contributions. First, based on both theoretical and practical considerations, we propose a set of HIP design guidelines that ensure a HIP system's security and usability. Second, we propose a new HIP algorithm based on detecting human faces and facial features. The human face is the most familiar object to humans, rendering it possibly the best candidate for HIP.

We name our HIP algorithm ARTiFACIAL, for Automated Reverse Turing test using FACIAL features. It relates to (and differs from) the original Turing test in several respects. First, our test is automatically generated and graded, i.e., the Turing test judge is a machine instead of a human. Second, the goal of the test is the reverse of the original Turing test – we want to differentiate bots from humans, instead of proving that a bot is as intelligent as humans. These two features constitute the first three letters (ART) in ARTiFACIAL: Automated Reverse Turing test.

ARTiFACIAL works as follows. Per each user request, it automatically synthesizes an image with a distorted face embedded in a cluttered background. The user is asked to first find the face and then click on six points (four eye corners and two mouth corners) on the face. If the user can correctly identify these points, ARTiFACIAL concludes the user is a human; otherwise, the user is a machine. We conduct user studies and show the ease of use of ARTiFACIAL for human users. We design attacks using the best existing face detectors and demonstrate the difficulty ARTiFACIAL presents for malicious bots.

The rest of the paper is organized as follows. In Sect. 2 we discuss related work, which mainly uses letters, digits, and audio. In Sect. 3 we propose a set of design guidelines that are important to the success of a HIP algorithm. We further evaluate existing HIP algorithms against the proposed guidelines. In Sect. 4 we first give a brief review of various face detection techniques and point out their limitations. Based on these limitations, we then design ARTiFACIAL, covering a 3D wire model, cylindrical texture map, geometric head transformation and deformation, and appearance changes. To demonstrate the effectiveness of a HIP algorithm, we need to at least show that it is easy for humans and very hard for computer programs. In Sect. 5 we describe our user study design and results, showing the ease of use to human users. In Sect. 6 we present various attacks to ARTiFACIAL using the best existing techniques. The results show that ARTiFACIAL has very high resistance to malicious attacks. We give concluding remarks in Sect. 7.

2 Related work

While HIP is a very new area, it has already attracted researchers from AI, cryptography, signal processing, document understanding, and computer vision. The first idea related to HIP can be traced back to M. Naor, who wrote an unpublished note in 1996 entitled "Verification of a human in the loop or identification via the Turing test." The note contained many important intuitive thoughts about HIP but did not produce functional systems. The first HIP system in action was developed in 1997 by researchers at Alta Vista (see http://altavista.com/sites/addurl/newurl). Its goal was to prevent bots from adding URLs to the search engine to skew the search results. The specific technique they used was based on distorted characters, and it worked well in defeating regular optical character recognition (OCR) systems.

In 2000, Udi Manber of Yahoo talked to researchers (von Ahn, Blum, Hopper, and others) at CMU that bots were joining in Yahoo's online chat rooms and pointing people to advertisement sites [1,3]. Manber challenged the CMU researchers to come up with solutions to distinguish between humans and bots. Later that year, von Ahn et al. proposed several approaches to HIP. The CMU team so far has been one of the most active teams in HIP, and we highly recommend readers to visit their Web site at http://www.captcha.net to see concrete HIP examples. The CMU team introduced the notion of CAPTCHA: Completely Automated Public Turing Test to Tell Computers and Humans Apart. Intuitively, a CAPTCHA is a program that can generate and grade tests that (i) most humans can pass but (ii) current computer programs cannot pass [1,2]. They have developed several CAPTCHA systems.

- Gimpy

Gimpy picks seven random words out of a dictionary, distorts them, and renders them to users. An example Gimpy test is shown in Fig. 1a. The user needs to recognize three out of the seven words to prove that he or she is a human user. Because words in Gimpy overlap and undergo nonlinear transformations, they pose serious challenges to existing OCR systems. However, they also place a burden on human users. The burden is so great that Yahoo pulled Gimpy off its Web site [3]. The CMU team later developed an easier version, EZ Gimpy, which is shown in Fig. 1b. It shows a single word over a cluttered background and is currently used on Yahoo's Web site.

- Bongo

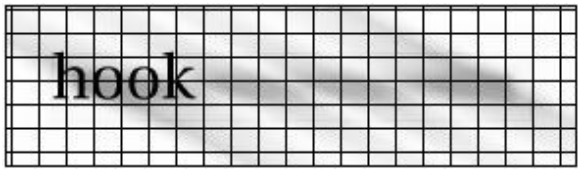
Bongo explores human ability in visual pattern recognition (see http://www.captcha.net/captchas/bongo/). It presents to a user two groups of visual patterns (e.g., lines, circles, and squares), named LEFT and RIGHT. It then shows new visual patterns and asks the user to decide if the new patterns belong to LEFT or RIGHT.

- Pix

Pix relies on a large database of labeled images. It first randomly picks an object label (e.g., flower, baby, lion, etc.) from the label list and then randomly selects six images containing that object from the database and shows the images to a user. The user needs to enter the correct object label to prove he or she is a human user.

- Animal Pix

a



b

Fig. 1. a Gimpy. **b** EZ Gimpy

Animal Pix is similar to Pix but differs in the following ways: (i) It uses 12 animals (bear, cow, dog, elephant, horse, kangaroo, lion, monkey, pig, and snake) instead of generic objects as the labeled objects; and (ii) instead of asking a user to enter the object label, it asks a user to select from the set of predefined 12 animals (see CAPTCHA Web site given above).

Almost at the same time that the CMU team was building their CAPTCHAs, Xu et al. were developing their HIP system at Georgia Tech [15]. Their project was motivated by the security holes in E-commerce applications (see meta services in Sect. 1). They developed a new type of trapdoor one-way hash function that transforms a character string into a graphical form such that humans can recover the string while bots cannot.

In the past 2 years, researchers at PARC and UC Berkeley have published a series of papers on HIP [3,4,5]. In their systems, they mainly explored the gap between humans and bots in terms of reading poorly printed texts (e.g., fax prints). In Pessimal Print [5], Coates et al. reported close to zero recognition rates from three existing OCR systems: Expervision, FineReader, and IRIS Reader. In BaffleText [4], Chew and Baird further used non-English words to defend dictionary attacks.

In addition to the above visual HIP designs, there also exist audio challenges, e.g., Eco (see http://www.captcha.net/captchas/). The general idea is to add noise and reverberation to clean speech such that existing speech recognizers can no longer recognize it. The audio challenges are complementary to the visual ones and are especially useful to vision-impaired users.

To summarize, HIP is still a young and developing area. But it has already attracted researchers from cryptograpy, AI, computer vision, and document analysis. The first HIP workshop was held in January 2002 in PARC (see http://www.aladdin.cs.cmu.edu/hips/events/), and [3] provides a good summary. For a new area to develop and advance, researchers must formulate design guidelines and evaluation criteria. The CMU and PARC teams have proposed many of the crucial aspects of HIP. In the next section, we present further guidelines on how to design a practical HIP system and evaluate existing approaches and our proposed approach against the guidelines.

3 HIP guidelines

The CMU and PARC researchers have summarized the following desired properties of a HIP system [1,3]:

1. The test should be automatically generated and graded by a machine.
2. The test can be quickly taken by a human user.
3. The test will accept virtually all human users.
4. The test will reject virtually all bots.
5. The test will resist attacks for a long time.

The above five properties capture many important aspects of a successful HIP system. But we realize that there are other theoretical and practical considerations that need to be taken into account. Furthermore, we think it would be beneficial to the HIP community if the desired HIP properties were orthogonal to each other and could be clearly evaluated. We therefore propose the following new guidelines for designing a HIP algorithm:

1. **Automation and gradability**. The test should be automatically generated and graded by a machine. This is the same as the old guideline and is the minimum requirement of a HIP system.
2. **Easy for humans.** The test should be quickly and easily taken by a human user. Any test that requires longer than 30 s becomes less useful in practice.
3. **Hard for machines.** The test should be based on a well-known problem that has been investigated extensively, and the best *existing* techniques are far from solving the problem. This guideline consolidates the old guidelines 4 and 5. The old guideline 4 is a consequence of this new guideline. The old guideline 5 is difficult to evaluate against, i.e., it is difficult to define "for a long time". Instead of predicting the *future*, we only require that the problem be a well-known problem, and the best *existing* techniques are far from solving the problem. This new guideline avoids the interrelationship between old guidelines 4 and 5 and is much easier to evaluate against. An example problem that satisfies our requirement is "automatic image understanding", which is well known and has been investigated for more than three decades but is still without success. On the other hand, printed clean text OCR is not a hard problem, as today's existing techniques can already do a very good job. As pointed out by von Ahn et al., HIP has an analogy to cryptography: in cryptography it is assumed that the attacker cannot factor 1024-bit integers in a reasonable amount of time. In HIP, we assume that the attacker cannot solve a well-known hard AI problem [2].
4. **Universality**. The test should be independent of a user's language, physical location, and educational background, among others. This new guideline relates to the old guideline 3 but is more concrete and clearer to evaluate against. This guideline is motivated by practical considerations and is especially important for companies with international customers, e.g., Yahoo and Microsoft. It would be a nightmare for Yahoo or Microsoft if they had to localize a HIP

Table 1. Evaluation of existing HIP tests against the proposed criteria

Guidelines	1. Automation and gradability	2. Easy to human	3. Hard to machine	4. Universality	5. Resistance to no-effort attacks	6. Robustness when database publicized
Gimpy	Yes	Yes But the partially overlapped text can be hard to recognize [4]	No It has been broken [8]	No People who know English have much more advantages	Yes	Yes
EZ Gimpy	Yes	Yes	No It has been broken [8]	Yes	Yes	No Has only 850 words [4]
Bongo	Yes	Yes	Yes	Yes	No A machine can randomly guess an answer	Yes
Pix	Yes But the labels can be ambiguous (cars vs. White cars)	Yes	Yes	No Some objects do not exist in some countries.	Yes	No With the database, it becomes simple image matching.
Animal Pix	Yes	Yes	Yes	No Some animals are only popular in a few countries.	No A machine can randomly guess an answer	No With the database, it becomes simple image matching.
Pessimal	Yes	Yes	Yes	No People who know English have much more advantages	Yes	No Has only 70 words [4][5]
BaffleText	Yes	Yes But has been attacked when using single font [4]	Yes	Yes But people who know English may have advantages	Yes	Yes
Byan	Yes	Yes	Yes	No Users need to know English	Yes	Yes
ARTiFACIAL	Yes	Yes	Yes	Yes	Yes	Yes

test in 20 different languages. As an example, no digit-based audio HIP tests are universal because there is no universal language on digits (even though visually they are the same). A different HIP test would have to be implemented for each different language, and thus not be cost effective. Strictly speaking, no HIP test can be absolutely universal as no two humans are the same. However, we can make reasonable assumptions. For example, we can consider EZ Gimpy as universal because if a user can use a computer, it is reasonable to assume he or she knows the 10 digits and the 26 English letters. In contrast, Gimpy is not as universal as EZ Gimpy because users who know English have a much better chance at succeeding. Gimpy is quite difficult for non-English speakers.

5. **Resistance to no-effort attacks.** The test should survive no-effort attacks. No-effort attacks are those that can solve a HIP test without solving the hard AI problem. For example, Bongo is a two-class classification challenge (Sect. 2). To attack Bongo, the attacker needs no effort other than always guessing LEFT. This will guarantee that the attacker will achieve 50% accuracy. Even if Bongo can ask a user to solve 4 tests together, that still gives no-effort attacks 1/16 accuracy. Animal Pix is another example of a challenge that will not survive a no-effort attack. Because there are 12 predefined animal labels, a no-effort attack can achieve 1/12 accuracy without solving the animal recognition problem. The HIP tests that cannot survive no-effort attacks do not have practical usefulness and cannot advance AI research.
6. **Robustness when database is publicized.** The test should be difficult to attack even if the database from which the test is generated is publicized. For example, both Pix and Animal Pix would be very easy to attack once the database is publicly available. They therefore are not good HIP tests [1].

Compared with the five old guidelines, the six new proposed guidelines are more comprehensive, more orthogonal to each other, and more clear to evaluate against. We summarize the evaluations of the existing approaches against the new guidelines in Table 1. From Table 1 it is clear that most of the existing HIP algorithms suffer from one or more deficiencies. In the following section, we propose a new HIP algorithm: ARTiFACIAL, which is based on detecting human faces and facial features. It is easy for humans, hard for bots, universal, survives no-effort attacks, and does not require a database.

4 Proposed test – ARTiFACIAL

Human faces are arguably the most familiar object to humans [9,11,13], rendering them possibly the best candidate for HIP. Regardless of nationality, cultural differences, or educational background, we all recognize human faces. In fact, our ability to recognize human faces is so good that we can recognize them even if they are distorted, partially occluded, or poorly illuminated.

Computer vision researchers have long been interested in developing automated face detection algorithms. A good survey paper on this topic is [17]. In general, face detection algorithms can be classified into four categories. The first is the knowledge-based approach. Based on people's common knowledge about faces, this approach uses a set of rules to do detection. The second approach is feature based. It first detects local facial features, e.g., eyes, nose, and mouth, and then infers the presence of a face. The third approach is based on template matching. A parameterized face pattern is predesigned manually and then used as a template to locate faces in an image. The fourth approach is appearance based. Instead of using predesigned templates, it learns the templates from a set of training examples. So far, the fourth approach has been the most successful one [17].

Despite decades of hard research on face and facial feature detection, today's best detectors still suffer from the following limitations:

1. **Head orientation**. Let axis x point to the right of the paper, axis y to the top of the paper, and axis z outside the paper. All face detectors handle frontal face well. That is, they work well when there is no rotation around any of the three axes. They can also handle rotations around the y-axis to some extent, but worse than frontal faces. They do not handle rotations around axes x and z well.
2. **Face symmetry**. Face detectors assume, either explicitly or implicitly, that the faces are symmetric, e.g., the left and right eyes are roughly the same height and are roughly the same distance from the nose bridge.
3. **Lighting and shading**. Face detectors rely on different intensity levels of landmarks on human faces. For example, they assume that the eyes are darker than the surrounding region and the mouth/lip region is also darker than the rest of the face. When a face image is taken under very low or high lighting conditions, the image's dynamic range decreases. This in turn results in difficulties in finding the landmark regions in faces. In addition, lighting also creates shading, which further complicates face detection.
4. **Cluttered background**. If there exist facelike clutters in the background of the face image, the face detectors can be further distracted.

The above four conditions are among the most difficult for automated face detection, yet we humans seldom have any problem under those conditions. If we use the above four conditions to design a HIP test, it can take advantage of the large detection gap between humans and machines. Indeed, this gap motivates our design of ARTiFACIAL. If we apply the HIP criteria to ARTiFACIAL, we see that it is one of the best HIP candidates (Table 1).

ARTiFACIAL works as follows. Per each user request, it automatically synthesizes an image with a distorted face embedded in a cluttered background. The user is asked to first find the face and then click on six points (four eye corners and two mouth corners) on the face. If the user can correctly identify these points, we can conclude the user is a human; otherwise, the user is a machine.

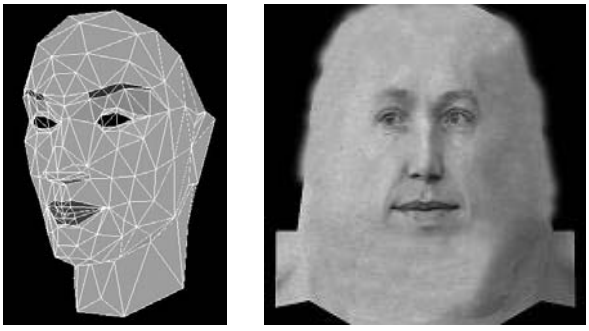
We next use a concrete example to illustrate how to automatically generate an ARTiFACIAL test image, taking into account the four conditions summarized above. For clarity, we use F to indicate a foreground object in an image, e.g., a face, B to indicate the background in an image, I to indicate the whole image (i.e., foreground and background), and T to indicate a cylindrical texture map.

[**Procedure**] ARTiFACIAL

[**Input**] The only inputs to our algorithm are the 3D wire model of a generic head (Fig. 2a) and a 512×512 cylindrical texture map *Tm* of an arbitrary person (Fig. 2b). Note that any person's texture map will work in our system, and from that single texture map we can in theory generate an infinite number of test images.

[**Output**] A 512×512 ARTiFACIAL test image I_F (Fig. 5d) with ground truth (i.e., face location and facial feature locations).

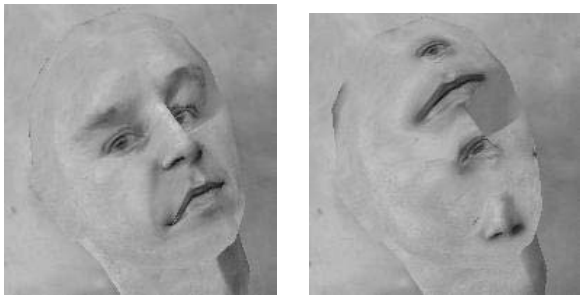
1. Confusion texture map Tc generation
   This process takes advantage of the **cluttered background** limitation to design the HIP test. The 512×512 confusion



a **b**

Fig. 2. a 3D wire model of a generic head. **b** Cylindrical head texture map of an arbitrary person

Fig. 3. The confusion texture map Tc is generated by randomly moving the facial features (e.g., eyes, nose, and mouth) in Fig. 2b to different places such that the "face" no longer looks like a face



a **b**

Fig. 4. a The head after global transformation and facial feature deformation. We denote this head by F_h. **b** The confusion head after global transformation and facial feature deformation. We denote this head by Fc

texture map *Tc* (Fig. 3) is obtained by moving the facial features (e.g., eyes, nose, and mouth) in Fig. 2b to different places such that the "face" no longer looks like a face.

2. Global head transformation
   Because we have the 3D wire model (Fig. 2a), we can easily generate any global head transformations we want. Specifically, the transformations include translation, scaling, and rotation of the head. Translation controls where we want to position the head in the final image I_F, scaling controls the size of the head, and rotation can be around all three axes. At run time, we randomly select the global head transformation parameters and apply them to the 3D wire model texture-mapped with the input texture *Tm*. This process takes advantage of the **head orientation** limitation to design the HIP test.
3. Local facial feature deformations
   The local facial feature deformations are used to modify the facial feature positions so that they are slightly deviated from their original positions and shapes. This deformation process takes advantage of the **face symmetry** limitation to design the HIP test. Each geometric deformation is represented as a vector of vertex differences. We have designed a set of geometric deformations including the vertical and horizontal translations of the left eye, right eye, left eyebrow, right eyebrow, left mouth corner, and right mouth corner. Each geometric deformation is associated with a random coefficient uniformly distributed in [–1,1], which controls the amount of deformation to be applied. At run time, we randomly select the geometric deformation coefficients and apply them to the 3D wire model. An example of a head after steps 2 and 3 is shown in Fig. 4a. Note that the head has been rotated and facial features deformed.
4. Confusion texture map transformation and deformation
   In this step, we apply the same exact steps 2 and 3 to the confusion texture map *Tc,* instead of to *Tm*. This step generates the transformed and deformed confusion head *Fc*, as shown in Fig. 4b.
5. Stage-1 image I_1 generation
   Use the confusion texture map *Tc* as the background B and use F_h as the foreground to generate the 512×512 stage-1 image I_1 (Fig. 5a).
6. Stage-2 image I_2generation
   Make L copies of randomly shrunk *Tc* and randomly put them into image I_1 to generate the 512×512 stage-2 image I_2 (Fig. 5b). This process takes advantage of the **cluttered background** limitation to design the HIP test. Note that none of the copies should occlude the key face regions including eyes, nose, and mouth.
7. Stage-3 image I_3 generation
   There are three steps in this stage. First, make M copies of the confusion head *Fc* and randomly put them into image I_2. This step takes advantage of the **cluttered background** limitation. Note that none of the copies should occlude the key face regions including eyes, nose, and mouth. Second, we now have $M+1$ regions in the image, where M of them come from Fc and one from F_h. Let $Avg(m), m = 0, \ldots, M+1$ be the average intensity of region m. We next remap the intensities of each region m such that $Avg(m)$s are uniformly distributed in [0,255] across the $M+1$ regions, i.e., some of the regions become darker and others become brighter. This step takes advantage of the **lighting and shading** limitation. Third, for each of the $M+1$ regions, randomly select a point within that region that divides the region into four quadrants. Randomly select two opposite quadrants to undergo further intensity changes. If the average intensity of the region is greater than 128, the intensity of all the pixels in the selected quadrants will decrease by a randomly selected amount; otherwise, it will increase by a randomly selected amount. This step takes advantage of both the **face symmetry** and **lighting and shading** limitations. An

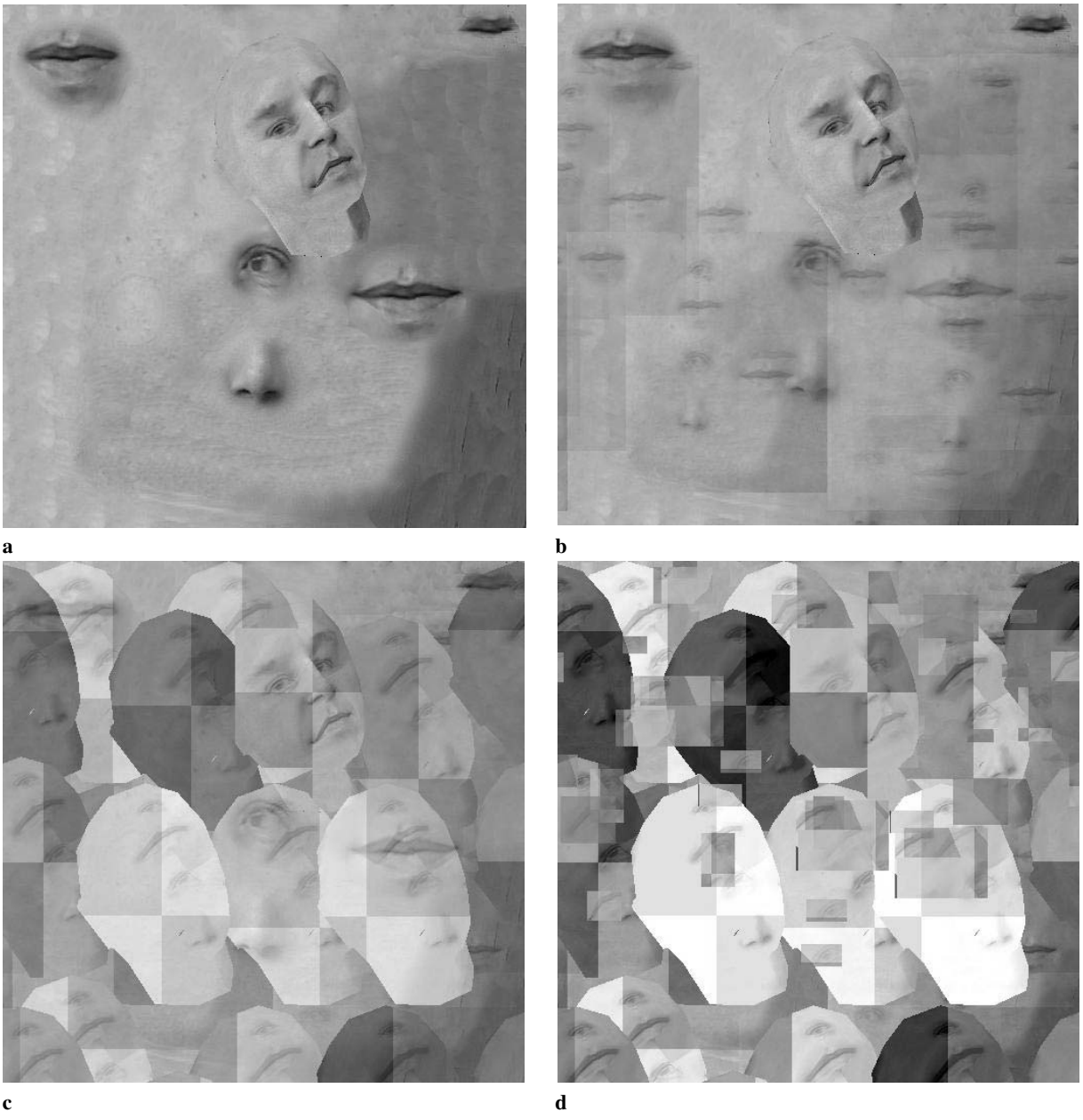


Fig. 5a–d. Different stages of the image. **a** Image I_1. **b** Image I_2. **c** Image I_3. **d** Final image I_F

example I_3 image is shown in Fig. 5c. Note in the image that (i) the average intensities of the $M + 1$ regions are uniformly distributed, i.e., some regions are darker while others are brighter; and (ii) two of the quadrants undergo further intensity changes.

8. Final ARTiFACIAL test image I_F generation
   Make N copies of the facial feature regions in F_h (e.g., eyes, nose, and mouth) and randomly put them into I_3 to generate the final 512×512 ARTiFACIAL test image I_F (Fig. 5d). This process takes advantage of the **cluttered background** limitation to design our HIP test. Note that none of the copies should occlude the key face regions including eyes, nose, and mouth.

The above eight steps take the four face detection limitations into account and generate ARTiFACIAL test images that are very difficult for face detectors to process. We used the above described procedure and generated 1000 images to be used in both user study (Sect. 5) and bot attacks (Sect. 6).

5 User study design and results

For a HIP test to be successful, we need to at least prove that it is easy for human users and very hard for bots. In this section, we design user studies to evaluate human user performance in our test. We will discuss bot attacks in the following section.

5.1 User study design

To evaluate our HIP system across diversified user samples, we invited 34 people to be our study subjects, consisting of

accountants, administrative staff, architects, executives, receptionists, researchers, software developers, support engineers, and patent attorneys. The user study procedure is summarized as follows:

1. A laptop is set up in the subject's office, and the subject is asked to adjust the laptop so that he or she is comfortable using the laptop screen and mouse.
2. The subject is given the following instructions: "We will show you ten images. In each image, there is one and only one distorted but complete human face. Your task is to find that face and click on six points: four eye corners and two mouth corners."
3. The user study application is launched on the laptop. It randomly selects an ARTiFACIAL test image from the 1000 images generated in Sect. 4 and shows it to the subject. The subject detects the face and clicks on the six points. The coordinates of the six points and the time it takes the subject to finish the task are both recorded for later analysis.
4. Repeat step 3 for another nine randomly selected images. Note that no two images of the ten tests are the same.
5. The user study application is closed and the subject is debriefed. At this stage, the subject is given the opportunity to ask questions or give comments on the system and on the study procedure.

Table 2. Average time (in seconds) taken for each of the ten tests. The last column gives the average time over all ten tests

Test	1	2	3	4	5	6	7	8	9	10	Avg
Time (s)	22	15	16	13	12	11	12	12	11	12	14

Table 3. Mismatches (in pixels) of the six points, averaged over the 34 subjects

Points (x, y)	Mismatches (in pixels)
Left corner of left eye	(2.0, 2.3)
Right corner of left eye	(3.3, 5.5)
Left corner of right eye	(3.2, 5.0)
Right corner of right eye	(2.6, 1.8)
Left corner of mouth	(2.5, 1.8)
Right corner of mouth	(2.7, 3.6)

5.2 User study results

Table 2 summarizes the average time taken for each of the ten tests. The numbers are averaged over all 34 subjects. Table 3 summarizes the average mismatch, in pixels, between the ground truth and what were actually clicked for the six points. Combining the statistics in the two tables and feedback obtained during debriefing, we can make the following observations:

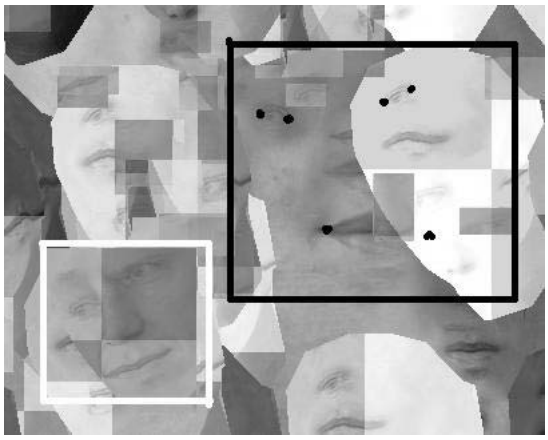


Fig. 6. The only wrong detection made by human users out of 340 tests. The six *black dots* indicate the six points clicked by the human user. The *black bounding box* is inferred from the six points as the user-detected face region. The ground truth face region is shown with a *white bounding box*. We only show part of the test image for clarity

- On average, it takes 14 s for a subject to find the face and click on the six points. This shows that the test is easy to complete for human users. Of the $34 \times 10 = 340$ tests, for the best case, it takes a subject 6 s to finish the task. For the worst case, there are three tests that take longer than 30 s to finish.
- Interestingly, all three of these tests occurred with the same subject. During our debriefing, the subject told us that he was a perfectionist and was willing to spend more time to ensure no mistakes.
- Table 3 tells us that the mismatches between the point coordinates of the ground truth and where the subjects actually clicked are small. This fact allows us to enforce a tight mismatch threshold (in pixels) to efficiently distinguish bots from human users. Currently we set the threshold to be twice the average mismatches (in pixels) as in Table 3. Of the 340 tests, human subjects only made one wrong detection (Fig. 6). The correct rate was 99.7%. During debriefing, the subject told us that she was not paying too much attention for this image but should be able to get it right if given a second chance. Indeed, she only made one mistake out of the ten tests.
- The first test takes longer than the rest of the tests (Table 2). This implies that our instruction may not be clear enough to the subjects. One possible solution, as suggested by several subjects, is to show users an example of the task before asking them to conduct the test.

To summarize, in this section we designed and conducted a user study and demonstrated that the proposed HIP test is easy for humans to take. A byproduct of the user study is that it also provides us with human behavior statistics (e.g., small mismatches for the coordinates of the six points), which enables us to defend our system from attacks.

6 Attacks and results

To succeed in an attack, the attacker must first locate the face from a test image's cluttered background by using a face de-

tector, and then find the facial features (e.g., eyes, nose, and mouth) by using a facial feature detector. In this section we present results of attacks from three different face detectors and one facial feature detector.

6.1 Face detectors

The three face detectors used in this paper represent the state of the art in automatic face detection. The first face detector was developed by Colmenarez and Huang [6]. It uses the information-based maximum discrimination (MD) to detect faces.

The second face detector was developed by Yang et al. [18]. It used a sparse network (SNoW) of linear functions and was tailored for learning in the presence of a very large number of features. It used a wide range of face images in different poses, with different expressions and under different lighting conditions.

The third face detector was developed by Li and colleagues [7,19] following the Viola-Jones approach [14]. They used AdaBoost to train a cascade of linear features and had a very large database consisting of over 10,000 faces. Their system has been demonstrated live in various places and is regarded as one of the best existing face detectors.

We apply the three face detectors to attack the 1000 images generated in Sect. 4. When evaluating the success of an attack, we use very forgiving criteria for the face detectors: as long as the detected face region overlaps with the ground truth face region for 60% (or above), we call it a correct detection. The MD face detector has only one correct detection, the SNoW face detector has three correct detections, and the AdaBoost face detector has zero correct detections. Comparing these results with the 99.7% detection rate of human users, we clearly see the big gap. Figure 7 shows the only correctly detected face region (in black bounding box) by the MD face detector and the ground truth face region (in white bounding box). It is clear that even this "correct detection" is debatable as it is apparently distracted by two dark regions above the true face.

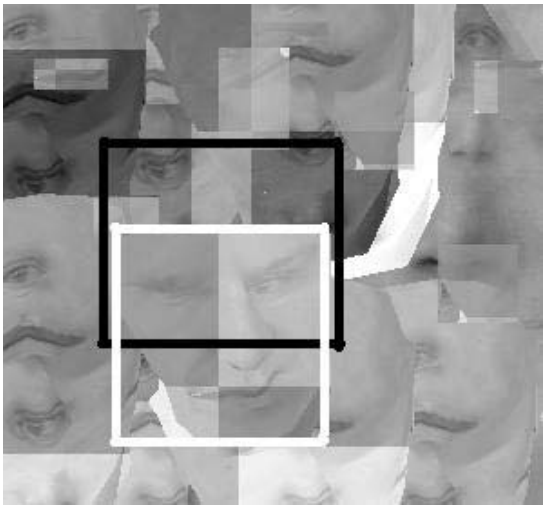


Fig. 7. The MD face detector's best detection out of the 1000 attacks. The detected face region is shown with a *black bounding box* while the ground truth face region is shown with a *white bounding box*. The face detector is distracted by the two *dark regions* above the true face – the face detector thinks the two *dark regions* are left and right eye regions. We only show part of the test image for clarity

Table 4. The number of images with 0, 1, 2, 3, 4, 5, and 6 correctly detected points

Number of correctly detected points	0	1	2	3	4	5	6
Number of images	509	257	114	79	33	6	2

6.2 Facial feature detector

The facial feature detector proposed by Yan et al. [16] is an improved version of the Active Shape Model (ASM). It assumes that a face detector has already found the general location of the face region. It then searches for the facial features in that region. It works quite well with undistorted and clean faces [16].

Again, we use the same 1000 images as our test set. During the attack, to the facial feature detector's advantage, we tell it exactly where the true face is. The detection results over the 1000 test images are summarized in Table 4, and the correct detection rate is only 0.2%.

6.3 Resistance to no-effort attacks

As a final sanity check, let us take a look at ARTiFACIAL's resistance to no-effort attacks.

- The chance for face detectors.

The image size is 512×512, and the face region is about 128×128. It is easy to compute that there are $(512 - 128) \times (512 - 128) = 147,456$ possible face regions in the image. If we allow a 10-pixel mismatch with the ground truth, the chance for a no-effort attack is therefore $(10 \times 10)/147,456 = 6.8\mathrm{E}-4$.

- The chance for facial feature detectors.

If we use the very forgiving mismatch tolerance region of 10×10 for each point, the chance for each point is $(10 \times 10)/(128 \times 128) = 0.0061$. For six points, $0.0061^6 = 5.2\mathrm{E}-14$. The final success rate is the product of the face detector and facial feature detector: $6.8\mathrm{E}-4 \times 5.2\mathrm{E}-14 = 3.5\mathrm{E}-17$.

Before concluding the paper, we would like to make an observation. HIP researchers normally choose hard AI problems to create a HIP test. The hope is that if attackers cannot defeat a HIP algorithm, then that algorithm can be used to defend applications and services; if attackers defeat a HIP algorithm, that means they have solved a hard AI problem, thus advancing the AI research. Mori and Malik's attack on EZ Gimpy is a good example of how HIP motivates people to solve hard AI problems [8]. But we should be careful to note that HIP tests do not necessarily lead to AI advancement. An obvious example is the no-effort attacks. In that case, the HIP

test is broken and there is no AI advancement. We therefore want to advocate the importance of the *presentation* aspect of a HIP system. Even if the problems themselves are hard, but if there is no good way to *present* them to users, e.g., the cases of Bongo and Animal Pix, they are not good HIP tests. Today's HIP researchers have not devoted enough attention to this *presentation* aspect of HIP design.

7 Conclusions

In this paper, we have proposed a set of HIP design guidelines that are important for ensuring the security and usability of a HIP system. Furthermore, we have developed a new HIP algorithm, ARTiFACIAL, based on human face and facial feature detection. Because the human face is the most familiar object to all human users, ARTiFACIAL is possibly the most universal HIP system so far. We used three state-of-the-art face detectors and one facial feature detector to attack our system, and their success rates were all very low. We also conducted user studies on 34 human users with diverse backgrounds. The results have shown that our system is robust to machine attacks and easy for human users.

Acknowledgements. We would like to thank Z. Xiong, University of Illinois at Urbana-Champaign, for helping us run the Maximum Discrimination face detector on our test images, M.-H. Yang of Honda Research Institute for helping us run the SNoW face detector on our test images, S. Li of Microsoft Research Asia for providing the AdaBoost face detector, S.C. Yan of Microsoft Research Asia for providing the facial feature detector, and Henrique Malvar, Cem Paya, and Patrice Simard of Microsoft for valuable discussions.

References

1. Ahn L, Blum M, Hopper NJ (2004) Telling humans and computers apart automatically. Commun ACM 47(2):56–60
2. Ahn L, Blum M, Hopper NJ, Langford J (2003) CAPTCHA: Using hard AI problems for security. In: Proceedings of Advances in Cryptology, Eurocrypt'03, Warsaw, Poland. Lecture notes in computer science, vol 2656. Springer, Berlin Heidelberg New York, pp 294–311
3. Baird HS, Popat K (2002) Human interactive proofs and document image analysis. In: Proceedings of Document Analysis Systems 2002, Princeton, NJ, August 2002, pp 507–518
4. Chew M, Baird HS (2003) BaffleText: a human interactive proof. In: Proceedings of the 10th IS&T/SPIE Document Recognition and Retrieval conference, Santa Clara, CA, 22 January 2003
5. Coates A, Baird H, Fateman R (2001) Pessimal print: a reverse Turing test. In: Proceedings of the IAPR 6th international conference on document analysis and recognition, Seattle, September 2001, pp 1154–1158
6. Colmenarez A, Huang TS (1997) Face detection with information-based maximum discrimination. In: Proceedings of IEEE CVPR, Puerto Rico, July 1997, pp 782–788
7. Gu L, Li SZ, Zhang H-J (2001) Learning probabilistic distribution model for multi-view face detection. In: Proceedings of IEEE CVPR, Hawaii, December, 2001 2:116–122
8. Mori G, Malik J (2003) Recognizing objects in adversarial clutter: breaking a visual CAPTCHA. In: Proceedings of IEEE CVPR, Madison, WI, June 2003, 1:134–141
9. Simion F, Macchi Cassia V, Turati C, Valenza E (2001) The origins of face perception: specific versus non-specific mechanisms. Infant Child Develop 10:59–65
10. Thompson C (2002) Slaves to our machines: welcome to your future as a PC plug-in. Weird Mag 10.10. `http://www.wired.com/wired/archive/10.10/start.html?pg=2`
11. Turati C, Milani I, Simion F, Umiltà C (2002) Newborn preference for faces: what is crucial? Develop Psychol 38:875–882
12. Turing A (1950) Computing machinery and intelligence. Mind 59(236):433–460
13. Valenza E, Simion F, Macchi Cassia V, Umiltà C (1996) Face preference at birth. J Exp Psychol Hum Percept Perform 22:892–903
14. Viola P, Jones M (2001) Robust real-time object detection. In: Proceedings of the 2nd international workshop on statistical and computational theories of vision – modeling, learning, computing and sampling, Vancouver, BC, Canada, 7–14 July 2001, pp 747–755
15. Xu J, Lipton R, Essa I, Sung M, Zhu Y (2003) Mandatory human participation: a new authentication scheme for building secure systems. In: Proceedings of IEEE ICCCN 2003, Dallas, TX, 20–22 October 2003, pp 547–552
16. Yan SC, Li MJ, Zhang HJ, Cheng QS (2003) Ranking prior likelihood Distributions for Bayesian shape localization framework. In: Proceedings of the 9th international conference on computer vision, Nice, France, 14–17 October 2003, pp 51–58
17. Yang M, Kriegman D, Ahuja N (2002) Detecting faces in images: a survey. IEEE Trans Patt Anal Mach Intell 24(1):34–58
18. Yang M, Roth D, Ahuja N (2000) A SNoW-based face detector. In: Solla SA, Leen TK, Muller K-R (eds) Advances in neural information processing systems 12 (NIPS 12). MIT Press, Cambridge, MA, pp 855–861
19. Zhang Z, Zhu L, Li S, Zhang H (2002) Real-time multiview face detection. In: Proceedings of the international conference on automatic face and gesture recognition, Washington, DC, 20–21 May 2002, pp 149–154