This CVPR2015 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

Protecting Against Screenshots: An Image Processing Approach

Alex Yong-Sang Chia [1,2] Udana Bandara [1] Xiangyu Wang [2] Hiromi Hirano [1]
[1] Rakuten Institute of Technology, Tokyo, Japan
[2] Institute of Infocomm Research, A*STAR, Singapore

Abstract

Motivated by reasons related to data security and privacy, we propose a method to limit meaningful visual contents of a display from being captured by screenshots. Traditional methods take a system architectural approach to protect against screenshots. We depart from this framework, and instead exploit image processing techniques to distort visual data of a display and present the distorted data to the viewer. Given that a screenshot captures distorted visual contents, it yields limited useful data. We exploit the human visual system to empower viewers to automatically and mentally recover the distorted contents into a meaningful form in real-time. Towards this end, we leverage on findings from psychological studies which show that blending of visual information from recent and current fixations enables human to form meaningful representation of a scene. We model this blending of information by an additive process, and exploit this to design a visual contents distortion algorithm that supports real-time contents recovery by the human visual system. Our experiments and user study demonstrate the feasibility of our method to allow viewers to readily interpret visual contents of a display, while limiting meaningful contents from being captured by screenshots.

1. Introduction

Print-screen key and screen grabber tools (e.g. Windows snipping tool) present a convenient way to capture a bitmap of the contents displayed on a computer monitor. Taking a screenshot of contents seen on a mobile device (e.g. smartphone) can also be readily achieved. As enterprises exploit electronic devices to share documents/images, there is an increasing need from business and social domains for technologies which protect documents/images (especially those whose contents are sensitive e.g. business plans and private chat messages) from being illegally copied by screenshots and thereby comprising the privacy of the data.

Traditional methods to protect against screenshots take a system architectural approach. In methods like [18, 15], a multilevel security operating environment is developed either with dedicated hardware or virtual machines to forbid the saving of screenshots. While this protects against screenshots, it is unrealistic to expect users to modify their hardware architecture or use virtualization technology when viewing copyright data. Hence, such approaches are limited to on-site viewing of data. Other methods [17, 10] install software on users' machines to continually poll for screenshots events, and remove the screen dump bitmap when such events are detected. Such methods do not require expensive changes to system architecture of users' machines. A key issue is on the prevention of the installed software from being overridden by third party drivers. A mobile application that was recently developed to support secure exchange of messages/images is SnapChat [1] which delivers more than 150 million messages/images daily. SnapChat automatically notifies the sender if a screenshot event is detected at the recipient's mobile device.

In this paper, we exploit image processing techniques, coupled with human biological vision, to limit meaningful contents of data presented on a display from being captured by screenshots. Our method takes visual data of the display as input, distorts the visual data, and then presents the distorted data back to the viewer. Given that visual data shown on the display is distorted, screenshots yield little meaningful visual information. A novelty here is that rather than using dedicated hardware or software to recover the distorted contents into a meaningful form, we instead rely solely on human biological vision for direct and automatic recovery.

The underlying idea of our method lies in the findings that humans process visual data as a series of visual snapshots [3, 5]. More importantly, pioneering works on visual scene memory by Pottet et al. [13, 12] showed that there is a brief persistence of visual information in the short-term memory of a viewer, in which the viewer exploits carry-over information from recent fixations to construct a coherent representation of the current scene. Similar conclusions have also been reported by other psychological studies [7, 11, 6]. In this aspect, what we see at an instance in time is a subtle blend of the preceding and present fixa-

tions. This phenomenon is well known and has been deftly exploited in various display devices e.g. flipbooks [9] and persistence-of-vision displays [2, 16].

We exploit this persistence (albeit briefly) of visual information in the short term memory to develop a visual contents distortion algorithm in which distorted visual data can be directly and mentally recovered by the viewer into a meaningful form. Towards this end, we model the blending of information from preceding and present fixations by an additive process, and exploit this in our design of the distortion algorithm. We do not claim our method completely eliminates the threat that visual contents displayed on a screen are stolen. In particular, we note that no method can prevent a committed adversary from recording visual contents of a screen with another imaging device. Nevertheless, we believe the proposed method increases the difficulty of stealing meaningful data with screenshots, and introduces a novel paradigm to limit useful visual information from being captured by screenshots.

1.1. Related Work

A common approach to protect against screenshots is to use application specific plugins to poll for screenshot events e.g. pressing of the print-screen key. Stamp [17] used low level utility functions to intercept and to filter illegitimate screen capture operations. A similar approach was proposed by Okhravi and Nicol [10] where they developed a graphics subsystem which masks surface pixels of administrator windows when the print-screen key is pressed. A central issue here is that presented data on the display is in its undistorted form, and hence such approaches are completely reliant on the integrity of the plugins to detect screenshot events. We depart from this framework and instead present only distorted data on the display. Thus, screenshots of the display yield limited meaningful data.

Gasmi et al. [4] took an Enterprise Right Management (ERM) approach to protect against screenshots, in which applications that are not registered with the ERM controller cannot be executed. Warren [18] proposed a secured operating system which defines the rules an application must abide by (e.g. no screenshots during video playback). The use of virtualization technologies to access copyright data could be used to protect against screenshots and has been advocated by Schmidt et al. [15] and Yu et al. [19, 20]. While such approaches offer protection against screenshots, they demand either the addition of dedicated hardware or modification to existing system architecture and hence cannot be readily implemented on end-user's machines.

To identify screenshots of videos, Scarzanella and Dragotti [14] exploited video jitter as a recognition cue. Lee et al. [8] extracted combing features from video frames and used these features to train a support vector machine to identify if an input image is a screen capture of a video. A weakness of their work is that they learned artifacts that arise from interlaced recordings and is thus ineffective on recordings obtained with more modern progressive devices. SnapChat [1] was recently developed as a mobile application for secure exchange of messages/images. To inhibit recipients from taking screenshots, SnapChat requires recipients to maintain physical contact with the mobile device's touchscreen while viewing the received message/image. Importantly, SnapChat notifies the sender if a screenshot event is detected at the recipient's device. Implicitly, this notification serves as a social deterrence to discourage recipients from taking screenshots. Our method provides a technological solution to deter users from taking screenshots, and has important application in empowering users to securely exchange messages/images. We note here that, unlike our method, [8, 14, 1] are restricted to checking if copyright violation has occurred and cannot actively limit meaningful visual data from being captured by screenshots.

2. Our Approach

Our objective is to distort visual contents of static data (e.g. text) shown on a screen with a focus that humans can automatically recover the distorted contents into its meaningful visual form in real-time. This poses a challenge to conventional visual contents distortion/recovery paradigm: Can a human be directly incorporated into a process to recover distorted visual contents? Recent psychological studies [13, 12, 7, 11, 6] have demonstrated that there is a brief persistence of visual information in a viewer's short term memory. This carryover of visual information from recent fixations helps a viewer constructs a coherent representation of the current scene over several fixations. In the followings, we exploit this persistence of visual information to design a visual contents distortion algorithm, where we model the smooth blending of visual information from previous and present fixations by an additive process.

Fig. 1 outlines our method. Given visual data of the screen, we first compute a set of intermediate distorting planes. Values within these planes are randomly generated and support lossless recovery of the distorted data. We use these planes to generate a set of final distorting planes, and distort the visual data with these planes. The distorted data are then presented in quick succession to the viewer, where meaningful contents of the screen are automatically and mentally recovered by the viewer. While visual quality of the distorted data is weaker than the original unprotected data, we note that there exist numerous business and social settings where adversaries may find the contents of the copied documents to be more important than visual quality of the documents. Given that our framework presents data that is distorted to the viewer, a screenshot yields limited meaningful representation of the visual data.

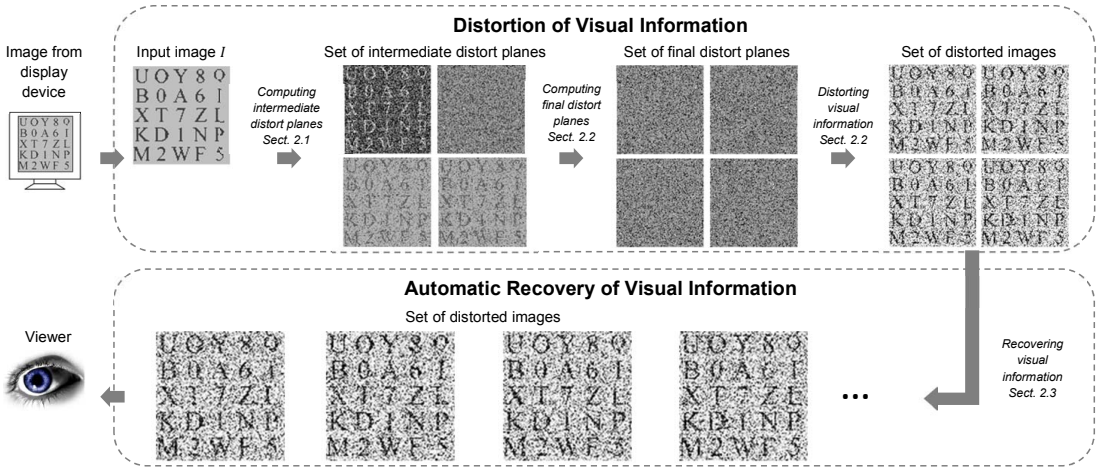

Figure 1: System overview of our method to limit meaningful data from being captured by screenshots. Illustrations shown are obtained with $n = 4$ distort planes. In practice, we use around $n = 22$ distort planes, which increases the method's ability to protect against screenshots.

2.1. Distortion by Random Values

We term a static display of the screen as image I, and denote the intensity at (x, y) coordinate of the screen as $I(x, y)$. Let α and β be respectively the minimum and maximum intensity that can be displayed on the screen. We seek n distorting planes $D_1, \ldots, D_n$ (each with the same dimension as I) that can be arithmetically added to I to hide its contents. In the followings, we term $D_j(x, y)$ as a distorting value and $D_j(x, y) + I(x, y)$ as a distorted value. Importantly, we desire set $\{D_j\}$ to satisfy the following equations,

$$D_j(x, y) = \text{random number}, \qquad j = 1, \ldots, n \quad (1)$$

$$\alpha \;\leq\; D_j(x, y) + I(x, y) \;\leq\; \beta, \qquad j = 1, \ldots, n \quad (2)$$

$$\sum_{j=1}^{n} \Big[D_j(x, y) + I(x, y) \Big] = \sum^{n} I(x, y). \quad (3)$$

Eq. (1) specifies our requirement that each distorting value $D_j(x, y)$ is randomly computed. In this aspect, $I(x, y)$ cannot be recovered from screenshot of distorted pixel $D_j(x, y)+I(x, y)$. Eq. (2) expresses our requirement that a distorted pixel can be displayed on the screen. Most importantly, eq. (3) models our requirement that $D_j(x, y)$ supports lossless recovery of $I(x, y)$ by a human. Specifically, we model the subtle blending of visual information from preceding and present fixations in the short term memory by an additive process. Consequently, eq. (3) models the notion that viewing of the distorted pixels over n period is mathematically equivalent to viewing of the original pixel over the same time period. We note here that an additive process may not be optimum in modeling this blending of visual information by our short term memory. Nevertheless, in our experiments, we found an additive process to be remarkably sufficient for a viewer to mentally recover meaningful visual contents of the original image.

Our aim here is to generate a set of n distorting planes which satisfies eqs. (1) - (3) for an arbitrary value of n, $n \geq 2$. We note that the tight coupling of distorting values $\{D_(x, y), \ldots, D_n(x, y)\}$ by these equations, together with the arbitrary values $I(x, y)$ can take, makes direct computation of a $D_j(x, y)$ challenging. Instead, we recognize that there is a range of values for which $D_j(x, y)$ can take. Thus, rather than computing $D_j(x, y)$ directly, we develop an iterative framework which computes the lower and upper limits of $D_j(x, y)$ at each j^{th} iteration and exploits these limits to compute a random value for $D_j(x, y)$.

Let $L_j(x, y)$ and $U_j(x, y)$ denote the lower and upper limits of $D_j(x, y)$ respectively. From eq. (2), we note that $D_j(x, y)$ is bounded as

$$\alpha - I(x, y) \;\leq D_j(x, y) \;\leq \beta - I(x, y), \quad (4)$$

where $D_j(x, y)$ is obtained from eq. (3) as

$$D_j(x, y) = -\sum_{k=1}^{j-1} D_k(x, y) - \sum_{k=j+1}^{n} D_k(x, y). \quad (5)$$

Close examination of eq. (5) reveals a fast method to compute $L_j(x, y)$ and $U_j(x, y)$. We first discuss the computation of $L_j(x, y)$. From eq. (5), we note that $D_j(x, y)$ is dependent on previous sum $\sum_{k=1}^{j-1} D_k(x, y)$. Since $\sum_{j=1}^{n} D_j(x, y) = 0$, as trivially derived from eq. (3), a possible lower limit for $D_j(x, y)$ is $-\sum_{k=1}^{j-1} D_k(x, y)$. At

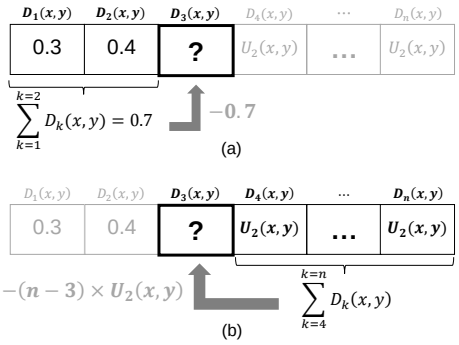

Figure 2: Toy example for computing a lower limit of $D_3(x, y)$. (a) Since $\sum_{j=1}^{n} D_j(x, y) = 0$, a possible lower limit of $D_3(x, y)$ is -0.7. (b) A tighter lower limit for $D_3(x, y)$ can be obtained by assigning $D_4(x, y), \ldots, D_n(x, y)$ to the currently known upper limit value $U_2(x, y)$.

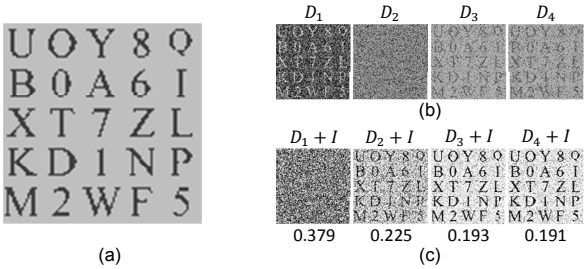

Figure 3: Distorting image I with $\{D_j\}$. (a) Input image I. (b) Set of D_j generated with $n = 4$ and visualized by normalizing between 0 and 1. (c) Distorted images obtained by arithmetically adding D_j to I. RMS distances between distorted images and input image are shown at the bottom of each distorted image. Observe that information is less hidden in latter distorted images, as also indicated by their decreasing RMS values.

the same time, the lower limit of $D_j(x, y)$ is also influenced by future sum $\sum_{k=j+1}^{n} D_k(x, y)$. As such, a lower limit value for $D_j(x, y)$ can be computed by assigning $D_{j+1}(x, y), \ldots, D_n(x, y)$ with the currently known upper limit value $U_{j-1}(x, y)$,

$$\widehat{L_j(x,y)} = -\sum_{k=1}^{j-1} D_k(x,y) - (n-j) \times U_{j-1}(x,y). \quad (6)$$

We illustrate the computation of $\widehat{L_j(x,y)}$ with the toy example in Fig. 2. This, however, ignores the lower bound on $D_j(x, y)$ as given in eq. (4). To ensure the lower limit of $D_j(x, y)$ is within this bound, we compute $L_j(x, y)$ as

$$L_j(x,y) = \max\left(\widehat{L_j(x,y)},\ \alpha - I(x,y)\right). \quad (7)$$

Similarly, $U_j(x, y)$ can be computed as

$$U_j(x,y) = \min\left(\widehat{U_j(x,y)},\ \beta - I(x,y)\right), \quad (8)$$

where

$$\widehat{U_j(x,y)} = -\sum_{k=1}^{j-1} D_k(x,y) - (n-j) \times L_{j-1}(x,y). \quad (9)$$

Given $L_j(x, y)$ and $U_j(x, y)$, we compute $D_j(x, y)$ as

$$D_j(x,y) = \gamma \times L_j(x,y) + (1-\gamma) \times U_j(x,y), \quad (10)$$

where γ is a random number between 0 and 1. Note that initial values $L_0(x, y)$ and $U_0(x, y)$ are respectively $\alpha - I(x, y)$ and $\beta - I(x, y)$, as given in eq. (4). It can be proved that $D_n(x, y) = -\sum_{k=1}^{n-1} D_k(x, y)$ and hence $\sum_{j=1}^{n} D_j(x, y)$ is guaranteed to be equal to 0.

For illustration, we show an image I in Fig. 3(a) and an exemplar set of $\{D_j\}$ generated with $n = 4$ in Fig. 3(b). Distorted images $\{D_j + I\}$ are shown in Fig. 3(c). We compute the root-mean-square (RMS) distances between each distorted image and the input image, and report them at the bottom of the distorted images. As observed in Fig. 3(c), visual contents in I are completely hidden by initial distorting planes but are increasingly revealed at subsequent planes, as also indicated by the decreasing RMS values. This is not surprising, since the computation of $L_j(x, y)$ and $U_j(x, y)$ in eqs. (7) and (8) results in a tighter range from which $D_j(x, y)$ is sampled from at latter iterations. This results in less variations to $D_j(x, y)$. Consequently, for pixels with similar intensity, their distorted values $D_j(x, y) + I(x, y)$ exhibit less variations at latter iterations, and hence visual contents of distorted images become increasingly revealed.

To visualize this observation, we create 1000 test images each of size 100×100. All pixels within each image have the same intensity value that is randomly selected in the range 0 to 1. For each distorting plane D_j, we compute across these 1000 images the range of values from which $D_j(x, y)$ can be sampled and depict it as the blue plots in Fig. 4(a). A blue dot in each panel corresponds to a $D_j(x, y)$ value, where the color intensity increases linearly with the number of pixels that can be assigned with this $D_j(x, y)$ value. Red plots depict the distributions of $D_j(x, y)$ values that are assigned to the pixels. Each red dot corresponds to a $D_j(x, y)$ value whose color intensity increases linearly with the number of pixels assigned with this $D_j(x, y)$ value. As observed from the blue plots, the range of values from which $D_j(x, y)$ is sampled becomes tighter at latter iterations. This results in more pixels being assigned with similar $D_j(x, y)$ values, as shown by reduced spread of the red dots. Thus, $D_j(x, y)$ exhibit less variations at latter iterations, which reduces the information distorting capability of the planes. In the following, we proposed a simple but effective method to circumvent this.

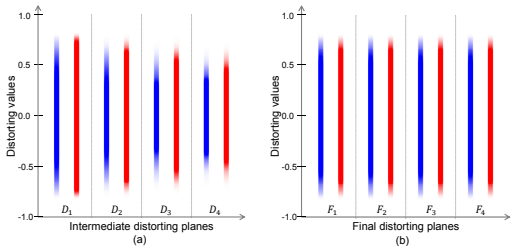

Figure 4: Empirical analysis of the number of pixels that are paired with a distorting value. Results are obtained by 1000 runs on different images (each measuring 100×100), in which pixels of each image have the same intensity value randomly sampled in the range 0 to 1. Blue plots depict the range from which the distorting values can be sampled and assigned to pixels, where the color intensity increases linearly with the number of pixels that can be assigned to a distorting value. Red plots depict the distribution of distorting values that are actually selected and assigned to the pixels, where the color increases linearly with the number of pixels assigned with a distorting value. Note in Fig. 4(a) that decreasing range from which distorting values are sampled (blue plots) result in less variations to the distorting values (red plots). This is effectively resolved in Sect. 2.2, as depicted in Fig. 4(b). Best viewed on screen.

2.2. Random Selection of Distorting Values

We want to ensure information distorting capability is consistent across different distorting planes. This is achieved when distorting values exhibit consistent (and large) variations for each distorting plane. When this is achieved, the distorted values $D_j(x,y) + I(x,y)$ at each j^{th} iteration will be different, even for image pixels with similar intensity. This ensures visual contents of the image can be consistently distorted by the planes. Our goal here is to generate such a set of final distorting planes $\{F_j\}$ which possesses this information distortion capability.

We define set $\Upsilon(x,y)$ to be $\{D_1(x,y), \ldots, D_n(x,y)\}$. Let $F_j(x,y)$ denote a distorting value at (x,y) location of F_j. We obtain $F_j(x,y)$ by randomly selecting without replacement the values from $\Upsilon(x,y)$. This randomly distributes the distorting values at each (x,y) location across different planes, and increases the likelihood that image pixels of similar intensity values are assigned different distorting values by each plane. Correspondingly, this limits the amount of meaningful visual contents revealed in each distorted image. Note that $\{F_j(x,y)\}$ obtained this way are guaranteed to satisfy eqs. (1) - (3) since they are directly obtained from $\Upsilon(x,y)$.

We visualize this improvement in Fig. 4(b). It can be observed that there exists a boarder and more consistent range of distorting values which can be assigned to image pixels (blue plots), as compared to that shown in Fig. 4(a). More importantly, the distribution of the distorting values that are assigned to pixels (red plots) also exhibit larger and more consistent variations across different distorting planes. Consequently, these distorting planes $\{F_j\}$ possess a more consistent ability to hide visual information of an image.

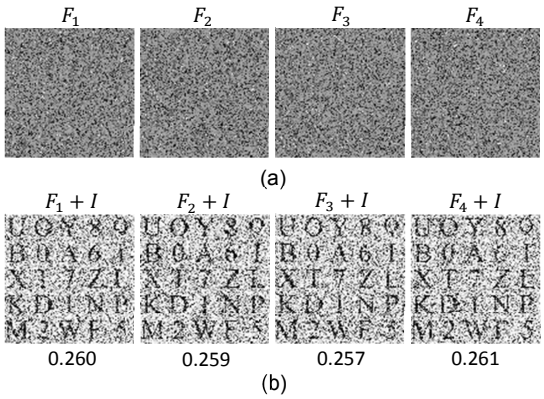

Figure 5: Distorting input image of Fig. 3(a) with final distorting planes $\{F_j\}$. (a) Set of $\{F_j\}$ generated with $n = 4$. (b) Distorted images $\{F_j + I\}$. RMS distances between distorted and input images are shown at the bottom of distorted images.

Given a set of final distorting planes $\{F_j\}$ generated for an input image I, we distort I in a similar way by arithmetically adding each F_j to I. Fig. 5(a) shows a set of final distorting planes $\{F_j\}$ generated with $n = 4$ for the image of Fig. 3(a). We show the corresponding distorted images in Fig. 5(b), where RMS distances between the distorted images and the input image are shown at the bottom of the distorted images. Compared to RMS distances reported in Fig. 3(c), RMS distances obtained by latter distorting planes are higher. This indicates that latter distorting planes can better hide visual information of the input image. We highlight here that while there is a lowering of RMS distances obtained by the earlier distorting planes, a quantitative evaluation across 1000 test images (discussed in Sect. 3.1) demonstrates that RMS distances can be increased by using more distorting planes.

2.3. Recovery of Distorted Visual Contents

We exploit the human visual system to automatically recover visual contents of the distorted data in real-time. Detailed psychological studies on scene memory show visual data to persist briefly in a viewer's short term memory [12, 13], where carryover information from recent fixations empowers the viewer to construct a coherent representation of the scene. In this work, we model this blending of visual contents from recent and current fixations by an additive process. As given in eq. (3), addition of the distorted images provides lossless reconstruction of the original visual contents. Thus, by rapid presentation of the distorted images to the viewer, the blending of the visual contents of distorted images helps a viewer to mentally recover the visual contents of the original image.

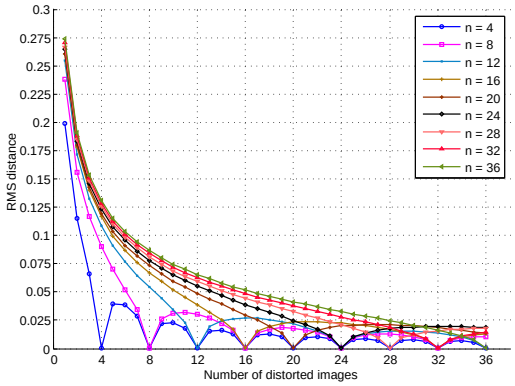

Figure 6: RMS distances of images formed by combining successive distorted images. Given a stream of distorted images produced by a tested n value, we combine the current distorted image with previous distorted images, and compute the RMS distance between the resulting image and the input image. This is repeated 1000 times, each time using input image whose pixels have the same intensity value that is randomly sampled in the range 0 to 1. Figure plots the mean RMS distances for various n values. As confirmed by the plot, perfectly reconstructed image are spaced at n intervals apart. This figure is best viewed in color.

Given an image I, we generate a set of final distorting planes $\{F_j\}$ to distort the image, and present the distorted images $\{F_j + I\}$ to the viewer at rapid succession. This process is applied repeatedly, in which at each repetition, we generate a different set of $\{F_j\}$. The number of F_j in the set, i.e. n, is randomly chosen at run time, $2 \leq n \leq \varphi$. φ is an empirically derived upper bound for the maximum number of distorted images which can be presented to a viewer while ensuring I can be recovered by the viewer. In this work we found $\varphi = 22$. Implicitly, φ models the maximum duration beyond which visual information from previous fixations no longer persists in the viewer's short term memory, and thus I can no longer be recovered.

Eq. (3) ensures contents of an image can be reconstructed losslessly only when distorted images obtained from the *same* set of $\{F_j\}$ are combined. We are aware that this condition is violated when a viewer is presented with a continuous stream of distorted image, since the viewer can mentally combine arbitrary number of successive distorted images together. Nevertheless, the proposed recovery process ensures perfectly reconstructed images to always be n frames apart. We visualize this property in Fig. 6. Given an input image of size 100×100 whose pixels have the same intensity that is randomly sampled in the range 0 to 1, we compute a stream of distorted images for a tested n value. We combine a current distorted image with previous distorted images, and compute the RMS distance between the resulting image and the input image. We repeat this experiment 1000 times and plot the mean RMS distances obtained with various n values. For clarity, we do not show the standard deviation of the RMS distances in the figure. As observed, perfectly reconstructed images (with zero RMS distances) are obtained at every n frames. Our user study (detailed in Sect. 3.2) provides further confirmation that visual contents of input image can be faithfully recovered.

3. Experimental evaluation

We present a detail evaluation of our method to protect contents of documents from being copied by screenshots, and compare it against a baseline method. Test images are generated as follows. Each test image measures 100×100 and contains 25 random characters (alphabets and numbers only) that are arranged in a 5×5 grid. All alphabets are in uppercase. We fix the intensity of the characters and background to be 0.25 and 0.75 respectively. Additionally, we fix the height of each character to be 15 pixels, and the font type to be Time News Roman. Fig. 3(a) shows an example of a synthetic test image. While our focus in this paper is to protect contents of static data against screenshots, we note that our method can be readily extended to protect the visual contents of a movie. We conclude this section by demonstrating one such possible extension in Sect. 3.3.

3.1. Quantitative Evaluation of Distortion Process

First, we quantitatively evaluate the extent of our method to distort image contents. For a test image, we compute a set of distorting planes $\{F_j\}$. We distort the test image with these planes, and compute the average-RMS distances between the distorted and test images. This is repeated across 1000 different test images to obtain 1000 average-RMS distances for a tested n value. We report the mean and standard deviation of the average-RMS distances for each n value in Fig. 7. We observe the mean average-RMS distances at all tested n values to be non-zero, which indicates planes $\{F_j\}$ to possess information distortion capability. It can also be observed that RMS distances increase with larger n values, albeit at a lowering rate. In this aspect, by increasing the number of distorting planes in a set $\{F_j\}$, we can enhance the information distortion capability of the planes.

3.2. User Study

We conducted a user study to qualitatively evaluate the visual data distortion and recovery processes. 5 subjects were engaged for this study. The mean age was 38.6 years, the youngest was 29 years and the oldest was 42 years. All subjects have normal (or corrected) vision. Subjects viewed images from a computer screen (1280×1024, 3×8 bit RGB) that is approximately 50 cm away, and which is connected to a quad core 3.4GHz computer. The refresh rate of the monitor was 60 Hz. We compare our method against a baseline method which distorts the test image by adding salt and pepper noise to the image. To ensure fair comparison against distorted images generated by our method, the noise

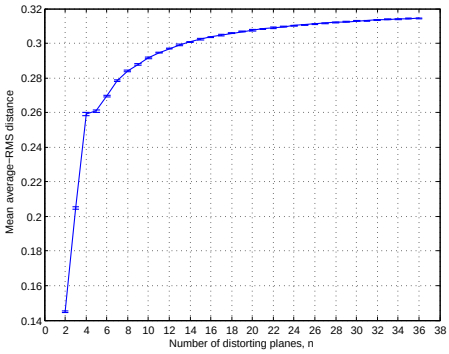

Figure 7: Quantitative evaluation of RMS distances at different n values. For a test image, we generate a set of distorting planes and evaluate the average-RMS distances between the distorted and test images. Graph plots the mean average-RMS distances evaluated across 1000 randomly generated test images at different n values. Vertical bars denote standard deviation of the average-RMS distances. Note that all distorting planes exhibit distortion capability, where stronger distortion is achievable at larger n values.

density was empirically tuned such that the RMS distance between the test image and the distorted image of the baseline method is within a small range of ± 0.005 away from the RMS distance obtained with our distorted image.

In the experiments that follow, we present either a single image (for evaluating the distortion process) or a continuous stream of images (for evaluating the recovery process) to subjects. We present images at a mean rate of one image per 0.0156 second (standard deviation of 7 $\times 10^{-5}$). Subjects were instructed to identify the characters seen in the presented images. Images produced by the baseline and the proposed methods are shown in random order to avoid bias against either methods. Beforehand, all subjects were told presented images contain 5×5 alphanumeric (i.e. alphabets and numbers only) characters, of which alphabets are shown in uppercase, but were otherwise naïve about the methods used to generate the images. Additionally, all subjects were told that their timing would not be recorded and to focus only on the accuracy of their inputs (except the speed evaluation experiment). To rigorously evaluate the distortion and recovery processes, subjects were also instructed to identify characters of badly distorted images on a best-effort basis. At the beginning of each study, 5 images were used to familiarize the subjects with the input interface. Responses for these images were excluded from all data analysis.

3.2.1 Evaluating Visual Information Distortion

We first evaluate the distortion process. For a tested n value, we generate a test image, and distort it with our method to produce n distorted images. We randomly select one from the n distorted images, and compute the RMS distance between the test image and the selected distorted image. Using the baseline method, we generate another distorted image which has similar RMS distance to the test image. Distorted images produced by the baseline and our methods at various n values are then presented to subjects in random order. We define an accuracy score as the percentage of characters that are identified correctly by the subjects. Different test images are used for different subjects.

Fig. 8(a) reports the accuracy scores of our method in blue and the baseline method in black for various n values. We fit least squares quadratic curves to the accuracy scores and depict them as dashed curves in the figure. For both methods, we observe subjects to identify fewer characters correctly as the number of distorting planes n increases. This is not surprising as the extent of image distortion increases with larger n, as shown by the increased RMS-distances in Fig. 7. More importantly, subjects attain lower accuracy scores on distorted images generated by our method on majority of the tested n values (33 of 35 tested values). This is also depicted by the lower regression curve of our method. We highlight here that distorted images generated by the baseline and our methods have similar RMS-distances (within a range of ± 0.005). In this aspect, this demonstrates our method to distort image can better hide its visual contents. A paired t-test shows this result to be significant, ($\rho < 10^{-9}$).

3.2.2 Evaluating Visual Information Recovery

In the second study, we evaluate the feasibility of our method to empower humans to automatically recover visual contents of distorted images. For a tested n value, we generate a stream of distorted images, and present these distorted images to subjects. Subjects were tasked to report the characters seen on the screen. Higher accuracy scores imply better ability of the method to support real-time and automatic recovery of visual contents. Fig. 8(b) reports the accuracy scores obtained by the subjects on images generated by the proposed and baseline methods for various n values. Accuracy scores of both methods show a general downward trend with increasing n values (as also indicated by their least squares curves). Importantly, we note that subjects show remarkable consistency in recovering visual contents of images distorted by our method where all characters are correctly identified even when as many as 22 distorting planes (i.e. a mean average-RMS distance of 0.31, as given in Fig. 7) are used. This value could be used as an upper bound for the number of distorting planes that can be used while ensuring visual contents of an image can be faithfully recovered by a viewer. Overall, visual contents in images distorted by our method are better recovered by the subjects, as indicated by a paired t-test ($\rho < 10^{-3}$).

Additionally, we evaluate the time and accuracy score at which humans can recover distorted visual contents. The s-

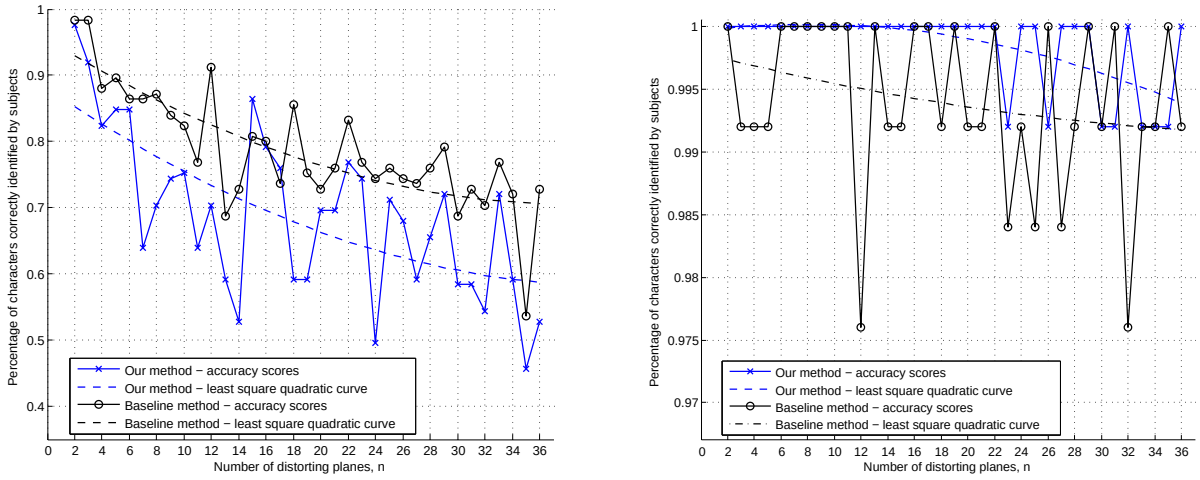

Figure 8: User study results on (a) distortion and (b) recovery processes. Accuracy scores on distorted images generated by our method are shown in blue 'x' and by the baseline method in black '∘'. Least squares curves fitted to accuracy scores are depicted by dashed curves. Overall, our method demonstrates better distortion capability, as confirmed by a paired t-test ($\rho < 10^{-9}$), while supporting better visual contents recovery ($\rho < 10^{-3}$). See text for details. This figure is best viewed in color.

Table 1: Time and accuracy of visual information recovery at different contrast levels.

Contrast Level		1.5	2.0	2.5	3.0
Distorted text	Time	15.75s	9.25s	8.16s	8.07s
	Accuracy	78%	97%	100%	100%
Undistorted text	Time	8.75s	9.00s	8.33s	8.1s
	Accuracy	100%	100%	100%	100%

tudy is conducted under different contrast levels by varying the intensity ratio of background and characters from 1.5 to 3 at intervals of 0.5, where higher values indicate greater contrast. Subjects were instructed to identify characters of distorted images on a best-effort basis in the shortest time possible. We compare the performance against undistorted text and report the results in Table 1. As observed, for distorted text, accuracy and time of recognition deteriorate at low contrast levels, but approaches that of undistorted text as contrast ratio increases. In particular, at contrast levels of 2.5 and 3.0, all visual contents of the distorted text were correctly recovered, in which humans take equal time to recover distorted and undistorted visual contents.

3.3. Extension to movies

Our method can be readily extended to prevent screenshots of movies from capturing meaningful visual contents. Given a currently considered video frame T_i, we identify the next frame T_j in the movie whose RMS distance to T_i is above a user defined threshold τ. We compute n as the number of frames between T_j and T_i, and distort T_i using n distorting planes to form n distorted frames. This process is repeated from T_j until the end of the movie. The distorted frames are then collected together to form a distorted movie. We note that our method provides the distorted movie with an advantage over the original movie during video playback. Specifically, pausing of a distorted movie during playback presents a distorted frame to the user. This empowers users to quickly keep their viewing of on-screen movie contents private from third parties, while allowing them to readily resume playing the movie at a later time.

4. Discussion

We proposed a method to limit meaningful visual information from being captured by screenshots. Our method takes visual data of the screen as input, distorts the visual data, and presents the distorted data back to the viewer. The novelty of our approach lies in the distortion method which exploits findings from psychological studies to empower a viewer to automatically and mentally recover the distorted visual data in real-time. This is a shift from traditional methods which takes a system architectural approach to protect against screenshots. Our experiments and user study demonstrate the feasibility of our method to limit meaningful information from being captured by screenshots, while empowering viewers to readily interpret visual contents of the display. We also demonstrate how the proposed method can be readily extended to protect meaningful visual data of movies from being captured by screenshots. To our knowledge, this is the first approach which exploits image processing techniques to limit useful visual information from being captured by screenshots.

References

[1] S. Berenbaum. 150 million messages are sent on snapchat every day. *http://www.digitaltrends.com/mobile/150-million-snaps-sent-a-day-on-snapchat/*, April 2013. 1, 2

[2] R. Eason. Display apparatus utilizing persistence of vision. *US Patent 5,748,157*, 1994. 2

[3] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona. What do we perceive in a glance of a real-world scene? *Journal of Vision*, (1):1–29, 2007. 1

[4] Y. Gasmi, A. Sadeghi, P. Stewin, M. Unger, M. Winandy, R. Husseiki, and C. Stble. Flexible and secure enterprise rights management based on trusted virtual domains. *ACM workshop on Scalable trusted computing*, pages 71–80, 2008. 2

[5] M. Greene and A. Oliva. The briefest of glances: the time course of natural scene understanding. *Psychological Science*, 20(4):464–472, 2009. 1

[6] A. Hollingworth. Failures of retrieval and comparison constrain change detection in natural scenes. *Journal of Experimental Psychology: Human Perception and Performance*, (2):388–403, 2003. 1, 2

[7] A. Hollingworth. Constructing visual representations of natural scenes: The roles of short- and long-term visual memory. *Journal of Experimental Psychology: Human Perception and Performance*, (3):519–537, 2004. 1, 2

[8] J. Lee, M. Lee, T. Oh, S. Ryu, and H. Lee. Screenshot identification using combing artifact from interlaced video. *ACM Workshop on Multimedia and Security*, pages 49–54, 2010. 2

[9] J. B. Linnet. Improvements in the means of producing optical illusions. *British Patent: N925*, 1868. 2

[10] H. Okhravi and D. Nicol. Trustgraph: Trusted graphics subsystem for high assurance systems. *Computer Security Applications Conference*, pages 254–265, 2009. 1, 2

[11] I. R. Olson, K. S. Moore, M. Stark, and A. Chatterjee. Visual working memory is impaired when the medial temporal lobe is damaged. *Journal of Cognitive Neuroscience*, (7):1087–1097, 2006. 1, 2

[12] M. C. Potter and F. Fox. Detecting and remembering simultaneous pictures in a rapid serial visual presentation. *Journal of Experimental Psychology: Human Perception and Performance*, (1):28–38, 2009. 1, 2, 5

[13] M. C. Potter, A. Staub, J. Rado, and D. H. Connor. Recognition memory for briefly presented pictures: The time course of rapid forgetting. *Journal of Experimental Psychology: Human Perception and Performance*, (5):1163–1175, 2002. 1, 2, 5

[14] M. Scarzanella and P. Dragotti. Video jitter analysis for automatic bootleg detection. *IEEE International Workshop on Multimedia Signal Processing*, pages 101–106, 2012. 2

[15] M. Schmidt, S. Fahl, R. Schwarzkopf, and B. Freisleben. Trustbox: A security architecture for preventing data breaches. *International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 635–639, 2011. 1, 2

[16] J. Spencer, R. Chutorash, S. Geerlings, J. Golden, M. Sims, and R. Eich. Persistence of vision display. *US Patents*, 2006. 2

[17] M. Stamp. Digital rights management: The technology behind the hype. *Journal of Electronic Commerce Research*, (3):102–112, 2003. 1, 2

[18] N. Warren. Palladium and the tcpa. paving the way for future multimedia? *Multimedia systems*, 2003. 1, 2

[19] Y. Yu and T. Chiueh. Display-only file server: A solution against information theft due to insider attack. *ACM workshop on Digital Rights Managment*, pages 31–39, 2004. 2

[20] Y. Yu, H. Kolam, L. Lam, and T. Chiueh. Applications of a feather-weight virtual machine. *ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pages 171–180, 2008. 2