Info Vis

Article

Information Visualization
1–13

DOI: 10.1177/1473871616630015
ivi.sagepub.com
SAGE

Hotspot: Making computer vision more effective for human video surveillance

Cuong Nguyen, Wu-chi Feng and Feng Liu

Abstract

Studies have shown that the human capability of monitoring multiple surveillance videos is limited. Computer vision techniques have been developed to detect abnormal events to support human video surveillance; however, their results are often unreliable, thus distracting surveillance operators and making them miss important events. This article presents *Hotspot* as a surveillance video visualization system that can effectively leverage noisy computer vision techniques to support human video surveillance. *Hotspot* consists of two views: a designated *focus view* to summarize videos with detected events and a *video-bank view* surrounding the *focus view* to display source surveillance videos. The *focus view* allows an operator to quickly dismiss false alarms and focus on true alarms. The *video-bank view* allows for extended human video analysis after an important event is detected. *Hotspot* further provides visual links to assist quick attention switch from the *focus view* to the *video-bank view*. Our experiments show that *Hotspot* can effectively integrate noisy, automatic computer vision detection results and better support human video surveillance tasks than the baseline video surveillance with no or only basic computer vision support.

Keywords

Video surveillance, video visualization, interaction

Introduction

Video surveillance systems have seen increasing use in both public and residential security.[1,2] A common task for a surveillance operator in a video surveillance system is to detect critical events, which involves scanning multiple video streams to search for suspicious activities such as loitering or intrusions.[3–5] Studies, however, have found that human capability for this task is limited.[4,6]

Automated computer vision techniques for event detection have been developed and incorporated to aid human video surveillance. Computer vision algorithms, however, are often unreliable and produce false alarms. False alarms are distracting and can often compromise an operator's surveillance performance. Figure 1(a) illustrates a typical installation of computer vision–enabled surveillance system. The alarms detected by computer vision algorithms can appear in any of the surveillance videos, requiring the operator to shift the attention to follow these alarms. Frequent attention re-orientation is cognitively demanding and makes the operator miss important events, especially those outside his attended area.

In this article, we present a *Hotspot* system that can effectively leverage the automatic event detection capability of computer vision algorithms to assist human video surveillance while minimizing the side effect of the noisy computer vision detection results (http://graphics.cs.pdx.edu/project/hotspot). *Hotspot* is designed according to the understanding of the behavior of off-the-shelf computer vision algorithms and

Department of Computer Science, Portland State University, Portland, OR, USA

Corresponding author:
Feng Liu, Department of Computer Science, Portland State University, 1900 SW 4th Ave., Suite 120, Portland, OR 97201, USA.
Email: fliu@cs.pdx.edu

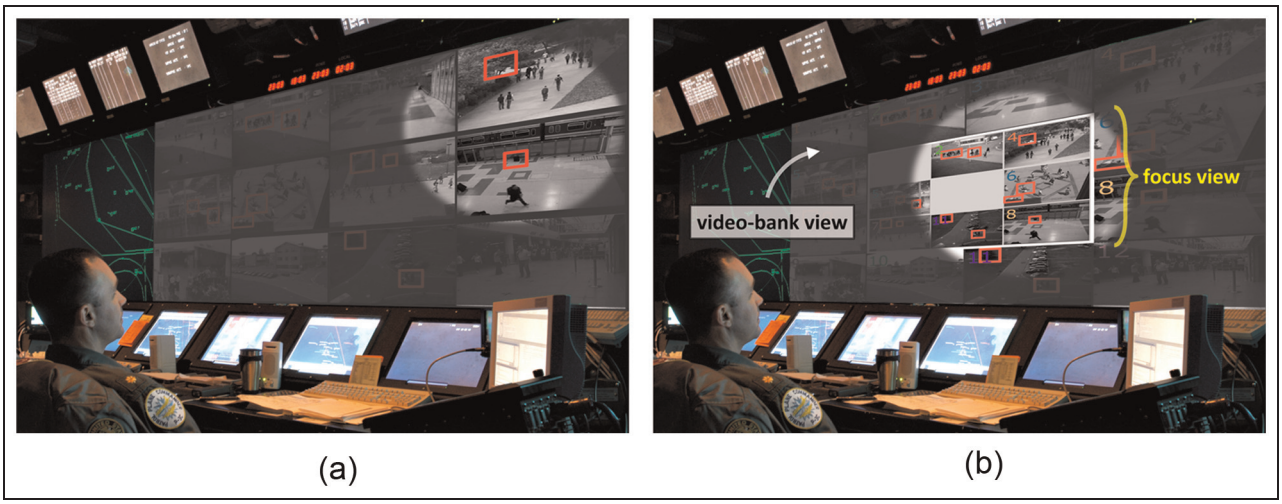

Figure 1. (a) Traditional computer vision–enabled surveillance system versus (b) *Hotspot* system. For illustration purpose only, the operator's attention focus region is highlighted as a bright circle and the screen outside the attention focus region is darkened. Using a traditional video surveillance system, an operator needs to shift his attention across the large display to cover/follow as many computer vision detected alarms as possible. This is cognitively demanding and often makes the operator miss important events. *Hotspot* summarizes the alarms in the focus view and allows the operator to quickly identify truly important events.

their effect on the performance of human video surveillance. As illustrated in Figure 1(b), *Hotspot* consists of two views: a designated *focus view* that summarizes videos with detected events and a *video-bank view* surrounding the *focus view* that displays all the original surveillance videos. The *focus view* provides an overview of detected events in a small region that enables a surveillance operator to quickly and conveniently dismiss false alarms and focus on true alarms. This frees the surveillance operator from frequently switching his attention across a large display area to follow the detected events. The *video-bank view* surrounding the *focus view* displays all the original surveillance videos and provides the useful context information for extended analysis of detected events. For example, the original videos are often organized in the *video-bank view* to preserve the spatial relationship of the surveillance cameras. This allows the operator to track suspicious personnel across multiple neighboring cameras. Accordingly, *Hotspot* designs visual links to provide cues for the operator to quickly switch from the detected event in the *focus view* to the corresponding original video in the *video-bank view*. The *video-bank view* also allows the operator to detect the events that are missed by the computer vision algorithms.

The main contribution of this article is the design of the *Hotspot* system that can effectively leverage noisy computer vision detection results to support human video surveillance tasks. Therein, *Hotspot* addresses two specific problems: how to support a surveillance operator quickly detect abnormal events from the noisy computer vision output and how to enable the operator to switch from the *focus view* to the *video-bank view*. Accordingly, this article conducts two studies to evaluate how *Hotspot* addresses these two challenges. The first study compares *Hotspot* with baseline video surveillance systems with no or only basic computer vision support in the task of important event detection. This study shows that *Hotspot* outperforms the baseline system with no computer vision support and *Hotspot* can better support event detection than the baseline system with the same computer vision support. The second study shows that the design of visual links in *Hotspot* can effectively help a surveillance operator quickly finds the original video in the *video-bank view* corresponding to the video of interest in the *focus view*.

Background

Human video surveillance

This article considers a typical video surveillance scenario where an operator monitors multiple, often more than 10, surveillance videos. These videos are often displayed on a monitor array or on a large screen. The arrangement of the videos on the screen(s) usually preserves the spatial relationship of the cameras capturing these videos. For example, the videos of the same building floor are placed close to each other and the videos captured by spatially neighboring cameras are displayed next to each other.

Surveillance operators need to actively search for suspicious events captured in the videos online. They need to examine each suspicious event to determine

whether it is important or not. Depending on the nature of suspicious events, they can sometimes be easily ascertained as important or unimportant. Sometimes, they require the operators to perform extended analysis, such as observing a suspicious activity for a longer time or tracking a suspicious person across multiple neighboring cameras (videos).

Studies have shown that human video surveillance is cognitively intensive for operators. According to the classic *Feature Integration Theory* in psychology,[7] human video surveillance can be considered as an inefficient search task, which is difficult as it requires an observer to look for complex targets that resemble many features within the environment. It was reported that ordinary surveillance operators can only actively work for about 20–40 min before their detection performance decreases dramatically.[8]

When dealing with many visual features in an inefficient search task, the observer's visual attention plays an important role in the search performance.[9,10] The *Spotlight* visual attention model describes that the human attention region is typically small, and stimuli outside the small attention region are difficult for an operator to detect.[9] Thus, the operator has to actively shift his attention across the display(s) to cover multiple videos. However, attention re-orientation is also cognitively intensive and causes failure of visual awareness. Two common failures of visual awareness are *Inattentional Blindness* and *Change Blindness*.[11] *Inattentional Blindness* happens when an observer fails to detect unexpected visual stimuli that do not receive enough attention from the observer. *Change Blindness* happens when an observer fails to detect changes from visual stimuli when the changes are not shown properly (e.g. changes may happen across the display). When such blindness conditions occur, the operator can miss important events although he is actively monitoring surveillance videos. As the number of videos increases, human video surveillance becomes even more challenging. Without proper attention allocation, the operator can suffer from failures of visual awareness and fail to detect important events.

Computer vision detectors and their effect

A wide variety of computer vision algorithms have been developed to assist human video surveillance or automate video surveillance. These algorithms can automatically detect objects or events of interest, such as human faces and abnormal human activities. Good surveys of these computer vision algorithms can be found in Wang[12] and Dee and Velastin.[13] The detected objects or events are then visualized or emphasized on the screen, such as highlighted by color bounding boxes, to direct operators to look at a particular video.[14] It has been reported that when given cues about where events may happen in the detection task, the operators can devote more attention to the task and can detect events more effectively.[2,9,15]

These computer vision algorithms, however, still fall short and their results are often noisy when applied to many real-world video surveillance scenarios with varying and challenging environment conditions.[16] The fundamental challenge for computer vision remains that semantic visual understanding is still beyond the capability of computer vision algorithms. For event and object detection, there is a key parameter in computer vision algorithms: *detection sensitivity*. Given a detector, a high-sensitivity setting leads to a high detection rate (recall) of true events at the expense of a high false alarm rate; a low-sensitivity setting makes the detector miss a high percentage of true events with a low false alarm rate. In practice, a computer vision detector is typically set to be very sensitive to detect as many important events as possible, as missing important events will pose significant hazards.[17]

A highly sensitive computer vision detector, however, will report many false alarms, which often distract and confuse operators.[15] As illustrated in Figure 1(a), the computer vision detector produces many visual alarms, many of which are false alarms. This makes the detection task more attention demanding since the operator has to ascertain each alarm on the display to identify the truly important ones. Moreover, alarms often appear randomly in the display, and then the operators do not know when and where the alarms will appear next. Thus, they have to shift their attention constantly to try to cover as many alarms as possible. This constant shift of attention between alarms, over a potentially large distance, increases the chance that some alarms may go unnoticed. Due to *Inattentional Blindness* and *Change Blindness*, the operators can still miss important changes in an alarm or miss the alarm entirely while focusing on some others.[14] This article describes a method to effectively make use of noisy computer vision detectors to support human video surveillance.

Related work

A comprehensive survey on video surveillance system design and evaluation is beyond the scope of this article. Please refer to Keval and Sasse[5,18] and Stedmon[19] for a good discussion. Our article is also relevant to the research on video visualization. Please refer to Borgo et al.[20] for a comprehensive survey. This section discusses the relevant work on incorporating computer vision algorithms for video surveillance.

Computer vision algorithms for object and event detection have been used in video surveillance. Some surveillance systems use computer vision algorithms to automatically detect objects of interest and visualize them in a three-dimensional (3D) simulation of the surveillance environment.[21–23] Such systems can benefit space-centric surveillance tasks such as object tracking or path reconstruction. Kurzhal et al. developed a *grid* visualization technique for a single surveillance video. This technique groups objects of interest detected by computer vision algorithms in a grid view.[14] When looking at the grid view, operators can distribute attention toward all detected objects in a single video. The SMV player supports the human tracking task by grouping geographically related videos in a separated view such that a person walking out of one video can be found in an adjacent one.[24]

All these techniques do not thoroughly consider the fact that computer vision algorithms are not perfect and their detection results are noisy. This article considers the behavior of computer vision algorithms and their effect on human video surveillance in designing *Hotspot* that can more effectively incorporate noisy computer vision detection results to assist human video surveillance tasks. We consider our work orthogonal to the previous work and the existing interface and system designs can be used to enhance our *Hotspot* system.

Hotspot

Before we elaborate the interface and design of the *Hotspot* system, we first describe a typical surveillance scenario using *Hotspot*. As shown in Figure 2, there are 12 surveillance videos in total. In order to capture as many true events as possible, the computer vision detector is set to be sensitive; thus, it reports many false alarms. At a particular moment, computer vision algorithms detect six alarms as indicated by red rectangles. To support surveillance monitoring tasks such as important event detection and post-detection analysis, *Hotspot* has two views: the *focus view* and the *video-bank view* as discussed before and illustrated in Figure 2.

Hotspot summarizes the detected important events in the *focus view*. Rather than scanning the entire set of surveillance videos, an operator can look at the *focus view* to find important events. Since the *focus view* is compact, the operator can easily shift his attention among alarms because they are all within his attention focus area. As a result, the operator can quickly dismiss false alarms, find and focus on true alarms.

Hotspot also has a *video-bank view* around the *focus view* that displays all the original videos. This is useful for the operator to capture important events that are missed by the computer vision detector, especially

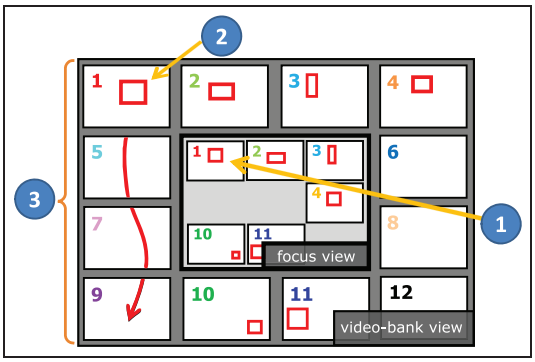

Figure 2. The Hotspot system consists of the hotspot view and the video-bank view. Each pair of videos between two views has a unique *imaginary* visual link based on color-coded label and spatial proximity.

when the detector is set to work at a low-sensitivity level. More importantly, the *video-bank view* is useful for the operator to perform post-detection analysis of the event, such as following the suspicious personnel across neighboring videos (cameras). For example, Figure 2 illustrates a scenario when a detected person of interest in Video 1 is found to be moving through multiple videos. After detecting this person (Step 1), the operator may need to follow him to further study his behavior or to inform the police about his location. In that event, the operator can switch attention to the *video-bank view* (Step 2). In the *video-bank view* (Step 3), the operator can benefit from contextual information such as the spatial relationship between cameras (videos) and can manually follow the suspect moving through these videos. *Hotspot* provides *imaginary* visual links between each video in the *focus view* and the *video-bank view* for the operator to quickly switch from the *focus view* to the *video-bank view*, as detailed later in this section.

Focus view

As discussed previously, computer vision detection results often contain many false alarms. Using a video surveillance system with the naïve computer vision support that highlights the detection results in originally video arrays, an operator has to possibly scan the videos over a large distance, which is a cognitively demanding task and often make the operator miss important events.

Navïe grouping. A potential solution is to group videos with detected events together, as illustrated in Figure 3. In this example, Videos 1, 3, 8, and 10 are moved close to each other. After grouping, the distance among videos of potential interest is small. An

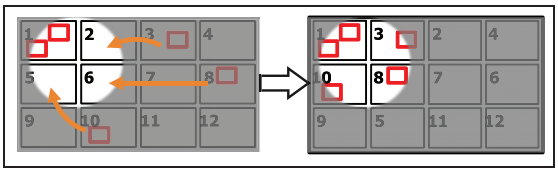

Figure 3. Naive grouping: videos with detected events (i.e. 1, 3, 8, and 10) can be grouped together to support better detection but at the cost of disrupting the video array layout.

operator can quickly shift attention among detected alarms in these videos to identify true alarms.

This navïe grouping method, however, is problematic in that moving videos around changes the layout of surveillance video array, which is distracting, especially while an operator has not finished watching the video being moved. Moreover, the video array layout, or the location of each video in the surveillance screen(s), provides operators with contextual information that is vital for the surveillance tasks.[4] Operators use such contextual information to construct mental maps of the surveillance environment. For example, an experienced surveillance operator can quickly tell which part of the building a video covers based on his mental map of the surveillance environment that has been established as a result of monitoring videos over time. If the video layout is inconsistent, it prevents operators from using contextual information and makes the system extremely hard to use.

Alarm summary in the focus view. *Hotspot* addresses the problem of navïe grouping using two views: a *focus view* and a *video-bank view*, as shown in Figure 2. The *video-bank view* displays all the original videos at fixed locations, thus preserving the useful spatial contextual information for video surveillance. The *focus view* summarizes the detected alarms by displaying the copies of the videos containing these alarms, as illustrated in Figure 2. When a source video has some alarms detected by computer vision algorithms, it is "copied" and displayed in the *focus view* as long as the alarms are present. The layout of the videos in the *focus view* is discussed in section "Visual link" later on. Adding and displaying a new video in the *focus view* can naturally serve as a "pop up" animation to attract an operator's attention toward the newly added video.[10,25]

The *focus view* displays all the alarms in a compact region, thus allowing an operator to quickly dismiss false alarms and detect truly important events. Moreover, the detected alarms always appear in a designated area, that is, the *focus view*. This helps an operator to detect important events, as research in psychology has shown that knowing where a signal will appear can improve the detection performance of that signal.[9]

Video removal. Videos in the *focus view* need to be removed when they no longer contain alarms and provide no interesting information for surveillance so that they do not distract operators. One solution is to allow operators to manually remove videos in the *focus view*. Manual removal, however, can be a heavy burden for operators, especially when the number of alarms is large.[26] Therefore, *Hotspot* automatically removes videos in the *focus view* that do not have any alarms, removing the burden of operators in dismissing them and allowing them to focus on identifying and analyzing truly important events.

In practice, the performance of automatic video removal can sometimes be compromised by the noisy computer vision detection results. The detection of some events may be intermittently interrupted by the environment or due to the limitation of the detection algorithms. For example, a suspicious person is first detected for a few seconds. After that, he is occluded by the environment and is missed by the computer vision detector. And then, he appears again shortly and detected again. Such intermittent detection of an event can affect human video surveillance in a unique way. In the *focus view*, a video appears if and only if it contains some alarms. When a video has intermittently detected alarms, its occurrence in the *focus view* is also intermittent. As a result, such a video may quickly disappear and reappear, creating a type of distracting flashing animation.[25] Moreover, the video may reappear at a different location in the *focus view* as its old location might have been taken by another video with alarms. This inconsistency can compromise the surveillance performance of operators.

To alleviate the intermittent detection problem, *Hotspot* extends the lifetime of a video in the *focus view* to compensate for the interruption time. That is, a video will stay in the *focus view* for a few seconds after the alarm in the video disappears. In this way, when the detection of an event is interrupted for a short amount of time, the video will still stay in the *focus view* and in the same location. A tradeoff of this design is that the duration of videos in the *focus view* that do not exhibit the intermittent detection problem is also extended. During the extended time, these videos contain no alarms and provide no useful information for surveillance. Therefore, a proper delay time needs to be selected. In *Hotspot*, the removal of a video is delayed for 2 s, which is selected based on our test on a very recent object detection algorithm[27] on a few surveillance videos. This test showed that a 2-s extended time is reasonable for reducing the distracting flashing caused by the intermittent detection problem. Furthermore, when a video enters the extended lifetime mode, we gradually reduce its brightness to acknowledge this status to operators. If some alarms in

this video are detected before the extended lifetime period ends, this video is brought back to full brightness with the new alarms.

Post-detection analysis in video bank

Computer vision algorithms often fail on complex video analysis tasks, such as tracking a person of interest across multiple cameras.[24] Semantic visual content understanding is beyond the capability of computer vision even more. Therefore, manual video analysis is necessary for video surveillance. *Hotspot* displays all the original videos in the *video-bank view* to facilitate extended manual video analysis. The *video-bank view* suits such tasks better than the *focus view*. It can enable an uninterrupted observation of the detected event in the video while in the *focus view*, the video will be removed due to the failure of the computer vision detector. More importantly, the *video-bank view* provides useful contextual information to extended manual analysis. For example, when a person of interest is not visible in the current video, the operator can quickly locate him in a neighboring video. This is particularly useful for tasks such as tracking a person of interest across multiple cameras (videos).

Visual link. To use the video bank for post-detection analysis, an operator has to switch his attention from the video in the *focus view* to the corresponding video in the *video-bank view*. To reduce the effort of the operator switching his attention and searching for the corresponding video in the *video-bank view*, *Hotspot* provides "visual links" to help connecting the corresponding videos in the two views.

A straightforward design of visual link is to use some visible graphics like an arrow to explicitly connect the corresponding videos. However, this will not only make the screen very cluttered but also bring in disturbing flashing animation of the graphics when videos are added to or removed from the *focus view*. Therefore, *Hotspot* adopts implicit and imaginary "visual links," which are non-intrusive and provide cues for operators to build the connection. As shown in Figure 2, the visual link design in *Hotspot* uses a combination of color-coded video labels and spatial video arrangement.

Color-coded labels. Color has been shown as an effective feature for visual search.[28] Labels, such as room numbers in a building, are often used to help an operator understand the semantic context of the surveillance area in the video. Other visual cues commonly used in perceptual research are motion, size, or shapes. However, they add complexity to the surveillance screen and therefore are not desirable for video surveillance. *Hotspot* uses color-coded label cues to provide both visual distinctiveness and semantic meaning for each video. Each video has a unique color-coded video label (number) on the top left corner of the video frame. The color is selected based on a distinctive color scheme generated using the CMC(I:c) color difference algorithm.[29]

Spatial cues. When the number of surveillance videos is large, the distinctiveness of both the color and label cues will be reduced. *Hotspot* uses spatial cues to increase the scalability. It has been shown that spatial information of the search target can be used to reduce the search space and improve search performance in the visual search task.[9] Treisman and Gormican[30] also found that the conjunction of spatial and color information can make visual search more efficient.

In *Hotspot*, each video in the *focus view* is positioned as close to the corresponding original video in the *video-bank view* as possible. Specifically, when a new video is copied into the *focus view*, it will be located in a slot in the *focus view* that is available and is closest to its original video according to the Euclidean distance metric. Thus, the video location in the *focus view* will direct an operator's attention toward the part of the screen where the original video is. This spatial cue based on the video arrangement can reduce the search space for operators when they need to search for the original video in the *video-bank view* in a large surveillance system.

Design choices

There are a few design choices that need to be considered. The first is the location of the *focus view*. To effectively support the visual link design that helps an operator to quickly switch from the *focus view* to the *video-bank view*, *Hotspot* positions the *focus view* in the center area of the surveillance screen(s). Figure 4 provides two sample *Hotspot* layouts for 12 and 16 videos, respectively. The second is the size of the *focus view*, which depends on a few factors, including the overall surveillance display area size, the number of video slots in the *focus view*, and the operator's attention focus area. In *Hotspot*, it is useful that the *focus view* stays within or is mostly covered by the operator's attention focus area so that the operator can quickly identify truly important events. *Hotspot* currently adopts an attention-focus-area priority design scheme that sets the size of the *focus view* similar to that of the attention focus area, which is currently determined empirically. The third is the number of video slots in the *focus view*. On one hand, the *focus view* should be able to accommodate all the original videos. On the

Table 1. Simulated surveillance scenarios.

Scenario/task	Important event	True	Missed	False
1. Workload (h) × CV (l)	42	18	24	9
2. Workload (h) × CV (h)	42	39	3	27
3. Workload (l) × CV (l)	18	6	12	3
4. Workload (l) × CV (h)	18	18	0	12

Each scenario can have high (h) or low (l) workload and computer vision (CV) detector performance by adjusting the number of important events and the number of true, missed, and false alarms.

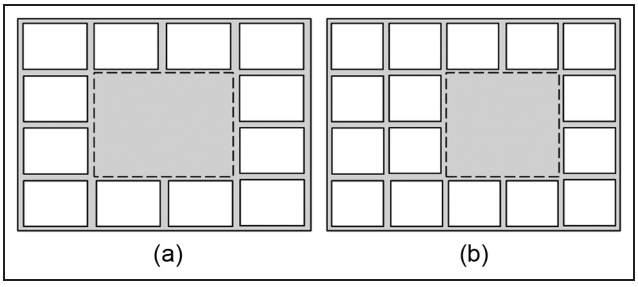

Figure 4. Sample layout designs for *Hotspot* with different numbers of surveillance videos. The dotted rectangles depict the *focus view*. (b) The 16-video layout shows that it is not always possible to position the *focus view* in the very screen center. This figure just provides a few layout samples. The location of the *focus view* can be adjusted as long as it is still in the central region: (a) 12 videos and (b) 16 videos.

other hand, if the number of videos is large, the video in the *focus view* will be too small for the operator to examine the video content. This tradeoff can be made according to the actual surveillance scenario if necessary. For example, for a less busy surveillance scenario, the *focus view* can be set to accommodate fewer videos than the total videos to allow for higher video resolution in the *focus view.*

Experimental evaluation

We conducted two experiments to evaluate *Hotspot* in supporting human video surveillance. The first experiment evaluates how *Hotspot* can support important event detection and the second experiment evaluates how the visual links in *Hotspot* enable quick attention switch from the *focus view* to the *video-bank view* for extended video analysis.

Important event detection

This experiment aims to evaluate how *Hotspot* supports important event detection with noisy computer vision detection results. Participants were asked to use *Hotspot* and two baseline systems to monitor four surveillance scenarios. The baseline systems include a basic video surveillance (BVS) system with no computer vision support, and a video surveillance system with navïe computer vision support (computer vision–supported surveillance (CVS)). The *BVS* system only displays a simple video-bank view. The *CVS* system also displays videos as a video bank and offers detection support from computer vision. Specially, whenever the computer vision algorithm detects an alarm, the detected content is highlighted by a red rectangle in the video, as shown in Figure 1(a). *Hotspot* uses the detection support from the same computer vision algorithm as *CVS*.

Task and stimuli. For each surveillance system, we asked the participant to perform surveillance tasks in four surveillance scenarios, each involving 16 videos. Each video lasts 2 min to prevent the effect of user fatigue. The surveillance scenarios cover two workload levels and two sensitivity levels of the computer vision detector, as detailed in Table 1. In each scenario, the participant was asked to use each of the three systems to click as many videos that contain important events as possible and as fast as possible. We also explicitly asked the participant not to randomly click on videos. As discussed later on, our studies automatically recorded the number of times that each participant clicks on the videos without truly important events and used the data for later analysis.

Like previous work,[6,14,31] we used simulated surveillance environments, events, and computer vision detectors to control the study complexity. Specifically, we used two-dimensional (2D) graphics to render surveillance videos. Figure 5(a) shows a sample video frame. Each video shows a top-down view of a busy intersection. The road intersections and visual landmarks, such as houses, trees, and bushes, are generated randomly so that each video is unique. Most surveillance videos involve human activities. Thus, we randomly added moving human characters in each video.

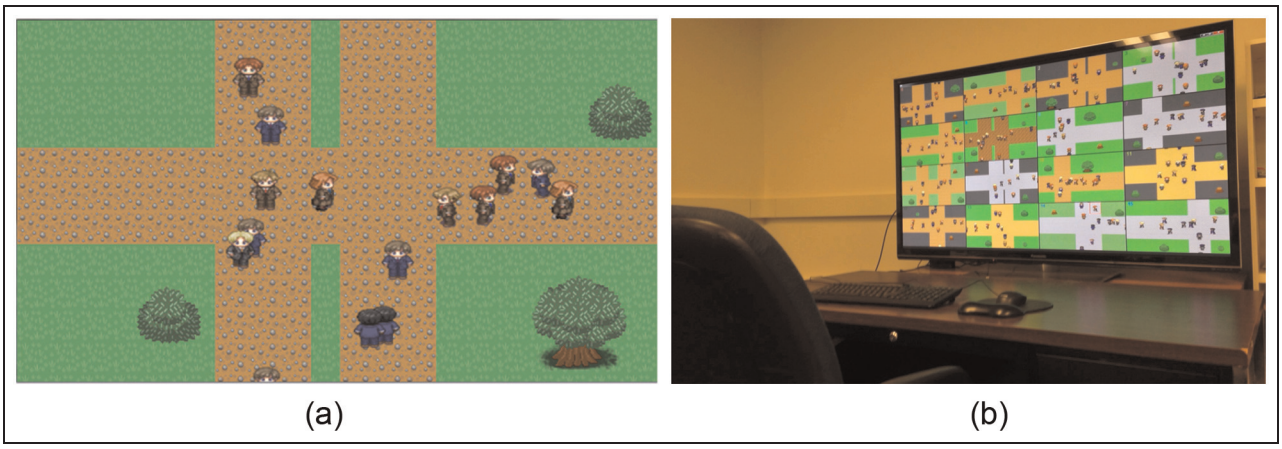

Figure 5. (a) Synthetic video and (b) system setup in studies. Our studies used synthetic videos to simulate surveillance environments with randomly generated landmarks, human activities, and special events for detection.

These characters first appear at one of the road paths and walk along the paths randomly. There are no more than 15 human characters in each video at a time. Please refer to the supplementary video demo for sample videos.

We randomly incorporated the same number of simulated important and unimportant events into the videos. Each event lasts 5 s, which is a typical amount of time that a walking person takes to pass through the field of view of a surveillance camera. After 5 s, each event disappears from the video by having the target character walks off screen or blends in the crowd. Important events are suspicious activities that a participant needs to detect. These events are simulated by modifying the behavior of some human characters in the video. We followed the guideline from Stedmon et al.[4] and simulated three types of important events: person of interest, bag dropping, and trespassing. The person-of-interest event shows a distinctive character who dresses in a black suit and wears a black hat. The bag-dropping event shows a person dropping a suspicious bag. The trespassing event shows a person straying off from the roads and walking on the grass area. Unimportant events are normal activities that look potentially suspicious but are not. These events include walking around, stopping on the side of the roads, or pretending to drop something. We simulated the surveillance workload by controlling the number of important events in the surveillance videos. As shown in Table 1, this study involves two workload levels: 42 and 18 important events, respectively.

A simulated computer vision event detector is incorporated into *Hotspot* and *CVS*. This simulated detector detects some events (including both true and false alarms) and highlights them with red boxes. The numbers of detected true alarms and false alarms are controlled by the sensitivity of the computer vision detector. In this study, we simulated two sensitiveness levels for each workload. At a high-sensitivity level, the simulated detector detects most of the important events at the expense of bringing in many false alarms. Vice versa, at a low-sensitivity level, many important events will be missed with a low level of false alarms. The performance detail of the simulated computer vision detector is reported in Table 1. As noted earlier, a computer vision detector can also produce intermittent detection results. To simulate this effect, randomly selected alarms were set to disappear and reappear in 1 s.

Experimental setup. A 60-in Panasonic display is used for all the surveillance scenarios, as shown in Figure 5(b). The display's resolution is 1920 × 1080 pixels. The videos in both *CVS* and *BVS* systems have the same size, which is 480 × 270. In *Hotspot*, the *focus view* is located in the middle region of the screen, occupying 20% of the display area. Since the *focus view* is large enough, it is set to accommodate the same number of videos as the *video-bank view* if needed. This makes the size of the videos in the *focus view* 200 × 130 pixels and that of videos in the *video-bank view* 380 × 260 pixels. We tested and made sure all videos in all three systems have a high enough resolution for participants to identify activities in the videos. The viewing distance between the display and each participant is 46 in. The display width is 56 in, which makes the field of view of a participant 63°.

A total of 14 participants were recruited from the university campus. These participants had no prior experience with video surveillance. A 10-min training session was provided to make the participants familiar with the detection task and the three systems. A

surveillance scenario with a similar setup to Scenario 1 was used for training.

Each participant conducted the important event detection task in the four scenarios listed in Table 1 using each of the three systems. In total, each participant finished 12 tasks. We used a 3 × 3 Latin square to counterbalance the order of the systems. Before the experiment, each participant was shown the list of the important and unimportant events. We also informed the participants about the sensitivity level of the computer vision detector in *Hotspot* and *CVS*. During each task, the participant was instructed to use a mouse pointer to click the videos with important events. The clicking action was chosen to resemble the real-world practice of surveillance operators to respond to the detection. The system recorded the total number of important events detected by each participant. The system also recorded the number of clicks on videos that do not have any important events. To prevent user errors, the mouse pointer is visualized as a big red arrow to help ensure that the participants did not accidentally click on videos that they did not intend to click.

We hypothesized the following:

1. Systems that have computer vision detection support (*Hotspot* and *CVS*) outperform *BVS* in high workload conditions (Scenarios 1 and 2), where the number of important events is large, making the task challenging.
2. When the workload is high, or the computer vision sensitivity level is high (Scenarios 1, 2, and 4), the number of false alarms increases, compromising the performance of the *CVS* system. The *Hotspot* system outperforms both the *CVS* and *BVS* systems.

Results. For each participant, we calculated the detection rate by dividing the number of detected important events by the total number of important events in the given scenario. We performed Shapiro–Wilk tests on the detection rate measurements to check for normality. The results showed that only the data for *Hotspot* in Scenario 3 were not normally distributed, but all the other data were normally distributed. We also computed the wrong-click rate by dividing the number of clicks on videos without important events by the total number of user clicks. We found that 2 out of the 14 participants have extremely high wrong-click rates 20.8% and 20.9%, respectively. We looked into their click records and found that the wrong clicks of one participant were mainly with *CVS* and those of the other participant were mainly with *Hotspot*. We considered that these two participants did not take the study seriously and removed their data from our later analysis. We analyzed the detection rate measurements with a *System* (*Hotspot*, *CVS*, and *BVS*) × *Scenario* (1, 2, 3, and 4) repeated measures analysis of variance (ANOVA) using Greenhouse–Geisser correction and employed Bonferroni correction for post hoc analysis.

Figure 6(a) provides a summary of the performance analysis with respect to the detection rate of the three systems over each scenario. There was a statistically

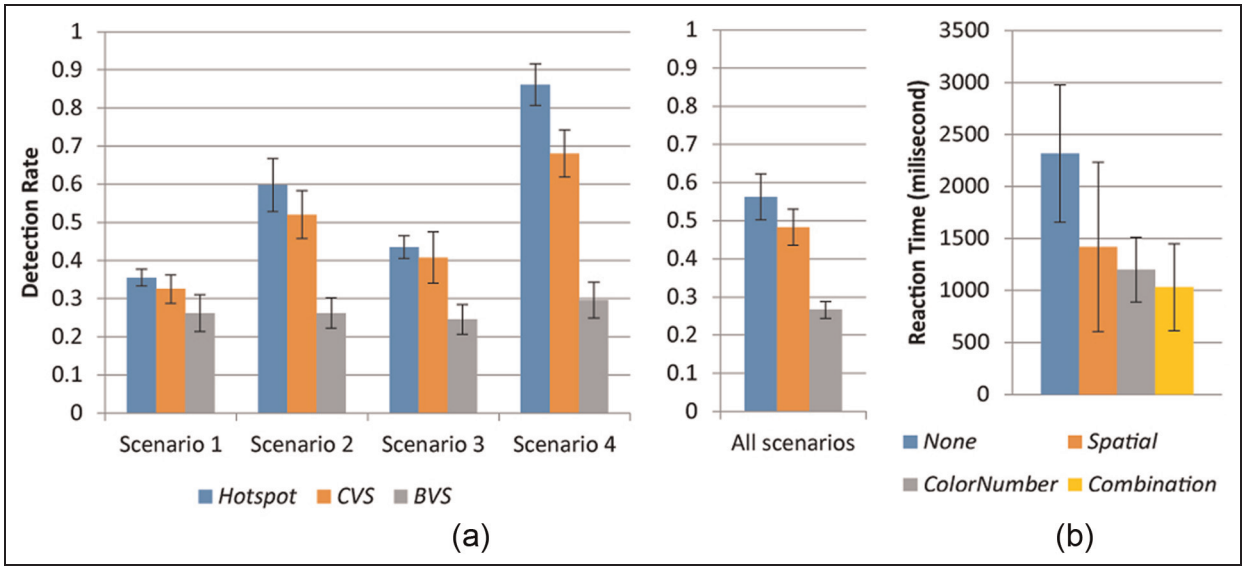

Figure 6. Summary of the study results: (a) detection rate of the three systems over each scenario, and the mean detection rate in all scenarios and (b) mean reaction time of the four visual link techniques.

significant interaction effect between *System* and *Scenario* ($F(6, 66) = 23.36$, $p < 0.0001$); Mauchly's test indicated that the assumption of sphericity had not been violated ($p = 0.88$). Since this interaction effect is significant, it is necessary to analyze whether the effect of *System* on the detection rate depends on each scenario, and whether the effect of *Scenario* on the detection rate depends on each system. Thus, we report these "simple main effects" below.

Testing on the simple main effect of *Scenario* on the detection rate over each system reveals that increasing the computer vision detector sensitivity level in both the high workload (from Scenarios 1 to 2) and low workload scenarios (from Scenarios 3 to 4) improve the detection rate of both *Hotspot* and *CVS*, as shown in Figure 6(a). This increase in performance is because when the sensitivity is increased, the computer vision detector misses less important events although the number of false alarms is increased, so both *Hotspot* and *CVS* benefit from the detection support. The difference in the detection rate between Scenarios 1 and 2 was significant for both *Hotspot* ($p < 0.0001$) and *CVS* ($p < 0.0001$). The difference in the detection rate between Scenarios 3 and 4 was also significant for both *Hotspot* ($p < 0.0001$) and *CVS* ($p < 0.0001$). However, increasing the computer vision detector's sensitivity level also increases the number of false alarms in Scenarios 2 and 4. *Hotspot* can better support important event detection than *CVS*, as shown in Figure 6(a). The detection rate of *CVS* was less than *Hotspot* in both Scenario 2 (*CVS*: M = 0.52, standard deviation (SD) = 0.11; *Hotspot*: M = 0.59, SD = 0.12) and Scenario 4 (*CVS*: M = 0.68, SD = 0.1; *Hotspot*: M = 0.86, SD = 0.09).

Further testing on the simple main effect of *System* over each scenario reveals that the choice of system affects the detection rate in each scenario. Figure 6(a) shows that *BVS* performed worse than both *Hotspot* and *CVS* in all four scenarios. The difference in the detection rate between *BVS* and *Hotspot* was significant in Scenario 1 ($p < 0.02$), Scenario 2 ($p < 0.0001$), Scenario 3 ($p < 0.0001$), and Scenario 4 ($p < 0.0001$). The difference in the detection rate between *BVS* and *CVS* was also significant in Scenario 1 ($p < 0.05$), Scenario 2 ($p < 0.0001$), Scenario 3 ($p < 0.01$), and Scenario 4 ($p < 0.0001$).

This analysis confirmed Hypothesis 1 and shows that the computer vision detection support in both *Hotspot* and *CVS* can improve the detection performance of surveillance operators compared to *BVS*. It is also interesting to note that *Hotspot* and *CVS* also significantly outperform *BVS* when the workload condition is low (Scenarios 3 and 4). This could be that in low workload scenarios, the participants felt bored and were not able to focus on the task. The decrease in an operator's alertness in low workload scenarios has already been studied.[17,32] In *Hotspot* and *CVS*, the computer vision detection support can display visual alarms and increase the alertness of the participants, leading to better performance.

According to the average performance over all scenarios in Figure 6(a), *Hotspot* (M = 0.56, SD = 0.21) outperforms *CVS* (M = 0.48, SD = 0.16). Our analysis reveals that the difference in the detection rate between *Hotspot* and *CVS* was not significant in Scenario 1 ($p = 0.43$), Scenario 2 ($p = 0.25$), and Scenario 3 ($p = 1.0$) but was found to be significant in Scenario 4 ($p < 0.01$). In Scenarios 1 and 2, although the performance of *CVS* is lower than *Hotspot*, the participants might have benefited from the increase in alertness due to high workload conditions. As mentioned above, participants can heavily rely on alarms and focus attention on as many alarms as possible. This makes false alarms not as great a problem in this case. In Scenario 3, the workload is very low. The participants can easily examine the alarms and therefore the false alarms did not cause a significant problem, making *CVS* performs similar to *Hotspot*. In Scenario 4, *Hotspot* outperforms *CVS* significantly. We looked into the simulated computer vision detection results and found that the false alarm rate was very high. This significantly distracts the participants. *Hotspot* can effectively reduce the distraction from these false alarms. Thus, this analysis partially confirmed Hypothesis 2; *Hotspot* can outperform both *CVS* and *BVS* in Scenario 4.

We also looked into the wrong-click rates. For each participant, we computed the average wrong-click rate for each system over four scenarios and analyzed the data with one-way repeated measure ANOVA on *System* (*Hotspot, CVS*, and *BVS*). On average, *Hotspot* (M = 0.086, SD = 0.0512) brought in less wrong clicks than *CVS* (M = 0.105, SD = 0.044) and *BVC* (M = 0.139, SD = 0.080) although the analysis result showed that *System* had no significant effect on the wrong-click rate ($F(2, 22) = 2.34$, $p = 0.12$).

Qualitative feedback. Some participants commented that having computer vision algorithms triggering alarms made the surveillance task less boring and *Hotspot* could particularly wake them up when the task became boring. The feedbacks from participants also confirmed that false alarms raised by the computer vision detector are distracting. They reported that when they were looking at videos with false alarms, they could not catch up with other detected events before these events disappeared. When there were many alarms at a particular moment, participants found it very overwhelming and difficult to examine

all the alarms. They found that *Hotspot* summarizes alarms in the *focus view*, making it easier to examine these alarms.

Visual link

The goal of our second experiment is to evaluate how the visual link design in *Hotspot* enables operators to quickly switch their attention from a video in the *focus view* to the corresponding video in the *video-bank view*. Four visual link support options were tested within *Hotspot*: no visual link support (*None*), the spatial cue (*Spatial*), the color-coded labels (*ColorNumber*), and both the spatial and color-coded label cues (*Combination*).

Task and stimuli. In this experiment, each participant was instructed to match a video of interest in the *focus view* to its corresponding original video in the *video-bank view* as quickly as possible. The same *Hotspot* system used in the first experiment with 16 surveillance videos was used in this experiment. We used the same simulated computer vision detector to detect walking events to show detected videos in the *focus view* for the task. Among these detected videos, four videos, each appearing at a different time in the *focus view*, were selected as the videos of interest for participants to find the corresponding original videos in the *video-bank view*. Each of these four videos was marked with a distinctive red circle. Each video appeared for 10 s and each participant was instructed to click on the marked video and its corresponding video in the *video-bank view* before the video disappeared. We recorded the reaction time between two clicks and the number of matches the participants can complete in each trial. To make the task non-trivial due to having only one video in the *focus view*, we added more videos in the *focus view* whenever a marked video appeared.

We hypothesized the following:

1. *Hotspot* systems that employ visual links will better support operators to switch attention from a video in the *focus view* to the corresponding video in the *video-bank view* than the system without visual links (*None*).
2. The *Combination* system will perform best.

Experimental setup. Eight participants were recruited on the university campus. Each participant was asked to complete the above-mentioned attention switching tasks using each of the four variations of the *Hotspot* systems. The order of the systems was counterbalanced using a 4×4 Latin square. Since the same surveillance scenario was used for each system, the participants may learn the results from the previous trials. To reduce this learning effect, we changed the four videos of interest with the red circles in the *focus view*. We created four different set of videos of interest, one for each trial. Participants performed the study with four set of videos of interest in the same order, but with a system selected using the counterbalanced order. Sufficient training was also provided to allow each participant to practice and to get familiar with the task and the four systems. Each participant conducted four attention switching trials with each system and therefore 16 trails with the four systems in total.

Results. For each participant, the reaction time was measured for each correct match between two clicks from the marked video of interest in the *focus view* and the corresponding original video in the *video-bank view*. If a user failed to match a pair of videos, a 5-s penalty is used as the reaction time for that video pair. This penalty was chosen empirically; we set to half of the time a video of interest would appear in the *focus view*. We computed the mean reaction time of each user for each technique and performed Shapiro–Wilk tests to check for normality. The results showed that the reaction time of *ColorNumber* was not normally distributed. Since we have a smaller sample size in this experiment, we applied a logarithmic transformation to normalize the distribution of the data. We then analyzed the reaction time with a one-way repeated measure ANOVA on *Technique* (*None*, *Spatial*, *ColorNumber*, and *Combination*) and employed Bonferroni correction for post hoc analysis. The mean reaction time of each technique is shown in Figure 6(b).

Our analysis shows that *Technique* had a significant effect on reaction time ($F(3, 21) = 6.37$, $p < 0.005$); Mauchly's test indicated that the assumption of sphericity had not been violated ($p = 0.41$). Figure 6(b) shows that *Combination* (M = 1031.74, SD = 603.12) outperformed all three other techniques. *None* performed the worst among all techniques (M = 2317.11, SD = 951.6). Post hoc analysis reveals that there was a significant effect in the difference between *Combination* and *None* ($p < 0.03$). This analysis partially confirmed Hypothesis 3; the Hotspot system that has visual links (*Combination*) outperforms the system without visual links (*None*). Although *Combination* outperformed both *Spatial* (M = 1417.6, SD = 1177.3) and *ColorNumber* (M = 1201.2, SD = 449.8), the difference between these three techniques was not significant. Thus, Hypothesis 4 could not be confirmed.

Discussion

Our experiments show that *Hotspot* can better support surveillance operators in important event detection

than the BVS systems with no or navïe computer vision detection support. This shows that *Hotspot*, in general, can make better use of noisy computer vision detection results to improve an operator's performance in important event detection than the system with navïe computer vision support. In addition, the degree of improvement depends on factors like the workload and the quality of computer vision detection results. Compared to the system without visual links, the *Hotspot* systems with visual links allow for quicker attention switch from videos of interest in the *focus view* to the corresponding ones in the *video-bank view*.

As discussed earlier, the video capacity of the *focus view* varies with the hardware specs and surveillance scenario. In our experiments, the *focus view* could accommodate all the surveillance videos in the *video-bank view*. However, when the video capacity of the *focus view* is smaller than the total number of surveillance videos, *Hotspot* can possibly leave out some videos with detected alarms. Although increasing the number of videos in the *focus view* can make the videos too small for human video analysis and the human capability to process multiple signals is very limited,[33] it is still helpful to consider this solution for future iterations of *Hotspot*. An alternative solution is to "buffer" the videos with detected alarms that are not displayed in the *focus view* due to its limited capacity and design an effective scheduling algorithm to manage the display of the buffered videos to operators. These problems will be explored in future work.

In our current visual link design, the video background might be similar to the color of the numbers. It is possible to increase the contrast between the numbers and the video background by adding visual effects such as shadow or border. On the other hand, these methods can possibly distract users as they create flashing animation when the videos are added or removed from the focus view. Thus, our system currently chooses to make the visual links more implicit by embedding color in numbers so that they are less intrusive and can still provide cues for attention switch. In addition, our system uses the spatial cues to further enhance visual links especially when the color cues become less effective.

Finally, it is common to use a very sensitive computer vision detector in video surveillance systems deployed in the real world.[2] Therefore, this article focused on supporting operators in detecting important events in surveillance systems with a pretty sensitive computer vision detector so that the computer vision detector misses as few important events as possible at the expense of a high false alarm rate. The presented *Hotspot* system enables an operator to quickly dismiss false alarms and identify truly important events. On the other hand, although a very sensitive setting is applied to the computer vision detector, it can still miss important events. *Hotspot* addresses this problem by keeping all the original videos in the *video-bank view* so that the operator can look into these videos to detect important events missed by the computer vision detector. Moreover, since *Hotspot* can help the operator quickly dismiss false events, the operator can possibly spend more time looking for these missed events. On the other hand, the reliance on the computer vision detector and the *focus view* might also prevent the operator from detecting the important events missed by the computer vision detector as the operator might not pay attention to the videos in the *video-bank view* over time. This problem demands more detailed study and analysis on the behavior and attention focus of operators and will be addressed in future.

Conclusion

This article presented *Hotspot* as a surveillance video visualization system that can effectively make use of noisy important event detection results from computer vision algorithms to support human video surveillance. There are two major contributions in the design of *Hotspot*. First, *Hotspot* summarizes the noisy computer vision detection results in a compact *focus view* that enables an operator to quickly dismiss false alarms and identify truly important events instead of having to scanning across a large display area. Second, *Hotspot* provides visual links that allow for quick attention switch from the video of interest in the *focus view* to the corresponding video in the *video-bank view* for extended event analysis. The experiments showed that *Hotspot* can better support operators in detecting important events than the baseline systems with no or only navïe computer vision support and the visual links can effectively support attention switch from the *focus view* to the *video-bank view*.

Acknowledgements

The authors thank the reviewers of this work for their helpful suggestions. Figure 1 uses an image from Wikipedia under public domain. Figure 5 uses free images from Sithjester.

Funding

This work was supported by the NSF IIS-1321119 and CNS-1218589.

References

1. Räty T. Survey on contemporary remote surveillance systems for public safety. *IEEE T Syst Man Cy C* 2010; 40(5): 493–515.

2. Porikli F, Brémond F, Dockstader SL, et al. Video surveillance: past, present, and now the future. *IEEE Signal Proc Mag* 2013; 30: 190–198.
3. Stedmon A and Harris S. Tracking a suspicious person using CCTV: but what do we mean by suspicious behaviour? In: Bust PD (ed.) *Contemporary ergonomics 2008*. London: Taylor & Francis, 2008, pp. 139–144.
4. Stedmon AW, Harris S and Wilson JR. Simulated multiplexed CCTV: the effects of screen layout and task complexity on user performance and strategies. *Secur J* 2011; 24(4): 344–356.
5. Keval H and Sasse MA. "Not the usual suspects": a study of factors reducing the effectiveness of CCTV. *Secur J* 2008; 23(2): 134–154.
6. Sam Zheng X, Kiekebosch J and Rauschenberger R. Attention-aware human-machine interface to support video surveillance task. *Proc Hum Factors Ergon Soc Annu Meet* 2011; 55(1): 1818–1822.
7. Treisman A and Gelade G. A feature-integration theory of attention. *Cognitive Psychol* 1980; 136: 97–136.
8. Dadashi N, Stedmon A and Pridmore T. Automatic components of integrated CCTV surveillance systems: functionality, accuracy and confidence. In: *Proceedings of the sixth IEEE international conference on advanced video and signal based surveillance, IEEE AVSS'09*, Genova, 2–4 September 2009, pp. 376–381. New York: IEEE.
9. Posner MI, Snyder CR and Davidson BJ. Attention and the detection of signals. *J Exp Psychol* 1980; 109(2): 160–174.
10. Ware C. *Information visualization: perception for design*. Waltham, MA: Morgan Kaufmann Publishers, Inc., 2000.
11. Varakin DA, Levin D and Fidler R. Unseen and unaware: implications of recent research on failures of visual awareness for human-computer interface design. *Hum Comput Interact* 2004; 19(4): 389–422.
12. Wang X. Intelligent multi-camera video surveillance: a review. *Pattern Recogn Lett* 2013; 34(1): 3–19.
13. Dee HM and Velastin SA. How close are we to solving the problem of automated visual surveillance? *Mach Vision Appl* 2007; 19(5–6): 329–343.
14. Kurzhals K, Höferlin M and Weiskopf D. Evaluation of attention-guiding video visualization. *Comput Graph Forum* 2013; 32(3pt1): 51–60.
15. Tullio J, Huang E and Wheatley D. Experience, adjustment, and engagement: the role of video in law enforcement. In: *Proceedings of the SIGCHI conference on human factors in computing system, CHI*, Atlanta, GA, 10–15 April 2010, pp. 1505–1514. New York: ACM.
16. Dadashi N, Stedmon AW and Pridmore TP. Semi-automated CCTV surveillance: the effects of system confidence, system accuracy and task complexity on operator vigilance, reliance and workload. *Appl Ergon* 2013; 44(5): 730–738.
17. Endsley MR. *Designing for situation awareness: an approach to user-centered design*. 2nd ed. Boca Raton, FL: CRC Press, 2011.
18. Keval H and Sasse M. Man or gorilla? Performance issues with CCTV technology in security control rooms. In: *Proceedings of the 16th world congress on ergonomics conference*, Maastricht, 10–14 July 2006.
19. Stedmon AW. The camera never lies, or does it? The dangers of taking CCTV surveillance at face value. *Surveill Soc* 2011; 8(4): 527–534.
20. Borgo R, Chen M, Daubney B, et al. A survey on video-based graphics and video visualization. In: *Proceedings of the Euro Graphics conference*, State of the Art Report, Llandudno, 11–15 April 2011, pp. 1–23. Geneve: The Eurographics Association.
21. Wang Y, Krum DM, Coelho EM, et al. Contextualized videos: combining videos with environment models to support situational understanding. *IEEE T Vis Comput Gr* 2007; 13(6): 1568–1575.
22. Girgensohn A, Kimber D and Vaughan J, et al. DOTS: support for effective video surveillance. In: *Proceedings of the ACM international conference on multimedia, MM'07*, Augsburg, 23–28 September 2007, pp. 423–432. New York: ACM.
23. Collins R, Lipton AJ, Fujiyoshi H, et al. Algorithms for cooperative multisensor surveillance. *P IEEE* 2001; 89(10): 1456–1477.
24. Girgensohn A, Shipman F, Turner T, et al. Effects of presenting geographic context. In: *Proceedings of the SIGCHI conference on human factors in computing systems, CHI*, San Jose, CA, 28 April–3 May 2007, pp. 1167–1176. New York: ACM.
25. McCrickard D, Catrambone R, Chewar C, et al. Establishing tradeoffs that leverage attention for utility: empirically evaluating information display in notification systems. *Int J Hum Comput St* 2003; 58: 547–582.
26. Lam H. A framework of interaction costs in information visualization. *IEEE T Vis Comput Gr* 2008; 14(6): 1149–1156.
27. Zhou X, Yang C and Yu W. Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE T Pattern Anal* 2013; 35(3): 597–610.
28. Wolfe JM and Horowitz TS. What attributes guide the deployment of visual attention and how do they do it? *Nat Rev Neurosci* 2004; 5(6): 495–501.
29. Kistner G, 2012, http://phrogz.net/css/distinct-colors.html
30. Treisman A and Gormican S. Feature analysis in early vision: evidence from search asymmetries. *Psychol Rev* 1988; 95(1): 15–48.
31. Qureshi F and Terzopoulos D. Towards intelligent camera networks: a virtual vision approach. In: *Proceedings of the IEEE international workshop on VS-PETS*, Beijing, China, 15–16 October 2005, pp. 177–184. New York: IEEE.
32. Bustamante EA, Anderson BL and Bliss JP. Effects of varying the threshold of alarm systems and task complexity on human performance and perceived workload. *Proc Hum Factors Ergon Soc Annu Meet* 2004; 48(16): 1948–1952.
33. Sears CR and Pylyshyn ZW. Multiple object tracking and attentional processing. *Can J Exp Psychol* 2000; 54(1): 1–14.