Abnormal Scene Change Detection from a Moving Camera Using Bags of Patches and Spider-Web Map

Jun-Wei Hsieh, *Member, IEEE*, Chi-Hung Chuang, Salah Alghyaline, Hui-Fen Chiang, and Chao-Hong Chiang

***Abstract*—This paper proposes a novel surveillance system for detecting exceptional scene changes as abnormal events with a mobile camera mounted on a robot. In contrast to abnormal event analysis using fixed cameras, three key problems should be tackled in this system, *i.e.*, scene construction, robot localization, and scene comparison. For the first problem, "scene construction", a clustering scheme is proposed for extracting a set of key frames from the surveillance environment. Each key frame is further divided into a set of patches, which forms a sparse representation for representing scene contents. In addition to the compression effect, the scheme can tackle the effects of misalignment and lighting changes well. For the localization problem, a novel patch matching method is proposed to reduce not only the size of the search space but also the size of the feature dimensions in similarity matching. To prune the search space, a set of projection kernels is used to construct a ring structure. Then, one order of time complexity in the similarity calculation can be reduced from the structure. After scene searching, the robot location is not always guaranteed to be successfully registered to the scene map. Thus, a novel spider-web map is proposed to tackle the effect of misalignment and then detect different exceptional scene changes from the videos. The proposed method has been rigorously tested on a variety of videos to demonstrate its superiority in object detection and abnormal scene change detection.**

***Index Terms*—behavior analysis, abnormal scene change detection, pattern matching, video surveillance**

I. Introduction

Video surveillance [1]-[9] is the use of cameras to analyze different security events directly from videos. For example, the task of missing object detection can be used for security monitoring, crime detection, and anti-terrorist surveillance. In most surveillance systems, the camera should be fixed to the background so that foreground objects can be extracted by a subtraction technique. For example, in [1], Kim *et al.* built a codebook model to extract foreground objects directly from videos. Stringa [2] used a key frame extraction technique to locate moving objects and then recognize suspicious objects according to their geometric properties. In [3], Foresti *et al.* used a long-term change detection algorithm to detect abandoned objects and then classified video sequences into four dangerous events. In [4], Piciarelli, Micheloni, and Foresti proposed a SVM-based scheme to cluster object trajectories to different event types from a fixed camera. However, a fixed camera is not appropriate for monitoring a large-scale environment because of its limited field of view.

To develop a wide-area surveillance system, two key problems should be tackled, that is, scene construction and scene comparison. For the first problem, the wide-area scene can be constructed using a set of multiple cameras or a mobile camera. If a set of multiple cameras is used, the spatial and temporal relations between cameras should be built. For example, Zhao *et al.* [5] developed a multiple overlapping camera system that uses space-time information and a segmentation technique to handle the handoff problem between cameras and then to track pedestrians among different views. Dockstadert and Tekalp [7] used Bayesian networks to overcome the occlusion problem and then tracked multiple persons when they moved across a set of overlapping cameras. Compared with the case of overlapping cameras, it is more challenging to link the relations between non-overlapping cameras because there is less common information between them. In [6], Sheikh, Li, and Shah took advantage of object trajectories to establish the relations between non-overlapping cameras. Chilgunde *et al.*[8] integrated object trajectories, camera relations, and object shapes to track objects across different non-overlapping cameras. In [12], Makris and Ellis assumed a common ground plane and then estimated the relations between non-overlapping cameras from event correlations. In [45], Micheloni, Rinner, and Foresti used PTZ camera networks to detect and track moving objects such as pedestrians and vehicles. For this multiple-camera surveillance system, due to the spatial constraints and cost consideration, the placement of cameras will allow some regions to become unsafe and blind because it cannot guarantee that all of the surveillance environment will be well monitored.

To more aggressively monitor the blind regions, another approach is to build a wide-scale surveillance environment

This work was supported in part by National Science Council of Taiwan, Taiwan, R.O.C., under Grant NSC 100-2221-E-019-030-MY3, and in part by the Ministry of Economic Affairs under Contract 98-EC-17-A-02-S1-032.

Jun-Wei Hsieh, Salah Alghyaline, and Hui-Fen Chiang are with the Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan. (e-mail: shieh@ntou.edu.tw, salahshaman2007@gmail.com, chelly.chiang@gmail.com)

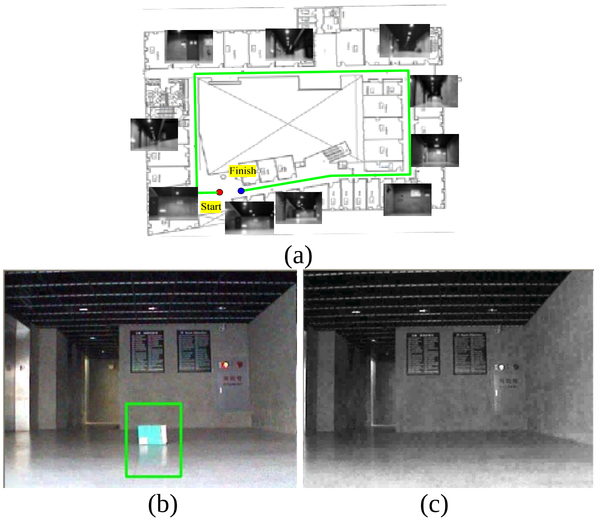
Chi-Hung Chuang is with the Department of Learning and Digital Technology, Fo Guang University, Yilan, Taiwan. (email: chchuang@mail.fgu.edu.tw)

Hui-Fen Chiang is also with the Department of Digital Multimedia Design, Taipei Chengshih University of Science and Technology, Taipei, Taiwan.

Chao-Hong Chiang are with the Department of E. E., Yuan Ze University, Chung-Li 320, Taiwan. (e-mails: s954616@mail.yzu.edu.tw)

using a mobile camera. Actually, in [11], Smith, Self, and Cheeseman addressed the problem of building a map of an environment from a sequence of landmark measurements that were obtained from a moving robot as a SLAM (Simultaneous Localization and Mapping) problem. In [13] and [14], it was decomposed by Thrun into two individual problems, *i.e.*, robot localization and landmark estimation, and then a probabilistic method was proposed to quickly estimate the robot paths and environment maps using the tracking technique of particle filters. To realize this problem, in [15] and [16], Medioni *et al*. built a UAV (Unmanned Airborne Vehicle) surveillance system to monitor urban areas by assuming that there is a common ground on which different scenes were stitched together for easy surveillance. In [18], Gandhi and Trivedi used Omni-directional vision sensors to build a mobile surveillance system for monitoring large-scale areas and then detecting interesting events such as moving vehicles and persons. In [37], Cao *et al*. proposed a pyramid sampling histogram of oriented gradients to train a SVM-based vehicle detector and then computed vehicle motion trajectories from airborne videos. Cornelis *et al*.[19] took advantage of a real-time structure-from-motion scheme and stereo data to reconstruct 3D city models from cameras mounted on a vehicle. A good scene construction algorithm must be as fast as possible and should result in a compact, memory-efficient scene model for future ease of scene searching and comparison.

After scene construction, another challenging problem in abnormal event analysis is scene comparison. This comparison task can be directly achieved by background subtraction [1], [40], motion comparison [20], or feature matching [21], [38], [39], [41]. For example, Pilet, Strecha, and Fua [40] modeled the background as a Gaussian mixture model whose parameters were estimated by the EM algorithm to compare scenes even under sudden illumination changes. Yu, Yuan, and Liu [20] proposed a sparse reconstruction cost over a normal dictionary for abnormal event analysis from a fixed camera. Li *et al.* [21] used a set of dense trajectories to model events and then trained a SVM-based classifier to classify activities into various classes. Ebrahimi and Mayol-Cuevas [38] used a dense set of corner correspondences to describe scenes and then proposed an adaptive sampling technique to compare their contents. In [39], Chen *et al*. proposed a landmark-based scheme to build different code books and then performed scene content comparisons from a mobile camera. Dragusu, Mihalache, and Solea [46] used edge contours to match objects on simple background from a robotic arm. However, when a moving camera is used, an additional task, "scene searching", is needed to find the best scene for scene comparison. To address this problem, in [23], Wu and Tsai used different circular-shaped landmarks to find all possible correspondences between omni-cameras and then derived their parameters to locate the robot positions. In [24] and [41], Jung and Sukhatme used an adaptive particle filter to find possible corner correspondences to compensate the ego-motion of camera so that possible moving objects could be detected through an EM algorithm. In [9], Castelnovi *et al*. proposed a color clustering technique to cluster scenes to different contents, and then extracted abandoned objects from a surveillance robot. To extract foreground objects from a moving camera, in [17], Cucchiara, Prati, and Vezzani used Markov random fields to model the consistencies between pixel motions. Two key issues in scene searching are the searching efficiency for the real-time requirement and the accuracy in object extraction and event analysis.



(a)
(b) (c)

Fig. 1 Scenario of the proposed system. (a) Surveillance map. (b) Abandoned object. (c) The best matched scene.

This paper presents a novel surveillance system that detects exceptional changes of scenes as abnormal events using a moving camera mounted on a robot. Fig. 1 shows the scenario of our proposed system. The observed scene is captured along a predefined path (such as the green line in Fig. 1(a)). Then, different abnormal scene changes (such as an abandoned object in Fig. 1(b)) will be detected along the path by scene matching and subtracting the best scene (such as Fig. 1(c)). Three challenges are addressed in this system, *i.e.*, scene construction, scene searching, and scene comparison. Two stages are included in the system to tackle these challenges, *i.e*., the training and detection stages. To construct the observed scene, a clustering scheme is applied at the training stage to extract a set of key frames from the surveillance environment. To reduce the effects of lighting changes, each key frame is further divided into a set of patches. Then, at the detection stage, the task of abnormal scene change detection will become a scene searching problem. To increase the searching efficiency, this paper presents a novel patch matching method to search for each desired scene in real time. The proposed searching scheme reduces not only the search space but also the size of the feature dimensions in similarity comparison. To prune the search space, this scheme uses a set of projection kernels (generated from an integral image) to construct a ring structure. One order of time complexity in the similarity calculation can be reduced. Then, a spider-web map is proposed for scene comparison, foreground object detection, and abnormal scene change detection, even if the robot position is not well registered to the environment map. Three main contributions are made in this work:

1) An efficient scheme using the concept of "bag of patches" is proposed to construct a wide-scale surveillance environment from a mobile camera.
2) A novel patch searching method is proposed to search the robot's localization very efficiently. One order of time

complexity in the similarity calculation is reduced.

3) A novel spider-web map is proposed to detect foreground objects and analyze abnormal scene changes from a moving camera. This method is efficient and robust even though the robot location is not well registered.

The remainder of the paper is organized as follows. In the next section, the definitions of our problem and the flowchart of our system are described. Then, Section III discusses the details of scene construction and classification. After that, the patch-based technique for scene searching is described in Section IV. Section V discusses the details of abnormal scene change detection using a novel spider web map. Section VI reports a variety of experimental results, and finally, a conclusion will be presented in Section VII.

II. Problem Definition and Flowchart

In the SLAM problem [13]-[14], [25]-[26], the robot builds a map of an environment and simultaneously uses the map to compute its location. Let x_t denote the position of the robot at time t and $X_{0:t} = \{x_0, x_1, ..., x_t\}$, the history of robot locations. In addition, let $\mathbb{M}$ denote the map represented by a set of landmarks or feature points. Then, given an initial position x_0, a set of observations $Z_{0:t} = \{z_1, z_2, ..., z_t\}$, and the history of control inputs $U_{0:t} = \{u_1, u_2, ..., u_t\}$, the SLAM problem attempts to find the optimal x_t and $\mathbb{M}$ such that

$$P(x_t, \mathbb{M} \mid Z_{0:t}, U_{0:t}, x_0) \quad (1)$$

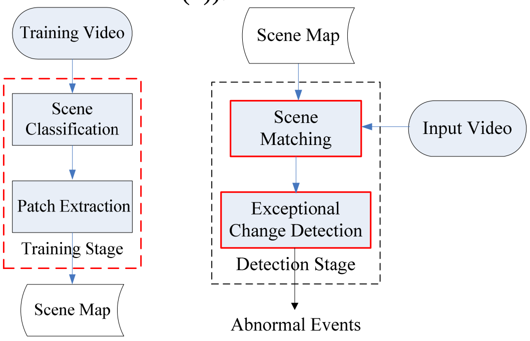
is maximized. The goal of this paper focuses on abnormal scene change detection with a mobile camera. To achieve this, the scene map should be built first before scene change analysis. Thus, in contrast to the original SLAM problem, which attempts to build the map $\mathbb{M}$ and estimate the location x_t at the same time, this paper factorizes Eq.(1) into two individual problems; that is, the map building problem and the location finding problem. The map building problem can be formulated as finding $\mathbb{M}$ to maximize the conditional density [25]:

$$P(\mathbb{M} \mid Z_{0:t}, U_{0:t}, x_0). \quad (2)$$

Conversely, the localization finding problem is to determine x_t so that the probability distribution is maximized:

$$P(x_t \mid Z_{0:t}, U_{0:t}, \mathbb{M}). \quad (3)$$

To tackle the map building problem, a training stage is first applied to extract a set of key frames and patches from the surveillance environment to construct $\mathbb{M}$. Based on $\mathbb{M}$, when a robot is moved with $U_{0:t}$, the optimal location x_t will be found from the observation $Z_{0:t}$ through a novel scene searching scheme. To detect abnormal scene changes from a moving robot, three challenges should be tackled. The first challenge is how to represent $\mathbb{M}$ more compactly because the size of $\mathbb{M}$ is huge. The second challenge is how to efficiently find x_t from $Z_{0:t}$ and $\mathbb{M}$ to meet the real-time requirement. The last challenge is how to detect abnormal scene changes from $\mathbb{M}$ if there is some misalignment between x_t and its exact position in $\mathbb{M}$. Fig. 2 shows the flowchart of our proposed abnormal scene change detection system to tackle the three challenges above. Fig. 2(a) shows the details of the training stage for map construction. Fig. 2(b) shows the details of the detection stage for scene searching and scene comparison. The problem of scene construction will be discussed in Section III. The upper red block (b) is used to consider the second challenge and will be discussed in Section IV. The details of abnormal scene change detection are described in Section V (the lower red block in (b)).



(a) (b)

Fig. 2 Flowchart of the system. (a) Training stage. (b) Detection stage.

III. Scene Construction through Classification

Assume the robot patrols the environment along a regular path. This section will propose a novel patch-based construction scheme to construct the map $\mathbb{M}$ along the path. In order to capture the scenes as can as possible, the camera view is vertical to the moving direction of the robot. In addition, at a corner site, the robot scans the environment by rotating the camera around its center to avoid the parallax effect. Fig. 3 shows the flowchart of the patch-based scheme. First, the scene is decomposed to a series of frames through a key frame selection technique (see Section III.A). Then, a patch-based representation scheme is applied to divide each frame into different patches (see Section III.B). To more compactly represent a scene, a stitching scheme is adopted to construct a scene panorama (discussed in Section III.C). Finally, to find the best scene, an efficient patch matching scheme is proposed and described in Section IV.

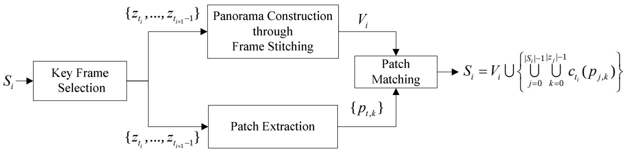

S_i
Key Frame Selection
$\{z_{t_i}, ..., z_{t_{i+1}-1}\}$
Panorama Construction through Frame Stitching
V_i
Patch Matching
$\{z_{t_i}, ..., z_{t_{i+1}-1}\}$
Patch Extraction
$\{p_{t,k}\}$
$S_i = V_i \cup \left\{ \bigcup_{j=0}^{|S_i|-1} \bigcup_{k=0}^{|z_j|-1} c_{t_i}(p_{j,k}) \right\}$

Fig. 3 Flowchart of the patch-based scene representation.

A. Key Frame Selection for Scene Construction

To construct the scene map $\mathbb{M}$ from a moving robot, a set of observation videos $Z_{0:t}$ is first collected along a predefined path (such as the green line in Fig. 1). At the training stage, the size of $Z_{0:t}$ is fixed and denoted by $|Z|$. Then, without loss of generality, we can rewrite $Z_{0:t}$ as Z, *i.e.*,

$$Z = \{z_1, z_2, ..., z_t, ..., z_{|Z|}\}.$$

Because the robot patrols the environment along a regular path, the initial location x_0 and the history of control inputs $U_{0:t}$ are known in advance. To construct $\mathbb{M}$ from Eq.(2), this paper represents $\mathbb{M}$ as a bag of scenes S_i, *i.e.*, $\mathbb{M} = \bigcup_i S_i$, where each scene S_i is further represented by a set of frames z_t, *i.e.*,

$$S_i = \bigcup_{t=t_i}^{t_{i+1}-1} z_t \, . \tag{4}$$

Each z_{t_i} is a key frame selected from Z by key frame selection. To obtain this set of key frames, the difference between two frames A and B should be measured first.

Let H_A and H_B denote the color histograms of A and B, respectively. Then, the chi-squared distance is used to measure their difference. It is derived from Pearson's chi squared test statistic and often used in computer vision to calculate the dissimilarity between some bag-of-visual-word representations of images [43]. Formally, it is defined as follows:

$$D_{his}\left(H_A, H_B\right) = \sum_{j=0}^{n_{bin}-1} \frac{2\left|H_A(j) - H_B(j)\right|^2}{H_A(j) + H_B(j)}, \tag{5}$$

where n_{bin} is the total number of color bins used. Suppose z_{t_i} is the *i*th key frame at the *i*th time t_i. A frame is identified as a key frame if the cumulative value of $D_{his}\left(H_{z_t}, H_{z_{t+1}}\right)$ is larger than a preset threshold T_{key}. T_{key} is a parameter fed into the key frame selection process. Before training, we manually select a set of pairs of key frames from possible surveillance sites at different captured conditions. Then, T_{key} is the average of the chi-squared distances between pairs of key frames in the selected set. T_{key} is fixed in the key frame extraction process for all surveillance scene maps. The value of the cumulative distance between two arbitrary frames, z_{t_A} and z_{t_B} (if $t_B > t_A$), can be calculated as follows:

$$Cdis\left(z_{t_A}, z_{t_B}\right) = \sum_{t=t_A}^{t_B-1} D_{his}\left(H_{z_t}, H_{z_{t+1}}\right). \tag{6}$$

For a video sequence to be characterized, we choose the first frame as a candidate key frame. Using the current key frame z_{t_i}, if the cumulative distance $Cdis\left(z_{t_i}, z_{t_i+\tau}\right)$ exceeds T_{key}, we choose the frame at that time instance as the next key frame k_{i+1}. If the time difference between z_{t_i} and $z_{t_{i+1}}$ is τ, we have $t_{i+1} = t_i + \tau$. After scanning all frames in Z, a set K of key frames can be extracted. However, there are still too many selected key frames in K to be accepted. Thus, we have to eliminate a set of redundant key frames from K.

To eliminate redundant key frames from K, we calculate the cross exponential entropy [30] to measure the dissimilarity between two key frames k_i and k_{i+1}, *i.e.*,

$$\begin{aligned} d_{crossE}(k_i, k_{i+1}) = &\sum_{j=0}^{n_{bin}-1} H_{k_i}(j) \exp(|H_{k_i}(j) - H_{k_{i+1}}(j)|) \\ &+ \sum_{j=0}^{n_{bin}-1} H_{k_{i+1}}(j) \exp(|H_{k_{i+1}}(j) - H_{k_i}(j)|). \end{aligned} \tag{7}$$

where H_{k_i} is the color histogram of k_i. Because $d_{crossE}(k_i, k_{i+1}) \geq 2$, we rewrite Eq.(7) as follows:

$$\begin{aligned} d_{crossE}(k_i, k_{i+1}) = \{ &\sum_{j=0}^{n_{bin}-1} (H_{k_i}(j) + H_{k_{i+1}}(j)) \\ &\exp(|H_{k_i}(j) - H_{k_{i+1}}(j)|)\} - 2. \end{aligned} \tag{8}$$

A set of redundant key postures will be eliminated when the degree of dissimilarity $d_{crossE}(k_i, k_{i+1})$ is smaller than a threshold T_{crossE}. T_{crossE} is set to twice the average of d_{crossE} between pairs of key frames in K.

B. Patch-based Representation

In Section III.A, a bag of frames bounded by two key frames z_{t_i} and $z_{t_{i+1}}$ is used to represent the *i*th scene S_i (see Eq.(4)). However, many redundant and repeated areas are also included to represent S_i. As shown in Fig. 4, there are large overlapping areas between the two adjacent frames. Then, huge memory space will be wasted to represent S_i. In addition, this scheme will lead to some difficulties in foreground object detection if the robot is not well aligned. In what follows, a patch-based method will be proposed to tackle these problems.

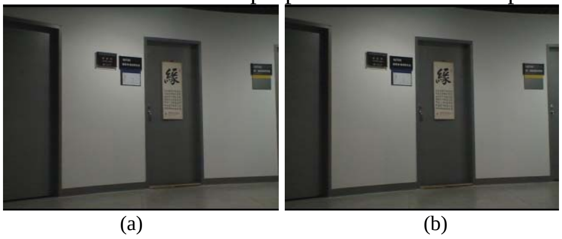


Fig. 4 Large overlapping areas found between two adjacent frames.

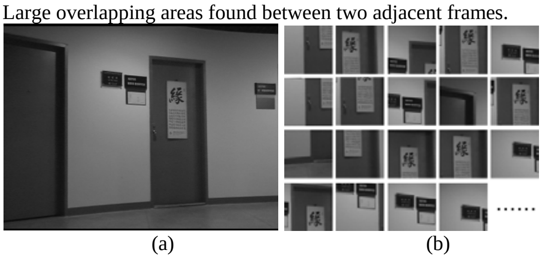


Fig. 5 Patch-based representation. (a) Input frame. (b) Different patches.

The patch-based method divides a frame into different patches. Before dividing, a feature extraction technique is first applied to z_t so that different point features $f_{t,k}$ can be detected, where $f_{t,k}$ denotes the *k*th feature point in the *t*th frame. The feature point can be extracted by a corner detector [10], SIFT[31], or SURF[32]. With $f_{t,k}$ as the central point, its corresponding patch $p_{t,k}$ will be extracted from z_t, where $p_{t,k}$ is an $m \times m$ sub-region extracted from z_t. Fig. 5 shows a frame divided to different patches. Fig. 5(a) is the input frame, and Fig. 5(b) shows a set of patches to represent (a), where m is set to 63. Then, z_t can be further represented as follows:

$$z_t = \left\{p_{t,k}\right\}_{k=0,\ldots,|z_t|-1}, \tag{9}$$

where $|z_t|$ denotes the number of patches in z_t. Substituting Eq.(9) to Eq.(4), we have

$$S_i = \bigcup_{t=t_i}^{t_{i+1}-1} z_t = \bigcup_{t=t_i}^{t_{i+1}-1} \left\{p_{t,k}\right\}_{k=0,\ldots,|z_t|-1}. \tag{10}$$

In fact, for a frame $z_j \in S_i$, large overlapping areas exist between z_j and z_{t_i}, where $z_j \neq z_{t_i}$. Thus, for a patch p in z_j, if its correspondence in z_{t_i} can be found, a smaller set of patches can be formed to represent S_i more compactly. For p, if its correspondence is not found in z_{t_i}, we denote it as $\bar{p}$. Then, the patches in z_j can be categorized to two disjointed sets P_j and $\bar{P}_j$, *i.e.*,

$$P_j = \left\{ p_{j,k} \right\}_{k=0,\dots,m-1} \text{ and } \bar{P}_j = \left\{ \bar{p}_{j,k} \right\}_{k=0,\dots,n-1}, \qquad (11)$$

where $n+m=|z_j|$ and $z_j = P_j \cup \bar{P}_j$. The method to find patch correspondences will be discussed later. Let $c_{t_i}(p_{j,k})$ denote the corresponding position of $p_{j,k}$ in z_{t_i}. The dimension of $c_{t_i}(p_{j,k})$ is 2 (the x and y coordinates), which is much smaller than the patch dimension $m \times m$. As shown in the flowchart in Fig. 3, a stitching algorithm will be adopted and discussed in Section III.C to stitch all z_j in S_i together to form a panorama view V_i. Then, the correspondence of each patch in z_j can be found from V_i. Thus, $\bar{P}_j$ will become an empty set. Then, z_j can be represented with the set of $\left\{ c_{t_i}(p_{j,k}) \right\}$ and the panorama V_i by:

$$z_j = V_i \cup \left\{ \bigcup_{k=0}^{|z_j|-1} c_{t_i}(p_{j,k}) \right\}. \qquad (12)$$

The union of z_j will form S_i as follows:

$$S_i = V_i \cup \left\{ \bigcup_{j=0}^{|S_i|-1} \bigcup_{k=0}^{|z_j|-1} c_{t_i}(p_{j,k}) \right\}. \qquad (13)$$

Once S_i is built, the union of S_i will form the map $\mathcal{M}$ for abnormal scene change detection, *i.e.*,

$$\mathcal{M} = \bigcup_i S_i. \qquad (14)$$

C. Panorama Scene for Abnormal Scene Change Detection

To build the panorama view V_i from all of the frames z_j in S_i, we use a perspective transform to model all possible global camera motions between any two adjacent frames. The relationship between two adjacent images can be defined as:

$$x' = \frac{m_0 x + m_1 y + m_2}{m_6 x + m_7 y + 1} \text{ and } y' = \frac{m_3 x + m_4 y + m_5}{m_6 x + m_7 y + 1}, \qquad (15)$$

where (x, y) is the coordinate of a pixel in the current frame, (x', y') is the coordinate of its corresponding point in the next frame z_{j+1}, and $M = (m_0, m_1, \dots, m_7)$ are the parameters associated with the focal length, rotation angle, and scaling of the camera, respectively. Then, this set of parameters in Eq.(15) will become linear if four pairs of correct matches are found. If more than four pairs of marches (even incorrect) are found, M can be estimated by voting and optimization estimation [27]. Through integration, the panorama view V_i can be constructed.

IV. Patch Matching

This section will propose a new real-time patch matching method for efficient scene searching and scene comparison.

A. Concentric Feature

Assume a 2D $m \times m$ patch $p(x,y)$ is to be matched within an input image $I(x,y)$ of size $n \times n$. Then, the time complexity to find p from I will be $O(m^2 n^2 s)$, where s is the dimension of the scaling space. If there are some rotation changes, the complexity will become $O(m^2 n^2 As)$, where A is the dimension of the rotation space. Clearly, the task of patch searching is very time-consuming. This section presents a novel method that reduces not only the search space but also the feature space in similarity matching. To prune the search space, a ring structure is proposed to filter out most of impossible candidates in advance. The ring structure is constructed by a set of projection kernels (using an integral image), which forms a series of weak hypotheses to filter out impossible candidates. With a pyramid structure and a cascade architecture, each desired patch can be well and efficiently located even though they have scaling changes. Because the kernel function is small and independent of patch size, the searching task can be very efficiently performed. Actually, the proposed method can reduce one order of time complexity in the similarity calculation.

The flowchart of our system for patch matching is shown in Fig. 6. Two major stages are included in our method. At the first stage, a set of kernel filters is extracted from the ring structure to eliminate all impossible candidates. At the second stage, from a down-scaled searching structure, the locality filter is used to examine local structures of the remaining patches and to locate them more accurately through a cascade structure.

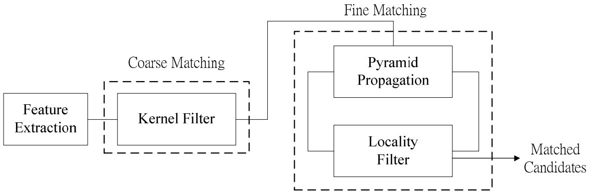


Fig. 6 Flowchart for patch matching.

Assume a two-dimensional $m \times m$ patch $p(x,y)$ is to be matched within an input image $I(x, y)$. For the patch p and one pixel c in $I(x,y)$, we can measure their distance as follows:

$$d^2(c,p) = \sum_{i=-m/2}^{m/2} \sum_{j=-m/2}^{m/2} \left(I(x_c + i, y_c + j) - p(i,j) \right)^2, \qquad (16)$$

where the coordinates of c are (x_c, y_c), and $p(i, j)$ and $I(i, j)$ denote the gray values of pixel (i, j) in the patch p and image I, respectively. Clearly, the time complexity of calculating $d^2(c, p)$ is $O(m^2)$. The time complexity can be reduced to $O(m)$ if a kernel function is used based on the concept of "integral image" to avoid many redundant calculations. Given a point c in $I(x, y)$, we define its kernel-based sampling function with the radius ρ as

$$k(c,\rho) = \frac{1}{\|R_\rho\|} \sum_{p \in R_\rho} E(p), \qquad (17)$$

where $R_\rho = \{p \mid \|p-c\| < \rho\}$ and E is the gray map of I or its edge map, *i.e.*, the set of gradient magnitudes of I obtained by the Sobel or Canny edge operators. Because edge feature is more robust than intensity to overcome the problem of lighting changes[44], the edge map is chosen in our real implementation. The kernel-based sampling function forms the basic statistical measurement in our task for patch matching. When different radiuses ρ are used, a new descriptor can be defined to describe the visual features of c, *i.e.*, $k(c, 1)$, $k(c, 2)$,…,$k(c, m)$. Then, a vector $k(c)$ with m elements can be defined to represent c, *i.e.*, $k(c)=(k(c, 1), k(c, 2), …,k(c, m))$. For the patch p, we also use a similar idea to obtain its descriptor $k(p)$. With the descriptors $k(p)$ and $k(c)$, the distance between c and p can be redefined as

$$d_k^2(c,p) = \sum_{i=1}^{m} w_i \left(k(c,i) - k(p,i)\right)^2, \qquad (18)$$

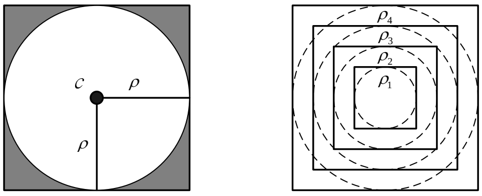
where w_i is a weight to weight the term $k(c, i)$. For a pattern, if its inner structure is more important than its outer structure, the value of w_i decreases according to i and vice versa. In our implementation, w_i is equally set. The time complexity for calculating $d_k^2(c,p)$ is $O(m)$, which is one order lower than the similarity calculation in Eq.(16).

B. Circular Sampling Function

To obtain $k(c)$, we use a circular kernel sampling function to calculate each term $k(c, \rho)$, as shown in Fig. 7(a). Technically, the circular kernel sampling function can be approximated with a square kernel sampling function $\pi(c,\rho)$ defined as

$$\pi(c,\rho) = \frac{1}{\rho^2} \sum_{i=-\rho/2}^{\rho/2} \sum_{j=-\rho/2}^{\rho/2} E(x_c + i, y_c + j). \qquad (19)$$

Fig. 7(b) shows the structure $\pi(c, \rho)$ as an approximation of $k(c, \rho)$. Because $\pi(c, \rho)$ can be very efficiently summed up by an integral image, we use $\pi(c)$ to approximate $k(c)$ for matching.



(a) (b)

Fig. 7 (a) A circular sampling function. (b) Its squared approximation centered at location *c*.

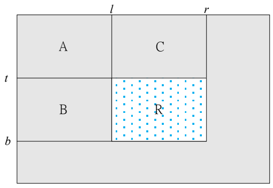


Fig. 8 Calculation of integral image.

Given a feature map E, its integral image $S(x,y)$ contains the sum of feature points in E accumulated from the original (0,0) to the pixel (x,y), *i.e.*,

$$S(x,y) = \sum_{i=0}^{x} \sum_{j=0}^{y} E(i,j).$$

It can be computed recursively as

$$S(x,y) = S(x,y\text{-}1) + S(x\text{-}1,y) + E(x,y) - S(x\text{-}1,y\text{-}1),$$

with the boundary conditions $S(-1,y)=S(x,-1)=S(-1,-1)=0$. Clearly, the computation of $S(x, y)$ can be completed using only one scan over E. Given a rectangle region R bounded by (l,t,r,b), its sum of edge magnitudes can be very efficiently achieved by S. As shown in Fig. 8, $sum(R)$ can be easily calculated using only one addition and two subtractions as follows:

$$\begin{aligned} sum(R) &= (A+B+C+R) + A - (A+B) - (A+C) \\ &= S(r,b) + S(l,t) - S(l,b) - S(r,t). \end{aligned} \qquad (20)$$

In real cases, $\pi(c,\rho)$ can be calculated similar to Eq.(20). Because the integral image can be calculated before pattern matching, the time to obtain $\pi(c,\rho)$ is independent of ρ and with the time complexity $O(1)$.

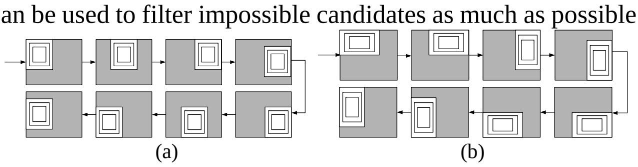
C. Ring Structure with Layers

We have presented a concentric feature to represent a point c so that the similarity between c and the patch p can be quickly calculated. To more accurately match p, a ring structure with various layers will be proposed here to compare its inner localities. As shown in Fig. 7 (b), with different ρ, $\pi(c,\rho)$ can form a ring structure, which is a collection of local concentric features to record the local structure of p. With different $\pi(c,\rho)$, we can perform a coarse-to-fine filtering process to quickly find the final best candidates of p.

In Eq.(19), if we permit the point c to have different shifts, different ring structures can be generated. Let $l_i = m \cdot 2^{-(i+1)}$ and $h_i = \{(\pm l_i, 0), (0, \pm l_i), (\pm l_i, \pm l_i)\}$, where $m\times m$ is the dimension of p. In addition, let C_s^i denote the set of all shifted points of c at the ith layer. C_s^i is generated from its previous layer C_s^{i-1} with the recursive form $C_s^i = \{C_s^{i-1} + b \mid b \in h_i\}$. Initially, C_s^0 includes only the element c, *i.e.*, $C_s^0 = \{c\}$. Clearly, C_s^1 owns eight elements, and C_s^2 has 64 elements. At the ith layer, its corresponding ring structure $\pi_i(c)$ is generated by collecting all elements c_j^i in C_s^i as follows:

$$\pi_i(c) = \bigcup_{c_j^i \in C_s^i} \pi(c_j^i),$$

where $\pi(c_j^i) = (\pi(c_j^i, l_i), \pi(c_j^i, l_i + 1), ..., \pi(c_j^i, l_{i-1}))$. The eight ring structures of $\pi_1(c)$ are listed in Fig. 9 (a). If two adjacent points in C_s^i are integrated, another type of ring structure can be generated (see Fig. 9 (b)). Then, the set of ring structures can be used to filter impossible candidates as much as possible.



(a) (b)

Fig. 9 Different types of ring structure with concentric integral sampling.

To consider the scaling change and improve the matching efficiency, a pyramid structure is constructed to divide the original image into different layers, as shown in Fig. 10. For each layer, the sampling rate is 0.95 for the x and y coordinates. Then, from the coarse layer, we can gradually find the desired patch locations until the fine layer is reached.

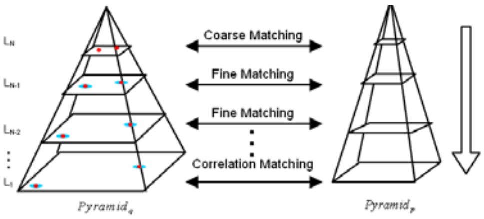


Fig. 10 Representation of a pyramid structure.

D. Cascade Structure

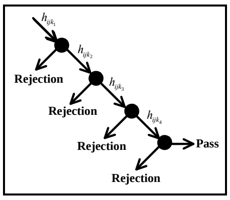
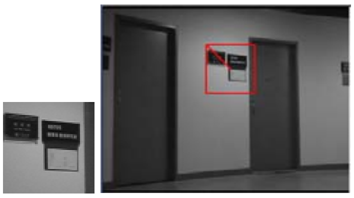


Fig. 11 Cascade structure for patch matching.

Actually, each concentric feature $\pi\left(c_j^i,k\right)$ extracted from its ring structure $\pi\left(c_j^i\right)$ can form a weak hypothesis h_{ijk} for quickly discarding impossible candidates while spending more computation on promising regions. With the set of concentric features h_{ijk}, a cascade structure can be easily constructed to find each desired patch in real time. The cascade structure is shown in Fig. 11. First, we use the concentric feature $\pi(c,k)$ to quickly filter out impossible patch candidates. Then, all of the remained candidates are further verified using the local concentric features $\pi\left(c_j^i,k\right)$ at different layers for $i > 0$. Then, the optimal location of the target can be detected from the input image. Fig. 12 shows a result of patch matching using our proposed method.



(a) (b)

Fig. 12 (a) Input patch. (b) Result of patch matching.

For a feature used in our method, we primarily consider its robustness for surviving under different lighting conditions and its low complexity for efficient matching. The pixel intensity can be directly used in our scheme for similarity comparisons. However, pixel intensity is not robust when images include illumination changes [44]. Thus, we extract edge features from image differential spaces for patch matching because they are more suitable for overcoming lighting changes.

V. Abnormal Scene Change Detection

After scene construction, the following task is to detect abnormal scene changes from the environment map $\mathbb{M}$. To analyze abnormal scene changes from $\mathbb{M}$, two problems should be tackled. First, when the robot moves to a position, its corresponding scene should be found very quickly. Second, the input frame will not be well aligned to the found scene. Thus, large subtraction errors will occur and lead to the failure of abnormal scene change detection. In what follows, a novel scene change detection scheme will be proposed to tackle the problems above.

A. Scene Searching

Given the current frame I, for abnormal scene change detection, we must search its corresponding scene quickly from $\mathbb{M}$. Because I occupies only parts of its corresponding scene, its global features, such as its color histogram, cannot be directly used for scene searching. Thus, we divide I into $L\times L$ grids b_l^I to define the similarity between I and a scene S_i. As shown in Fig. 13, I is divided into 10×10 grids. Then, the matching score between I and S_i is defined as

$$Score(I,S_i)=\sum_{l=1}^{L^2} s_{i,l}^I\ , \tag{21}$$

where

$$s_{i,l}^I=\begin{cases}1, & \text{if one of feature points in } S_i \text{ is matched within } b_l^I;\\ 0, & \text{otherwise.}\end{cases}$$

Then, the best scene S_{best} will be returned using the equation

$$S_{best}=\arg\max_i Score(I,S_i)\,. \tag{22}$$

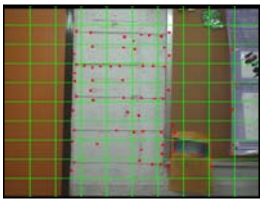


Fig. 13: Dividing the input frame into several grids for scene searching.

To enhance the efficiency and accuracy of scene searching, the transition probability between two scenes should be included. Let $T(i,j)$ denote the transition probability from the ith scene S_i to the jth scene S_j, defined as follows:

$$T(i,j)=\exp(-\frac{(j-i)^2}{\sigma_t^2})\,, \tag{23}$$

where σ_t is the variance of $|i-j|$. Let S_{best}^t denote the best scene obtained at time t and $Id(S_{best}^t)$ be its index. Then, the best scene S_{best}^t can be sought with the following equation:

$$S_{best}^t=\arg\max_j T(Id(S_{best}^{t-1}),j)Score(I,S_j)\,. \tag{24}$$

In real implementation, one scene cannot be searched far from its adjacent scenes. Thus, a spatial constraint is needed and added to Eq.(24) to reduce the search space. Only the scenes close to S_{best}^{t-1} will be searched to obtain the current best scene S_{best}^t, *i.e.*,

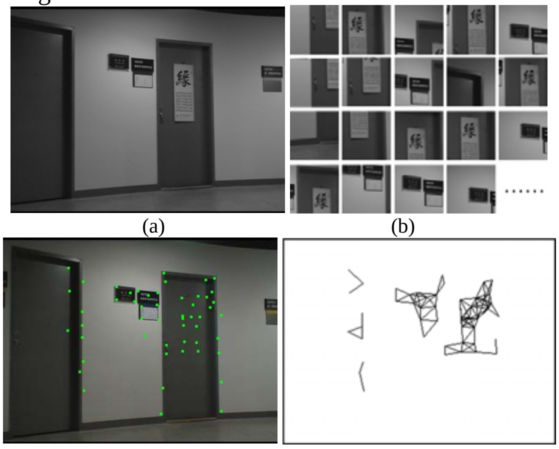
$$S_{best}^t=\arg\max_{|Id(S_{best}^{t-1})-j|\le 2} T(Id(S_{best}^{t-1}),j)Score(I,S_j)\,. \tag{25}$$

Once S_{best}^t is obtained, it is ready to detect different abnormal scene changes from $\mathbb{M}$ using a novel spider-web map.

B. Spider-web-like Map for Abnormal Scene Change Detection

After scene searching, the commonly adopted approach for scene comparison is background subtraction. Because an efficient matching scheme has been proposed in Section IV, we can search the best searched scene with the highest matching cost and then threshold it for abnormal scene change detection. However, when I is not well registered, large subtraction error will still occur and lead to the failure of scene comparison. To tackle the problem, a novel spider-web-like map is proposed to analyze various abnormal scene changes between I and S^t_{best}.

Let $\Omega_{t,best}$ and Ω_I denote the sets of patches extracted from S^t_{best} and I, respectively. In addition, we use $\Omega^I_{t,best}$ to denote the set of correspondences of $\Omega_{t,best}$ on I after position projection. For all elements in $\Omega^I_{t,best}$, we can connect their centers to construct a spider-web-like map. Different scenes will form different spider-web-like maps. An object addition or removal operation will lead to the changes of this map that can be used for scene change detection. In this scheme, for a point p in $\Omega^I_{t,best}$, we select its two closest points from $\Omega^I_{t,best}$ to form a triangulation mesh. The points within a larger mesh seldom belong to the same object. Thus, if the distances between p and its adjacent points are not sufficiently small, the triangulation mesh will not be formed.


(a) (b)

(c) (d)

Fig. 14 (a) Part of the best searched scene S^t_{best}. (b) Set of patches found from S^t_{best}. (c) Patch centers of (b) projected onto (a). (d) Result of triangulation.

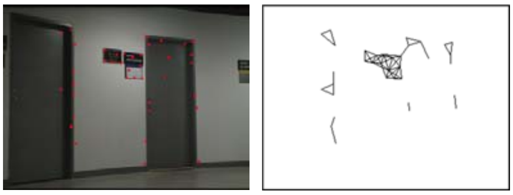

(a) (b)

Fig. 15. Triangulation result of input frame. (a) Input frame *I*. Each patch center of *I* is denoted by a red color. (b) Result of triangulation.

Let $SLD(p)$ denote the second lowest distance between p and all other points in $\Omega^I_{t,best}$ and $T_{SLD}(\Omega)$ be the average value of $SLD(p)$ for all points p in Ω, *i.e.*,

$$T_{SLD}(\Omega)=\frac{1}{|\Omega|}\sum_{p\in\Omega}SLD(p), \qquad (26)$$

where $|\Omega|$ is the number of points in Ω. For a pair of points p and q in $\Omega^I_{t,best}$, if their Euclidean distance $d_E(p,q)$ is larger than $T_{SLD}(\Omega^I_{t,best})$, a connection line between them will not be formed. Then, after browsing all pairs of points in $\Omega^I_{t,best}$, a spider-web-like map can be constructed. Fig. 14(a) shows part of the best searched scene $\Omega_{t,best}$. Some of patches in $\Omega_{t,best}$ are shown in Fig. 14(b). After position projection, their correspondences onto *I* will be found and collected as the set $\Omega^I_{t,best}$. Fig. 14(c) shows the projection result of each center of patches in $\Omega^I_{t,best}$ on I (with a green color), and Fig. 14(d) is the result by connecting the patch centers in $\Omega^I_{t,best}$. Similarly, the technique is also applied to Ω_I to obtain another spider web map. In Fig. 15(a), the centers of patches in Ω_I are denoted by red points. Fig. 15(b) is the result of connection to these centers.

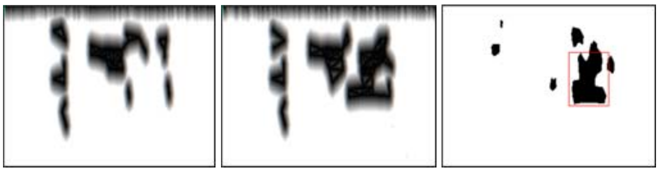

(a) (b) (c)

Fig. 16: Result of exceptional change detection. (a) Distance map of Fig. 15(b). (b) Distance map of Fig. 14(a). (c) Result of abnormal object detection.

Let W_I and $W^I_{t,best}$ denote the spider-web maps of Ω_I and $\Omega^I_{t,best}$, respectively. In real cases, there are many pixels with zeros in W_I and $W^I_{t,best}$. To detect abnormal objects in I, we need to transform W_I and $W^I_{t,best}$ to different distance maps. Let W denote one spider web map and DT_W be its distance map. Then, the value of a pixel p in DT_W is its shortest distance to all edge pixels in W, *i.e.*,

$$DT_W(p)=\min_{q\in edges\ of\ W} d(p,q), \qquad (27)$$

where $d(p, q)$ is the Euclidian distance between p and q. Before calculating DT_W, W is normalized to a unit size, and its center is the origin. Let DT_{W_I} and $DT_{W^I_{t,best}}$ denote the spider webs W_I and $W^I_{t,best}$, respectively. Then, the abnormal changes between W_I and $W^I_{t,best}$ can be detected through subtraction as

$$Diff_{W_I,W^I_{t,best}}(p)=\begin{cases}1, \text{ if } |DT_{W_I}(p)-DT_{W^I_{t,best}}(p)| > \lambda,\\ 0,\end{cases} \qquad (28)$$

where λ is the mean value of $|DT_{W_I}(p)-DT_{W^I_{t,best}}(p)|$. The region in $Diff_{W_I,W^I_{t,best}}$ with significant changes corresponds to an abnormal scene change happening. Through a connected component analysis [44], this region can be extracted for abnormal object (or scene change) detection. Fig. 16 shows the result of exceptional scene change detection. Fig. 16(a) and (b) are the distance maps of the spider-web maps of Fig. 15(b) and Fig. 14(d), respectively. Fig. 16(c) is the result using Eq.(28). The region bounded by a red rectangle (detected by a connected component analysis) is the abnormal object. Details of the spider-web-like map for abnormal object detection are described as follows.

Algorithm for Foreground Object Detection Using Spider-web-like Map

```
Input: Two sets of patches, i.e., $\Omega_I$ and $\Omega^I_{t,best}$;
Output: Foreground object $O$;
S1: Build $W_I$ from $\Omega_I$ using the following step:
    For each pair of points $p$ and $q$ in $\Omega_I$,
        Build the connection line between $p$ and $q$ if
            $d_E(p,q) < T_{SLD}(\Omega_I)$.
S2: Build $W^I_{t,best}$ from $DT_{W^I_{t,best}}$ using the following step:
    For each pair of points $p$ and $q$ in $DT_{W^I_{t,best}}$,
        Build the connection line between $p$ and $q$ if
            $d_E(p,q) < T_{SLD}(DT_{W^I_{t,best}})$.
S3: Obtain $DT_{W_I}$ and $DT_{W^I_{t,best}}$ through Eq.(27).
S4: Obtain $Diff_{W_I,W^I_{t,best}}$ using Eq.(28).
S5: Return the largest region $O$ from $Diff_{W_I,W^I_{t,best}}$.
```

VI. Experimental Results

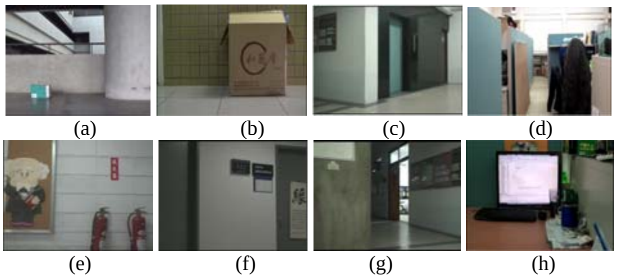

(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 17: Eight sites used for exceptional change analysis.

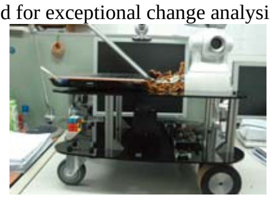


Fig. 18: The robot used for capturing videos for exceptional change analysis.

To analyze the performance of our approach, an intelligent video surveillance system that allows a robot to detect and analyze abnormal scene changes was implemented. Because there is no commonly used video database to benchmark different surveillance methods, a database containing 240 videos from eight sites was generated. Each site includes 30 video sequences that were captured under different lighting conditions, different starting points, and moving directions. The frame dimension of each video is 320×240. Fig. 17 shows the snapshots of the eight sites used in this database. Fig. 18 shows the robot we used for navigation and abnormal scene change detection. Three different abnormal scene changes were analyzed in this paper, *i.e.*, door-open, abandoned object, and object-lost. The platform used was a general laptop with the Intel Core Dual CPU L2400 1.66GHz and 2G RAM.

To construct the surveillance environment, this paper presented a patch-based scheme to build the scene map $\mathbb{M}$. The first set of experiments was performed to evaluate the performances of patch matching. Fig. 19 shows the results of patch matching when a traffic sign and a face pattern were given. There is some rotation change between (b) and (d). However, both of these cases were detected well using our method. Fig. 20 shows the result of patch searching for industrial component inspection. Fig. 20(a) is the inspection pattern, and Fig. 20(b) is the result of matching. Even though the orientations of the desired patterns in (b) are very different than those in (a), our method still works very well to detect and locate them. Table I lists the efficiency comparisons between traditional block matching schemes and our proposed method, where the time unit is *ms*. Our proposed method can reduce one order of time complexity in similarity calculation over the traditional block matching schemes [36]. The fourth row shows the improvement ratio of matching efficiency between our method and the traditional matching scheme. When the block size is larger, the improvement is more significant.

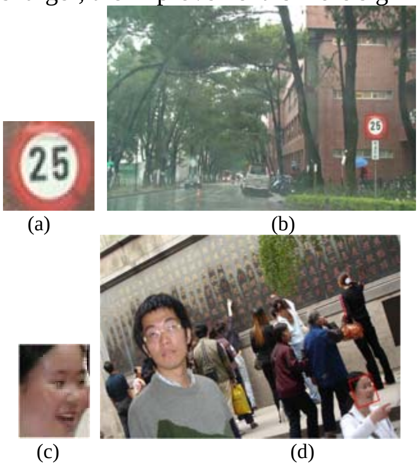

(a) (b)
(c) (d)

Fig. 19: Results of patch matching. (a) and (c): Input patches. (b) and (d): Results of matching.

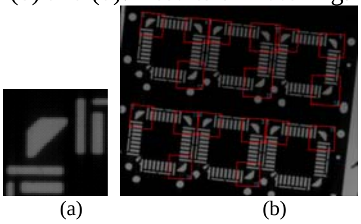

(a) (b)

Fig. 20: Result of patch matching. (a) Input patch with the dimension 73×54. (b) Input image with the dimension 717×578.

TABLE I
EFFICIENCY COMPARISONS BETWEEN TRADITIONAL BLOCK MATCHING AND CONCENTRIC BLOCK MATCHING UNDER DIFFERENT PATCH SIZES.

Patch size Methods	7×7	15×15	31×31	63×63	127×127
Traditional (ms)	155	462	1511	4503	16012
Concentric (ms)	60	61	62	109	203
Improvement Ratio	2.58	7.57	24.37	41.31	78.88

TABLE II
NUMBERS OF SCENES EXTRACTED FROM DIFFERENT SITES

Sites No. of Scenes	S1	S2	S3	S4	S5	S6	S7	S8	Avg.
With panorama	48	61	49	57	42	54	47	53	51.4
No panorama	183	231	175	232	154	225	164	215	197.4

To examine the performances of scene searching, a database including 240 videos were collected from eight surveillance sites (see Fig. 17). Each video was divided into several scenes, and each scene was bounded by two key frames. Table II lists the numbers of scenes extracted from these sites. The average number of scenes used to represent a map is 51.4. It depends on the complexity of the site content. The third row shows the number of scenes for map representation without a scene panorama. The average number of scenes for map representation without a scene panorama is approximately 197.4. Clearly, with a scene panorama, the number of used scenes is significantly decreased. The number of scenes used to represent a site will not affect the efficiency of scene change detection. Because the robot moves along a predefined path, only the scenes close to the current scene should be analyzed (see Eq.(25)). Actually, the time spent for scene searching strongly depends on the number of patches rather than the number of scenes. Table III lists the average number of patches used for scene representation at different sites. The number of patches is proportional to the number of detected corners. The dimension of each path is 63×63. Fig. 21 shows the snapshots of patch representations of Fig. 17(a), (e), and (g), respectively. Fig. 22 shows some scene panoramas obtained by the stitching method [27] for Fig. 17(c) and (e), respectively.

TABLE III
AVERAGE NUMBER OF PATCHES FOR EACH SCENE AT DIFFERENT SITES.

Sites / Patches per scene	S1	S2	S3	S4	S5	S6	S7	S8	Avg.
With panorama	76.5	79.8	66.3	60.2	90.4	98.8	96.1	61.9	78.75
No panorama	271.6	291.5	212.5	190.7	322.6	338.4	310.5	223.7	270.2

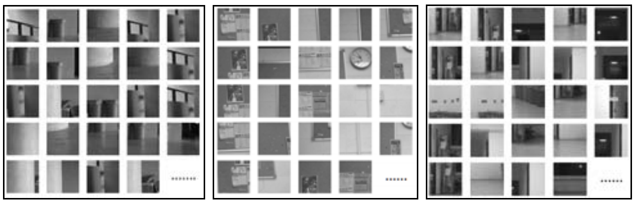

(a) (b) (c)

Fig. 21: Snapshots of patch representations for Fig. 17. (a) Snapshot of Fig. 17(a). (b) Snapshot of Fig. 17(e). (c) Snapshot of Fig. 17(g).

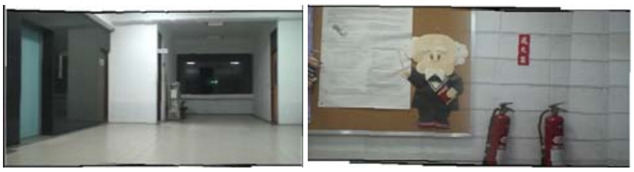

(a) (b)

Fig. 22: Stitching results of frames extracted from Fig. 17(c) and (e). (a) Panorama of Fig. 17(c) constructed from the 212*th* frame to the 272*th* frame. (b) Panorama of Fig. 17(e) constructed from the 98*th* frame to the 252*th* frame.

TABLE IV
EFFICIENCY COMPARISONS OF SCENE SEARCHING BETWEEN TRADITIONAL BLOCK MATCHING AND OUR PROPOSED METHOD WHEN DIFFERENT SCENES WERE HANDLED.

Scenes / Methods (ms)		S1	S2	S3	S4	S5	S6	S7	S8
Traditional block matching	Eq.(24)	19912	25827	18796	23534	15926	20754	19727	22582
	Eq.(25)	1242	1324	1212	1286	1209	1203	1189	1341
Concentric block matching	Eq.(24)	1856	2716	1831	2297	1574	1958	1684	2189
	Eq.(25)	121	138	117	126	116	114	112	128

Table IV shows the efficiency comparisons of scene searching between the traditional block matching method and our proposed method at different scenes. Clearly, our searching scheme is more efficient than the traditional block matching scheme [36]. Actually, the scene searching can be performed through Eq.(24) or Eq.(25). When Eq.(24) is used, all scenes should be screened. However, for Eq.(25), only some scenes located in the neighborhood of the current scene should be searched. Thus, Eq.(25) is more efficient than Eq. (24). Table V shows the accuracy comparisons when Eq.(24) and Eq.(25) were compared for scene searching. When Eq.(24) was adopted, because the map included too many similar scenes, many incorrect scenes were searched. Eq.(25) performs more accurately than Eq.(24) in scene searching because the spatial constraint in Eq.(25) will filter out most of the false candidates in advance. Table VI shows the accuracy comparisons with other searching features including color histogram, correlation, and concentric features. The color histogram will fail to handle the cases when the scenes contain many objects with similar colors. The accuracy of the correlation scheme is similar to that of the concentric features. However, our proposed method is significantly more efficient than the correlation scheme [36].

TABLE V
ACCURACY COMPARISONS BETWEEN EQ.(24) AND EQ.(25) ADOPTED FOR SCENE SEARCHING.

Accuracy (%)	Site 1	Site 2	Site 3	Site 4	Site 5	Site 6	Site 7	Site 8	Average
Using Eq.(24)	63.5	58.6	55.7	54.5	64.1	48.9	53.8	49.6	56.09
Using Eq.(25)	98.5	97.3	97.8	96.5	98.1	95.5	96.8	95.3	96.97

TABLE VI
ACCURACY COMPARISONS OF SCENE SEARCHING AMONG THE COLOR HISTOGRAM, CORRELATION, AND THE CONCENTRIC FEATURES.

Accuracy (%)	Site 1	Site 2	Site 3	Site 4	Site 5	Site 6	Site 7	Site 8	Average
Color Histogram	89.1	86.2	85.5	84.3	87.6	84.4	85.7	84.1	85.86
Correlation	94.5	95.6	96.9	93.1	94.4	91.2	93.4	92.1	93.9
Concentric features	98.5	97.3	97.8	96.5	98.1	95.5	96.8	95.3	96.97

The following experiment is used to evaluate the performance of our method to detect foreground objects and analyze abnormal scene changes. Fig. 23 shows the result of foreground object extraction using the spider-web map. Fig. 23(a) shows the input frame, and Fig. 23(b) is the cropped version of the best searched frame using (a). Fig. 23(c) and (d) are the spider-web maps of Fig. 23(a) and (b), respectively. Fig. 23(e) is the subtraction result using (a) and (b). In real cases, Fig. 23(a) is not always guaranteed to be well registered to (b). Thus, it is not surprising that there were many subtraction errors found in (e). Fig. 23(f) is the subtraction result between (c) and (d) using the spider web map. Only the region with the largest area is considered as the detected object (denoted by a green color in (a)). Fig. 24 shows another result of foreground object extraction from complicated backgrounds. Fig. 24(c) is the result of subtraction between (a) and (b). In (c), there were many subtraction errors found along object boundaries. Fig. 24(d) is the subtraction result using the spider-web map. Clearly, even though (a) and (b) were not well registered, the desired foreground object was correctly detected using the spider-web map. Table VII shows the accuracy comparisons of foreground object detection between the direct background subtraction scheme and our proposed spider-web map for all the sites collected in the dataset. Here, an object is detected "correctly" if the overlapping area between it and its ground-truth object is greater than 80%. Quite large subtraction errors were found from the direct subtraction scheme. This table proves that our method indeed performs more accurately than the direct method in foreground object extraction.

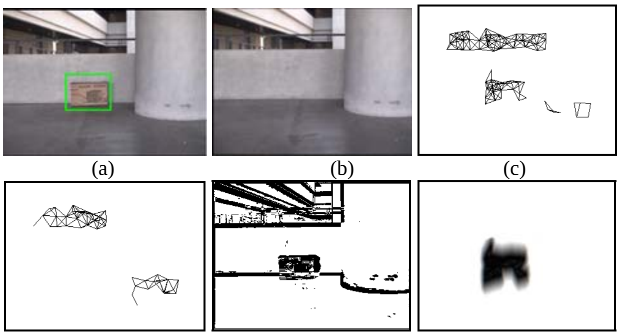

(a) (b) (c)
(d) (e) (f)

Fig. 23 Result of foreground extraction under simple background. (a) Input image (a green rectangle denoting the detection result). (b) Cropped version of the best frame. (c) Spider-web map of (a). (d) Spider-web map of (b). (e) Direct subtraction between (a) and (b). (f) Subtraction between (c) and (d) using the spider-web map.

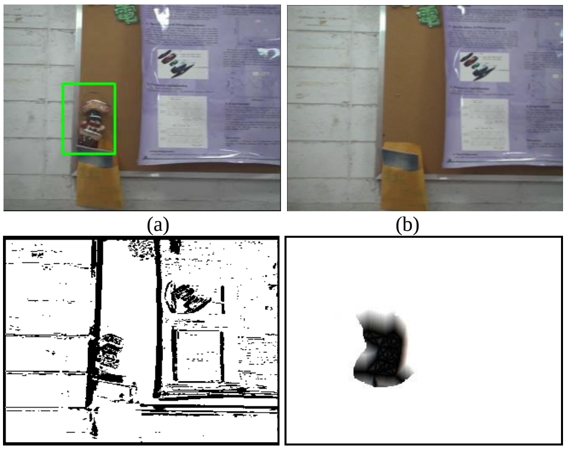

(a) (b)
(c) (d)

Fig. 24 Results of foreground extraction under complicated background. (a) Input image. (b) Cropped version of the best searched frame. (c) Result of background subtraction. (d) Subtraction result using the spider-web map.

TABLE VII
ACCURACY COMPARISONS OF FOREGROUND OBJECT EXTRACTION BETWEEN THE DIRECT SUBTRACTION AND THE SPIDER WEB MAP UNDER DIFFERENT CONDITIONS.

Conditions	Different Lighting	Simple Background	Complicated Background	Average
Direct background subtraction	62.34%	83.42%	77.37%	74.38%
Spider-web Scheme	83.79%	91.18%	89.67%	88.21%

Three abnormal scene change events were analyzed in this paper, *i.e.*, door-open, abandoned object, and object-lost. In some cases, the abnormal object might be a pedestrian and considered "normal". It can be further verified using a SVM-based or part-based classifier [33]-[34] to avoid this false detection. As shown in Fig. 25, different pedestrians were detected by a SVM-based pedestrian detector from a moving camera. Fig. 26 shows the case when a "door open" event and multiple pedestrians appeared together. Clearly, our proposed system can still detect them well (shown by different colors). To analyze more complicated abnormal events, different event analyzers can be designed using the detection results with HMMs [35] or sparse coding [21]. Fig. 27 shows some results of abnormal scene change detection when the background is simple. Fig. 27(a) is the input frame, and (b) is the cropped version of its best searched scene. Fig. 27(a) was not well registered to its background. However, the missed objects in (a) were still detected well. Fig. 28 shows another result of exceptional object detection when a textured object was handled. If the direct subtraction scheme is adopted, many errors will be found along the edges of this object. However, our method still detected the missing object well.

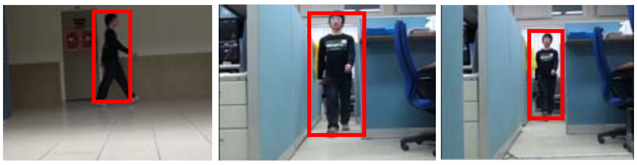

(a) (b) (c)

Fig. 25 Results of pedestrian detection through a moving camera.

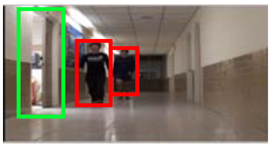


Fig. 26 Result of pedestrian detection when a "door-open" event and multiple pedestrians occurred together.

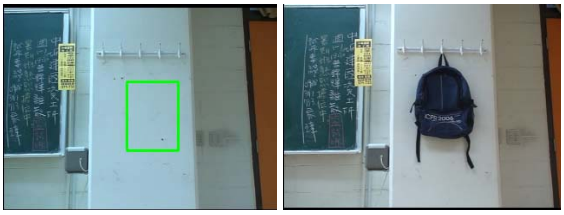

(a) (b)

Fig. 27 Results of exceptional change detection (shown by a green rectangle) with simple background. (a) Input image. (b) Cropped version of the best-searched scene.

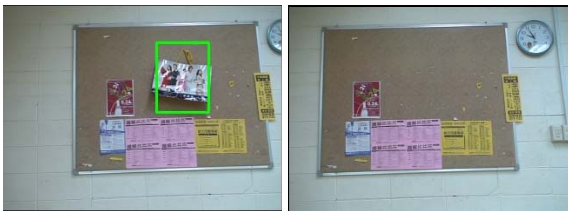

(a) (b)

Fig. 28 Results of exceptional change detection with texture patterns. (a) Input frame. (b) Cropped version of the best-searched scene.

Another challenging problem is to detect abnormal scene changes at a corner site. Fig. 29 shows the case at a corner site. Because of the lighting changes, the frame at a corner was not well registered to its desired scene. However, by using the concentric features and the spider-web map, the abandoned object was still extracted. Fig. 30 shows the detection result of the "door-open" event. Lighting is another factor that affects the accuracy of abnormal scene change detection. The next set

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2014.2381257, IEEE Sensors Journal

of experiments was used to examine the performance of abnormal scene change detection at night. Fig. 31 shows the result of missed object detection from a night scene. Fig. 31(a) is the input frame, and (b) is the cropped version of the best-searched scene. The detection result of this missed object is denoted by a green rectangle.

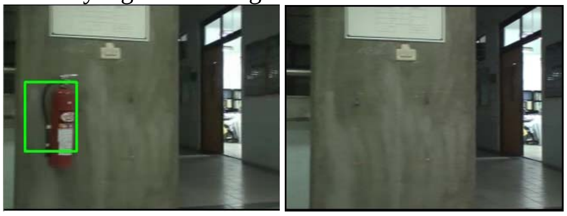

(a) (b)

Fig. 29 Results of exceptional change detection when the robot moved at a corner. (a) Input image. (b) Cropped version of the best-searched scene.

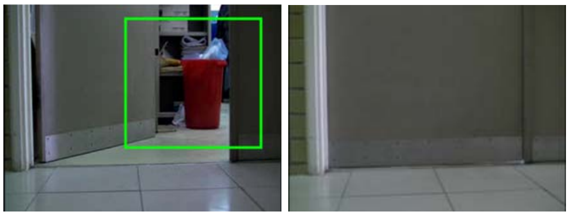

(a) (b)

Fig. 30 Detection result of "door-open" event. (a) Input image. (b) Cropped version of the best-searched scene.

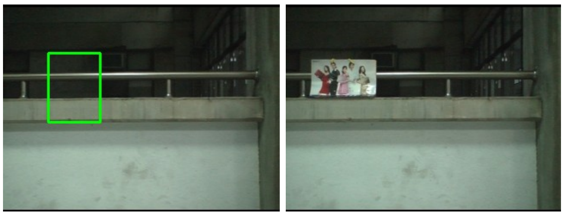
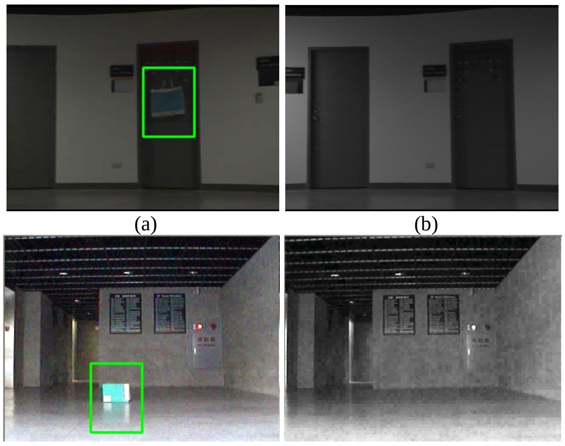

(a) (b)

Fig. 31 Results of exceptional change detection from a night scene.


(a) (b)

(c) (d)

Fig. 32 Results of exceptional change detection from a night scene.

Fig. 32 shows the cases of abandoned object detection at night. There were many similar scenes that appeared in the analyzed environment. However, benefiting from a transition probability (see Eq.(25)), the desired scene and abandoned object were still found well for the scene change detection. Another challenging problem is to detect abnormal scene changes caused by a smaller object. Fig. 33 shows the case of a green bag left on the floor at night. The scaling change of scenes between (a) and (b) is larger. With the spider-web map, the abandoned bag was still successfully detected. It is more challenging to detect abnormal scene changes from complicated scenes. Fig. 34 shows the result of a small abandoned object detected from a complicated scene. Various tables with strong edges appeared in the background. Even though the object is small and located behind a table, it is still successfully detected from this complicated scene by our method. Fig. 35 shows two failure cases of our proposed system. In (a) and (b), a false detection was found due to the poor lighting conditions. Fig. 35(a) is the input frame, and (b) is the best found scene. Because of poor lighting conditions, few features could be found in (b) for scene change detection. Thus, a false alarm was detected by the spider-web map. In the second failure case, there was a missed detection in (c) and (d). Fig. 35 (c) is the input frame, and (d) is the best found scene. Because the bag in (d) is too small and very close to the scene boundary, our method failed to detect it.

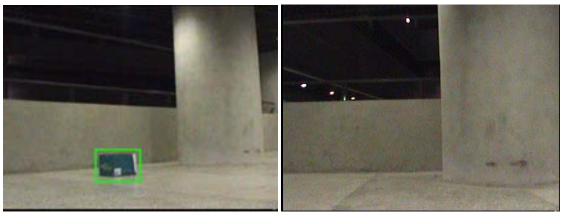

(a) (b)

Fig. 33 Detection result of a small abandoned object from a night scene.

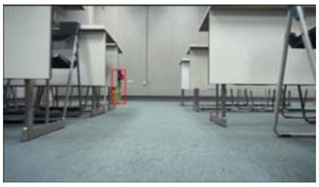
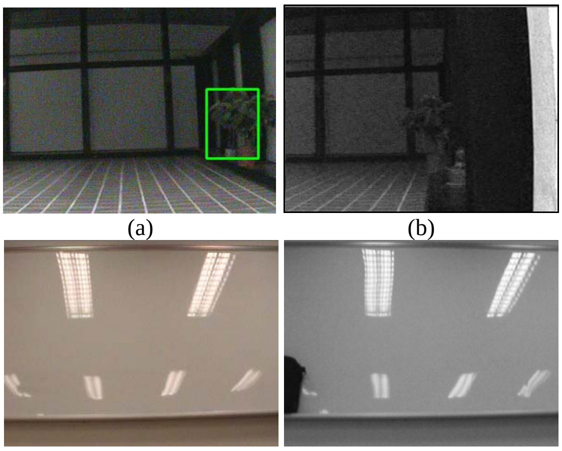


Fig. 34 Detection result of a small abandoned object from a complicated scene.


(a) (b)

(c) (d)

Fig. 35 Failure cases of our system. (a) and (b): failure case (false detection) due to poor lighting condition. (c) and (d): failure case (miss detection) due to object close to scene boundary.

For comparisons, the schemes proposed by Castelnovi *et al.*[28], Pilet *et al.* [40] and Jun *et al.*[41] were also implemented. The first one used the color histogram (extracted from the HSV color space) to describe a scene and then detected abnormal scene changes through a clustering technique. If a region does not belong to the color clusters recorded in the background, it will be classified as an abnormal object. The rule is not stable when the scene contains large lighting changes or the object colors are similar to the background. As shown in Fig. 36(a), the colors of the box are similar to the case colors (shown in (b)). Thus, the box object was not identified well using their method [28]. However, the box still was successfully identified by our spider-web map (see (a)). As to the second one, the EM algorithm is adopted to model the intensity ratios between background pixels and input pixels as the background to overcome the problem of sudden illumination changes. The method works well to deal with the problem of sudden illumination changes. However, if the robot ego-motion is well compensated, larger subtraction errors will be found along object boundaries (as shown in Fig. 5 of [41]). In the method of Jun *et al*.[41], the robot ego-motion is first estimated using dense corner correspondences between two consecutive images, and then different moving objects are detected by frame differencing. Thus, a laser rangefinder is needed but not adopted here for fair comparison. Because of the usage of frame differencing, an object is often doubly detected. Table VIII shows the accuracy comparisons among the three methods under simple backgrounds. The second, third, and fourth rows list the detection rate, false alarm rate, and miss rate of our method, respectively. The fifth, sixth, and seventh rows show the accuracies of scene change detection of [28], [40] and [41], respectively.

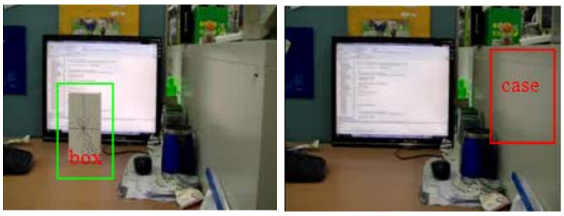
TABLE VIII
PERFORMANCE COMPARISONS OF ABNORMAL SCENE CHANGE DETECTION WITH SIMPLE BACKGROUNDS AMONG OUR METHOD, [28], [40] AND [41].

Event types	Abandoned	Missed	Door
Performances	Object	object	Open
Detection rate	81.01%	75.12 %	78.31%
False detection rate	3.31 %	3.36 %	3.23%
Miss rate	4.87%	4.73%	3.87%
Detection rate using [28]	71.32%	64.70%	68.54%
Detection rate using [40]	74.58%	70.97%	75.48%
Detection rate using [41]	72.78%	66.21%	73.35%

TABLE IX
PERFORMANCE COMPARISONS OF ABNORMAL SCENE CHANGE DETECTION WITH COMPLICATED BACKGROUNDS AMONG OUR METHOD, [28], [40] AND [41].

Event types	Abandoned	Missed	Door
Performances	Object	object	Open
Detection rate	87.10 %	94.55 %	85.79%
False detection rate	1.010 %	3.472 %	2.58%
Miss rate	4.43%	4.25%	3.76%
Detection rate using [28]	65.75%	61.71%	66.87%
Detection rate using [40]	72.57%	70.69%	73.86%
Detection rate using [41]	66.78%	65.86%	69.98%

Table IX shows the same comparisons but under complicated backgrounds. Actually, when simple backgrounds were handled, there were many similar scenes stored in the database, leading to the failure of abnormal scene change detection. Because fewer features were extracted from the simple background, the accuracy of our method under simple backgrounds is lower than that of the complicated backgrounds. For the method proposed in [28], it is more difficult to identify the color clusters of foreground objects in complicated backgrounds than simple backgrounds. Thus, in contrast to our method, the scheme in [28] performs worse under complicated backgrounds than simple backgrounds. For the techniques in [40] and [41], many subtraction errors were detected along object boundaries when complicated backgrounds were handled. Thus, their accuracies are worse under complicated backgrounds than simple backgrounds. The method of Pilet *et al*. [40] performs better than [41] because it works well when the scene contains sudden lighting changes. As to [41], it performs better when a large object is handled. Thus, the accuracy in the “Door-Open” event is higher than those of the “Abandoned object” and “Missed object” events. Because our proposed spider-web map has higher tolerance to environmental changes, it really performs better than the schemes proposed by Castelnopvi *et al*. [28], Pilet *et al*. [40], and Jun *et al*.[41] in both cases. All of the above experiments have proved the superiority of our method in abnormal scene change detection using a mobile camera.



(a) Input frame (b) Best frame in database.

Fig. 36 The clusters of the box color in (a) are similar to the case color in (b).

VII. CONCLUSIONS

This paper proposes a novel surveillance system for abnormal scene change detection from a mobile camera mounted on a robot using spider-web maps. Three key contributions were reported in this paper and are summarized as follows:

1) A patch-based method was proposed for scene construction. Then, a compact representation scheme can be formed for efficient scene searching and scene comparison.
2) A novel patch matching method was proposed to tackle the robot localization problem. It reduces not only the search space but also the feature space in patch matching. In addition, a ring structure was constructed to form a series of weak hypotheses, which can quickly discard impossible candidates. One order of time complexity in the similarity calculation is reduced.
3) A novel spider-web map was proposed for abnormal scene change detection, although the robot location was not well registered to the environment map.

Experimental results have demonstrated the superiority of our proposed system in exceptional change detection using a mobile robot under different background conditions.

REFERENCES

[1] K. Kim, et al., “Real-time foreground-background segmentation using codebook model,” Real-time Imaging, Vol. 11, Issue 3, pp. 172-185, June 2005.

[2] E. Stringa and C. S. Regazzoni, “Real-time video-shot detection for scene surveillance applications,” IEEE Transactions on Image Processing, vol. 9, no. 1, pp. 69-79, Jan. 2000.

[3] G. L. Foresti, L. Marcenaro, and C. S. Regazzoni, “Automatic detection and indexing of video-event shots for surveillance applications,” IEEE Transactions on Multimedia, vol. 4, no. 4, pp.459 – 471, Dec. 2002.

[4] C. Piciarelli, C. Micheloni, and G.L. Foresti, “Trajectory-based anomalous event detection,” IEEE Transaction on Circuits and Systems for Video Technology, vol. 18, no. 11, pp. 1544- 1554, Nov. 2008.

[5] T. Zhao, M. Aggarwal, R. Kumar, and H. Sawhney, "Real-time wide area multi-camera stereo tracking," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp.976-983, 2005.

[6] Y. Sheikh, X. Li, and M. Shah, "Trajectory association across non-overlapping moving cameras in planar scenes," IEEE Conf. of Computer Vision and Pattern Recognition, pp.1-7, 2007.

[7] S. L. Dockstadert and A. M. Tekalp, "Multiple camera fusion for multi-object tracking," IEEE Workshop on Multi-Object Tracking, pp395-102, 2001.

[8] A. Chilgunde, P. Kumar, S. Ranganath, and H. WeiMin, "multi-camera target tracking in blind regions of cameras with non-overlapping fields of view," British Machine Vision Conference, pp.397-406, 2004.

[9] M. Castelnovi, P. Musso, A. Sgorbissa, and R. Zaccaria, "Surveillance robotics: analyzing scenes by colors analysis and clustering," Proc. of IEEE International Symposium on Computational Intelligence in Robotics and Automation, Vol. 1, pp. 229-234, July 2003.

[10] C. G. Harris and M. J. Stephens, "A combined corner and edge detector," Proceedings Fourth Alvey Vision Conference, Manchester, pp. 147-151, 1988.

[11] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," In Autonomous Robot Vehicles, pp. 167-193, 1990.

[12] D. Makris and T. Ellis, "Learning semantic scene models from observing activity in visual surveillance," IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 35, no. 3, pp. 397–408, 2005.

[13] S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots," International Journal of Robotics Research, vol.20, no.5, pp.335-363, 2001.

[14] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: a factored solution to the simultaneous localization and mapping problem," In Proceedings of the National Conference on Artificial Intelligence, pp. 593–598, Edmonton, Canada, 2002.

[15] G. Medioni, I. Cohen, F. BreÂmond, S. Hongeng, and R. Nevatia, "Event detection and analysis from video streams," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 8, pp.873-889 Aug. 2001.

[16] Q. Yu and G. Medioni, "A GPU-based implementation of motion detection from a moving platform," IEEE Workshop on Computer Vision on GPU, 2008.

[17] R. Cucchiara, A. Prati, and R. Vezzani, "Real-time motion segmentation from moving cameras," Real-Time Imaging, vol.10, pp.127-143, 2004.

[18] T. Gandhi and M. M. Trivedi, "Motion analysis for event detection and tracking with a mobile omni-directional camera," ACM Multimedia Systems Journal, Special Issue on Video Surveillance, vol. 10, pp.131-143, 2004.

[19] N. Cornelis, B. Leibe, K. Cornelis, and L. V. Gool, "3D urban scene modeling integrating recognition and reconstruction," International Journal of Computer Vision, vol.78, pp.121-141, 2008.

[20] G. Yu, J. Yuan, and Z. Liu, "Unsupervised random forest indexing for fast action search," Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), 2011.

[21] F. Jiang, J. Yuan, S. A. Tsaftaris and A. K. Katsaggelos, "Anomalous video event detection using spatiotemporal context," Computer Vision and Image Understanding (CVIU), vol. 115, No. 3, pp. 323-333, 2011.

[22] B.-L. Li, M. Ayazoglu, T. Mao, O.-I. Camps, and M. Sznaier, "Activity recognition using dynamic subspace angles," Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'11), 2011.

[23] C. J. Wu and W. H. Tsai, "Location estimation for indoor autonomous vehicle navigation by Omni-directional vision using circular landmarks on ceilings," Robotics and Autonomous Systems, Vol. 57, No. 5, pp. 546-555, 2009.

[24] B. Jung and G. S. Sukhatme, "Detecting moving objects using a single camera on a mobile robot in an outdoor environment," In the 8th Conference on Intelligent Autonomous Systems, pp. 980-987, March 10-13, 2004.

[25] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I: the essential algorithms," IEEE Robotics and Automation Magazine, vol.13, no.2, pp.99-110, 2006.

[26] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): Part II: State of the art," IEEE Robotics and Automation Magazine, vol.13, no.3, 108-117, 2006.

[27] J.-W. Hsieh, "Fast stitching algorithm for moving object detection and mosaic construction," Image Vision and Computing Journal, vol. 22, no. 4, pp. 291-306, April 2004.

[28] M. Castelnovi, P. Musso, A. Sgorbissa, and R. Zaccaria, "Surveillance robotics: analyzing scenes by colors analysis and clustering," IEEE International Symposium on Computational Intelligence in Robotics and Automation, vol. 1, pp. 229-234, July 2003.

[29] L. P. Chew, "Constrained delaunay triangulations," Algorithmica, vol. 4, no.1, pp. 97-108, 1989.

[30] T. O. Kvalseth, "On exponential entropies," Proc. IEEE Int. Conf on System, Man, and Cybernetics, pp. 2822-2826, 2000.

[31] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[32] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," Computer Vision and Image Understanding, vol. 110, no. 3, pp. 346-359, 2008.

[33] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: survey and experiments," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.31, no.12, pp.2179-2195, 2009.

[34] P. F. Felzenszwalb, et al.,"Object detection with discriminatively trained part based models," IEEE transactions on Pattern Analysis and Machine Intelligence, pp.1627-1645, Sept. 2010.

[35] J.-W. Hsieh, C.-L. Lin, P. Wu, J.-C. Cheng, and D.-Y. Chen, "Handhold object detection and event analysis using visual interaction clues," International Conference on Distributed Multimedia Systems, Italy, Aug. 18-20, 2011.

[36] M. Sonka, V. Hlavac, and R. Boyle, Image Processing, Analysis, and Machine Vision, London, U. K., Chapman & Hall, 1993.

[37] X.-B. Cao, C.-X. Wu, J.-H. Lan, P.-K. Yan, and X.-L. Li, "Vehicle detection and motion analysis in low-altitude airborne video under urban environment," IEEE Trans. on Circuits and System for Video Technique, vol. 21, no. 10, pp.1522-1533, 2011.

[38] M. Ebrahimi and W. W. Mayol-Cuevas, "Adaptive sampling for feature detection, tracking, and recognition on mobile platforms" IEEE Trans. on Circuits and System for Video Technique, vol. 21, no. 10, pp.1467-1475, 2011.

[39] T. Chen, K.-H. Yap, and L.-P. Chau, "Integrated content and context analysis for mobile landmark recognition," IEEE Trans. on Circuits and System for Video Technique, vol. 21, no. 10, pp.1476- 1486, 2011.

[40] J. Pilet, C. Strecha, and P. Fua, "Making Background Subtraction Robust to Sudden Illumination Changes," In Proc. European Conf. on Computer Vision, 2008.

[41] B. Jung and G. S. Sukhatme, "Real-time motion tracking from a mobile robot," International Journal of Soc. Robot, vol.2, pp.63-78, 2010.

[42] J.-W. Hsieh, F.-J. Fang, G.-J. Lin, and Y.-S. Wang, "Template matching and Monte Carlo Markova chain for people counting under occlusions," pp.761-771, 18th International conference on Multimedia Modeling, Austria, Jan.4-6, 2012.

[43] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization," In 14th Int'l Conf. on Machine Learning, pp. 412–420, 1997.

[44] Shapiro, L., and Stockman, G. (2002). Computer Vision. Prentice Hall. pp. 69–73.

[45] C. Micheloni, B. Rinner, and G. L. Foresti, "Low-Level Processing in PTZ Camera Networks," IEEE Signal Processing Magazine, Vol. 27, No. 5, pp. 78-90, 2010.

[46] M. Dragusu, A. N. Mihalache, and R. Solea, "Practical applications for robotic arms using image processing," 16th International Conference on System Theory, Control and Computing, 2012, pp. 1 - 6, 12-14 Oct. 2012.