Background Subtraction using an Active Contour Model

Sakharin Buachan [1], Teerasit Kasetkasem [2], Sanparith Marukatat [3] and Hiroaki Kunieda [4]

[1] ICTES program, Electrical Engineering Department, Faculty of Engineering, Kasetsart University, Bangkok, Thailand.

[2] Electrical Engineering Department, Faculty of Engineering, Kasetsart University, Thailand.

[3] Image Technology Laboratory, National Electronics and Computer Technology Center, Thailand.

[4] Department of Communications and Integrated Systems, School of Science and Engineering, Tokyo Institute of Technology, Japan.

Correspondence:
Sakharin Buachan
ICTES program, Electrical Engineering
Department, Faculty of Engineering,
Kasetsart University, Bangkok, Thailand.
Email: g5314552531@ku.ac.th

Abstract

Clear background images are useful for various video processing algorithms. Using the fact that non-background objects and their background scenes are not dependent, a contour-based background subtraction algorithm is introduced. The proposed contour-based background subtraction relies on maximizing the joint distribution between the background image and the contour. The proposed algorithm can correctly guide the contour to fit a non-background object even if the initial contour is located far from the boundary. Contour tracking is applied to predict the initial contour of the current frame. Using it results in faster convergent, thus, the initial contour is positioned near the object boundary. An inital initial contour generator is used in the case that the contour tracking cannot find the best estimated initial contour. The algorithm is able to be used without requiring human intervention. The experimental results show that the proposed algorithm has benefits over background subtraction algorithms, the moving average (MA), and the moving average with selective (MAS), with extra computation cost. Comparing selected background images from the video scene, the average of root mean square is 5.65 pixel values, which is much lower than 8.23 from the MA algorithm and 9.72 from the MAS algorithm.

Keywords: Active Contour Model; Background Subtraction; Background-Foreground Model; Probabilistic Approach

1. Introduction

Background subtraction in image processing is used to extract non-background objects such as foreground, shadows, reflections, etc., from captured images and, for some algorithms, to recover the missing parts of background that those objects cover. The results, the clear background images, are useful for various purposes in image processing. Analysis tools can gain benefits from the clear background, especially for video surveillance systems. With clear background images, object tracking, object detection, object classification, and object recognition algorithms can be more accurate [1].

The complexity of a background subtraction algorithm depends on the complexity of the input images. For some image sequences, a basic background subtraction algorithm named moving average is proper. Moving average (MA) is the same concept as an infinite impulse response (IIR) filter in signal processing [2], [3], [4]. For image sequences which are taken from a static camera, having few moving foreground objects and have no static foreground, the background images from this algorithm are quite clear. Since using foreground as a part of background is not reasonable, the moving average with selective (MAS) algorithm excludes foreground objects from the background construction [4]. However, the MAS algorithm needs background/foreground information to selectively update clear background images. The background images from the MAS algorithm can be better than the MA algorithm, but quality of them depends directly on the correctness of the background-foreground information.

Pixels in background image, Z_t, of the t^{th} frame which is defined as $z_t(S)$ can be constructed from the combination of pixel $y_t(S)$ of capture image, Y_t, and the pixel $z_{t-1}(S)$ of the previous background frame. The MA and MAS algorithms are defined as Eq.1 and Eq.2, respectively.

$$z_t(s) = \alpha y_t(s) + (1-\alpha) z_{t-1}(s) \tag{1}$$

$$z_t(s) = \begin{cases} \alpha y_t(s) + (1-\alpha) z_{t-1}(s), s \in BG \\ z_{t-1}(s) & \text{, otherwise} \end{cases} \tag{2}$$

The MAS can be adapted with various object detection algorithms in order to construct a clear background image from their background-foreground information. One kind of algorithms adapted for object extraction is contour-based algorithms. Contours are used to separate non-background (NBG) objects from background (BG) image in various algorithms [5], [6], [7], but the well-known contour-based algorithm which is used in this study is the active contour model.

2. The Active Contour Model

Active Contour Model (ACM) was proposed by Kass et al. in 1988 [8]. The algorithm uses a contour to detect edges or lines in images. This simple algorithm is applied to feature extraction algorithms [9], [10] and there are improvements applied for faster convergent or better extraction [11], [12], [13], [14], [15], [16], [17]. Moreover, there are algorithms which use contour to separate features from the images [5], [7], [18]. These documents clearly show benefits of contour-based algorithms in object detection, extraction and tracking.

The contour X_t of t^{th} frame is a closed graph consisting of K vertices. A vertex $P_{t,k} = (x_{t,k}, y_{t,k})$ is the k^{th} vertex locating at $(x_{t,k}, y_{t,k})$ where $k \in [1,K]$. The graph has to be initialized around the object of interest. This graph is called the initial contour. At the initial state, vertices in an initial contour are manually located at positions near the features. Initializing the contour can be done by human interpolation. However, for video processing, it must be done by some automatic methods [9], [10].

The algorithm uses an energy-minimizing technique as a force to guide each vertex to feature points. Once a vertex is moved to a better position, the same operation is applied to the next vertex in the contour until a convergent condition is reached. The better position of a vertex can be found by searching it on a bounded area around the vertex called the searching window. Thus, a more effective algorithms can be used instead.

Using the searching window, the algorithm can be called “Greedy Snake”. The size of the searching window directly affects the speed of the Greedy Snake algorithm. Bigger searching windows increase chance of the vertices to reach their optimum positions, but they require higher computational cost. There are algorithms which are designed to increase its size without increasing computational cost. For example, particle swarm optimization is adapted to reduce computational cost of large search windows [16].

Using the concept of entropy, an energy function defines how positions of vertices are close to features in the image. The better positions of vertices result in lower energy. The ACM consists of two types of energy terms, the internal and external energies. The internal energy terms are calculated from the contour. They are responsible to stabilize the shape of the contour, or maintain equally distances between vertices. Some internal energy terms promote vertex movement which agrees to movement of its adjacent vertices [13]. The external energy terms are calculated from the image. They are designed to guide the contour to features in the image.

For the conventional ACM, there are three energy terms, continuity and curvature are internal energies, and gradient is an external energy. Since vertices are likely to have roughly the same distances apart and no two (or more) vertices should occupy the same location, this energy is calculated from distances between vertices and promotes the movement which equalizes the distances between vertices. The continuity energy is defined as:

$$E_{cont}(p_{t,k}) = avgDist - \left\| p_{t,k-1}(s) - p_{t,k}(s) \right\|_2 \tag{3}$$

where $p_{t,k-1}$ is a adjacent vertex of $p_{t,k}$ and *avgDist* is defined as:

$$avgDist = \frac{1}{K} \times \sum_{p_{t,j} \in X_t} \left\| p_{t,j-1} - p_{t,j} \right\|_2 \tag{4}$$

The continuity energy is the lowest if the distances between vertices are the same. With this energy term, vertices are guided to separate themselves from adjacent vertices and stop when all distances are the same.

The object boundary is assumed to be continuous and smooth lines. Therefore, the contour should be continuous and smooth as well. To keep the contour smooth and to avoid sharp edges, there is an energy term for this purpose. Curvature energy is computed from the straight lines between edges. It promotes the edge pair with less curvature, which will result in a smoother contour than a pairs with higher curvature. The curvature energy is defined as:

$$E_{curv} = \left\| p_{t,k-1} - 2 \times p_{t,k} + p_{t,k+1} \right\|_2 \tag{5}$$

where $p_{t,\,k-1}$ and $p_{t,k+1}$ are adjacent vertices of $p_{t,k}$.

The curvature energy is the lowest if the whole contour becomes a smooth curve or a circle. This energy term will promote the vertex movement which increases the smoothness of the contour. It provides good extraction results for objects with a smooth edge. However, for objects with corners, this energy term will lead to false detection on the corners since the high curvature edge pairs will be demoted.

The gradient is the only energy term which can pull a contour to the object boundary. Since a high change in intensity area is discontinuity between foreground objects and background scene, gradient energy calculated from change of intensity of a gray scale image can represent the position of the object boundary. The gradient energy is defined as:

$$E_{grad} = -\left\| \nabla I(p_{t,k}) \right\|_2 \tag{6}$$

The gradient is only calculated along a single axis at a time before both axis gradients are combined. In the case that the image is true color, the image has to be converted to a gray scale image before the gradient energy is calculated.

Since the values of these three energy terms are not similar in both value and scale, the energy function must be a weighted summation of these energy terms. The weighting values should equalize the forces from those three energy terms and adjust the importance of each energy term at once. So weighing the energies is a complex task. A better way to weight them is to normalize each energy term in the searching window before the summation of them as the energy function. With this technique, the weighting values will responsible to adjust the importance of each energy term only, and the tuning process will become a simpler task.

The Active Contour Model (ACM) is simple and effective, but the only feature used for extracting an object from the background image is the object boundary. For some objects and backgrounds having little texture, this boundary is suitable to separate background and non-background areas. However, for high texture background, this approach may lead to incorrect segmentation results. Hence, the ACM is needed to be modified in order to cope with this situation.

One feature which can be used to extract the non-background areas in a video scene is the difference in intensity between backgrounds and non-backgrounds. In this study, the objects which do not belong to background, such as foreground objects and shadows are needed to be excluded from the backgrounds, and these objects are called non-background objects. The ACM is adapted to deal with this problem by separating non-background and background from each other. Using a probabilistic approach, the ACM is guided by the variation of subtraction image until the probability of the contour and background image is maximized.

3. The Problem Statement

Let $Y_t \in \Re^{M\times N}$ and X_t be the observed image of size M x N, and the corresponding boundary of the non-background in the t^{th} frame. Here, X_t is defined as a closed contour (CC) having K vertices that separates a non-background object from the scene. Let $p_{t,k} = (x_{t,k}, y_{t,k})$ where $k \in [1,K]$ is the k^{th} vertex locating at $(x_{t,k}, y_{t,k})$. Pixels inside and outside the closed contour X_t are called the non-background and background pixels, respectively. For simplicity, we assume that there is only one connected non-background area in a scene at a given time. Since X_t is the contour describing the border of non-background area, the continuity and curvature energy of the ACM is selected as a marginal probability of X_t. Indeed, the marginal probability of X_t is defined as:

$$\Pr(X_t) \propto \exp[-(E_{cont} + E_{curv})] \tag{7}$$

where E_{cont} and E_{curv} are the continuity and curvature energies used in the active contour model [8]. The continuity represents average distances between vertices on the contour, since vertices are likely to have roughly the same distances apart and no two (or more) vertices should occupy the same location. The curvature energy represents the smoothness of the contour. We assume that a border contour is unlikely to have shape edges and the non-background objects are likely to occupy a convex area in a scene. The continuity and curvature energy terms are defined in Eq.3 and Eq.5 respectively.

Next, denote Z_t as the corresponding background image (BGI) which is the observed image at the t^{th} frame if no non-background object is presented on the frame. Here $z_t(s)$ denotes an intensity value of background image at a pixel s. In this study, the BGI is assumed to change very slowly, i.e. Z_t Z_{t-1}. Next, if vertices in X_t do not occupy the same position, F_t denotes the corresponding observed data in the non-background area that is the area within the contour X_t, and $f_t(s)$ is an intensity value of a pixel s if s belongs to a non-background area. Hence, for a pixel s, the observed data is given by:

$$y_t(s) = \begin{cases} z_t(s), s \text{ is background} \\ f_t(s), s \text{ is non} - \text{background} \end{cases} \tag{8}$$

where background and non-background areas are specified by the contour X_t.

We assume that all pixels are independent and are normally distributed around previous BGI with common variance. The probability density function (PDF) of BGI at t^{th} frame is then defined as:

$$\Pr(Z_t|Z_{t-1}) = \prod_{s\in S} \Pr\left(z_t(s)|z_{t-1}(s), \sigma^2\right) \tag{9}$$

Here S is a set of all pixels in the image and Pr$(z_t(s) \mid z_{t-1}(s),$ and σ^2 satisfies a Gaussian distribution with mean $z_{t-1}(s)$ and variance σ^2.

Next, we assume that, for a given BGI of the (t-1)th frame, the intensity values of a non-background image (NBGI) are statistically independent and have a uniform distribution for all possible values of gray scales, which is in the range 0 to 255 for 8-bit gray scale image, i.e.:

$$\Pr(F_t|Z_{t-1}) = \prod_{s\in S} \Pr\left(f_t(s)\right) \tag{10}$$

where Pr $(f_t(s))$ satisfies a probability mass function of a discrete uniform distribution of all possible pixel values. This assumption can be modified if prior information regarding the non-background (NBG) area is available. In the next section, we will investigate the optimum criteria to extract the BGI from the observed video data.

4. Optimum Criteria

The maximum a posteriori (MAP) criteria is chosen to find the optimum close contour (CC) and the BGI at the t^{th} frame from the observed image at the same frame, and the estimated BGI from previous frame. The MAP criteria is given by:

$$(Z_t, X_t)^{opt} = \arg\left[\max_{Z_t, X_t} \Pr\left(Z_t, X_t|Y_t, Z_{t-1}\right)\right] \tag{11}$$

To find the optimum background image Z_t let us define $I_b(s,X_t)$ as an indicator whether a pixel s belongs to non-background or background image with respect to the contour X_t. The value of $I_b(s,X_t)$ is one if s belongs to background area, which is the area outside the contour X_t, and zero otherwise. Now, the optimum background image of the current frame is defined as the combination of Z_{t-1} and Y_t with the area specified by the contour X_t as:

$$z_t^{opt}(s) = \begin{cases} \alpha y_t(s) + (1-\alpha) z_{t-1}(s), I_b(s,X_t) = 1 \\ z_{t-1}(s), I_b(s,X_t) = 0 \end{cases} \tag{12}$$

From the above equation, the optimum background image Z_t is known from choice of X_t. Now, the optimum Z_t and X_t can be found from the choice of X_t only.

Using the chain rule, the Eq.11 can be rewritten as:

$$\begin{aligned} X_t^{opt} &= \arg\left[\max_{X_t}\left[\frac{\Pr(X_t, Y_t, Z_{t-1})}{\Pr(Y_t, Z_{t-1})} \cdot \frac{\Pr(X_t, Z_{t-1})}{\Pr(X_t, Z_{t-1})}\right]\right] \\ &= \arg\left[\max_{X_t}\left[\Pr(Y_t|X_t, Z_{t-1}) \cdot \Pr(X_t, Z_{t-1})\right]\right] \end{aligned} \tag{13}$$

Since the observed image is segmented into non-background and background, the Eq.13 is investigated separately in two cases: namely non-background and background pixels.

For background pixels, the background images must be similar to the observed image, i.e., $z_t(s) = y_t(s)$for $I_b(s, X_t) = 1, s \in S$. In addition, for background pixels, the term Pr $(Y_t|X_t, Z_{t-1})$ in Eq.13 becomes:

$$\prod_{I_b(s,X_t),s\in S} \Pr\left(y_t(s)\middle|z_{t-1}(s)\right) = \prod_{I_b(s,X_t),s\in S} \Pr\left(z_t(s) = y_t(s)\middle|z_{t-1}(s)\right). \tag{14}$$

For non-background pixels, the intensity values can have any value and they are independent of Z_{t-1}. Thus, the term Pr $(Y_t \mid X_t, Z_{t-1})$ becomes Pr (F_t). Hence, the term Pr $(Y_t \mid X_t, Z_{t-1})$ in the Eq.13 for non-background area can be rewritten as:

$$\prod_{I_b(s,X_t),s\in S} \Pr\left(y_t(s)\middle|z_{t-1}(s)\right) = \prod_{I_b(s,X_t),s\in S} \Pr\left(f_t(s)\right) \tag{15}$$

where $f_t(s)$ denotes a pixel value of pixel s in the non-background image. Finally, the Eq.11 can be rewritten as:

$$X_t^{opt} = \arg\left[\max_{X_t}\left(\prod_{s\in S}\left[\begin{matrix}[\Pr\left(z_t(s) = y_t(s)\middle|z_{t-1}(s)\right)] \\ \cdot I_b(s, X_t) + \Pr\left(f_t(s)\right) \\ \cdot \left(1 - I_b(s, X_t)\right)] \cdot \Pr\left(X_t\right)\end{matrix}\right]\right)\right]. \tag{16}$$

$$\Pr\left(z_t(s) = y_t(s)\middle|z_{t-1}(s)\right) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_t(s)-z_{t-1}(s)\right)^2}{2\sigma^2}} \tag{17}$$

if s belongs to background area, from Eq.10, we have:

$$\Pr\left(z_t(s) = y_t(s)\middle|z_{t-1}(s)\right) = \frac{1}{255} = e^{\ln\frac{1}{255}} = e^{-\ln 255} \tag{18}$$

if s belongs to non-background area, and from Eq.7, we have:

$$\Pr(X_t) = e^{-(E_{cont}+E_{curv})} \tag{19}$$

Substituting, Eq.17, Eq.18 and Eq.19 into Eq.16, we obtain:

$$X_t^{opt} = \arg\left[\max_{X_t}\left(\prod_{s\in S}\left[\begin{matrix}[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\left(y_t(s)-z_{t-1}(s)\right)^2}{2\sigma^2}} \\ \cdot I_b(s, X_t) + e^{-\ln 256} \\ \cdot \left(1 - I_b(s, X_t)\right)] \cdot \Pr\left(X_t\right)\end{matrix}\right]\right)\right] \tag{20}$$

By taking a natural logarithm, the above equation can be modified further as:

$$X_t^{opt} = \arg\left[\min_{X_t}\left(E_{prob} + E_{cont} + E_{curv}\right)\right] \tag{21}$$

where

$$E_{prob} = \sum_{s \in S}\begin{bmatrix} E_{BG}(s)I_b(s, X_t) \\ +E_{NBG}(s)\left(1 - I_b(s, X_t)\right)\end{bmatrix} \tag{22}$$

In this equation, $E_{BG}(s) = \frac{\left(y_t(s) - z_{t-1}(s)\right)^2}{2\sigma^2} + \frac{1}{2} \log (2\pi\sigma^2)$ are $E_{NBG}(s)$ = In (255) the energies when the observation $y_t(s)$ are background and non-background, respectively. Note that the values of E_{prob} changes according to the choice of X_t.

Eq.21 can indicate the optimum contour and it can be used to find the optimum background image. However, the equation does not show the algorithm which can construct those terms. To solve for the solution, the active contour model is adapted for the optimum contour and the optimum contour is used to construct an optimum background image.

5. Solution of the Optimum Criteria

The solution of Eq.21 cannot be analytically obtained since the Z_{t-1} changes according to the choice of X_t. So, the proposed algorithm determines the solutions using a three-step approach.

First of all, without human intervention required, the initial contour generator is designed to handle the contour generator in the case that a new initial contour is needed. Next, the solution of Eq.21 will be solved by an ACM-base algorithm. Finally, a clear background image will be constructed from the solution of the previous step. This three-step approach is designed to be an automatic algorithm which can provide clear and updated background images for detection algorithms without human intervention required.

5.1 Initial Contour Generator

Assume that a background image Z_{t-1} of the previous frame is constructed by the proposed algorithm and an initialized background image, which is constructed by some simple background subtraction algorithms such as moving average [3], and does not consist of parts of non-background objects. Differences between frames are further assumed to be small. Moreover, the non-background objects are assumed to be different from the background image. Thus, it is clear that the high and low variation areas in subtraction image D_t between a background Z_{t-1} of the $(t\text{-}1)^{th}$ frame, and the observed image Y_t of the t^{th} frame, can be defined as an estimated non-background and an estimated background, respectively, as:

$$D_t = \begin{cases} NBG^{est}, & |Y_t - Z_{t-1}| > \tau \\ BG^{est}, & otherwise \end{cases} \tag{23}$$

where τ is a predefined threshold.

To reduce noise and smoothen the result from the Eq.23, a dilation operation is applied to the subtraction image D_t [9]. Then the threshold is applied to that result.

As mentioned that only one non-background object can appear in the scene as a time, a bounding box can be drawn around the estimated non-background and the initial contour can be automatically generated by positioning vertices on that bounding box. The distances between vertices in the initial contour should be equal, and the number of vertices depends directly on the distances between vertices.

5.2 Object Tracking

Assume that on some frames, non-background objects move slowly and their shapes are not varying significantly. The converged contour of previous frame X_{t-1} can be used as an initial contour of current frame. Instead of using it directly, the contour is moved along *x*- and *y*- axes on a limited area. On each position, the energy function in Eq.21 is calculated. The best match contour which has the lowest energy is used as the initial contour of current frame.

However, before the contour tracking is applied to the contour X_t, the system has to make sure that the initial contour is still usable. In the case that the converged contour of previous frame fail to track the non-background object, the contour usually becomes a small circle. So, the average distance of vertices in the contour is checked to detect this mistake. If the average distance is less than a constant, the initial contour is regenerated to recover the system, before the contour tracking will be executed. The converged contour is also compared to the frame differences in order to check if the contour miss-tracked the object or not.

5.3 Searching for the Solution Final Equation

Now, the problem of searching for a boundary between non-background and background areas becomes a problem of maximizing posteriori probability. To solve this problem, X_t and Z_t which maximize Pr $(X_t, Z_t \mid Y_t, Z_{t-1})$ are needed to be estimated. Finding all possible X_t and Z_t is not suitable in real applications. Computing the probability from the whole image is also not necessary. So, a vertex in the initial contour is moved to candidate positions in a searching window and the probabilities of moved versions are computed from all pixel values in a region of interest. This technique can reduce computational cost and represents approximate probabilities of the whole image calculation.

Once the probabilities of all candidate contour versions are computed, the best candidate contour will become a better candidate contour. Since the vertex movement can improve the correctness of the candidate contour, applying the same operation to all vertices in the candidate contour until one of convergent conditions is reached can finally find the best candidate contour, which separate non-background area from background area, and it becomes the converged contour X_t.

Fig. 1 shows the region of interest and a searching area of a vertex $P_{t,k}$. The size of a region of interest and searching area of a vertex can be adjusted, but the region of interest must be bigger. For example, if the region of interest is 5×5 pixels, the size of the searching area should be 3×3 pixels.

We have experimentally found that by slightly modifying the energy function proposed in the Eq.21, a better result can be obtained. The constant term $\frac{1}{2}\log(2\pi\sigma^2)$ is eliminated from E_{BG} and it somehow increases the importance of the energy of the non-background component. Preliminary experiments showed that this modification does increase the accuracy of the algorithm. In the following, we rely on this modified energy function.

5.4 Background Construction

The updated background can be constructed by using a simple background subtraction technique, moving average with selective (MAS) Thus, it is proper for contour-based background subtraction [2], [3]. From the converged contour X_t which is already fit to the non-background area in the observed image Y_t, the background image is represented by an area outside the contour. The updated background of t^{th} frame, Z_t, is defined from:

$$z_t(s) = \begin{cases} \alpha y_t(s) + (1-\alpha) z_{t-1}(s), & s \text{ is outside of } X_t \\ z_{t-1}(s) & \text{, otherwise} \end{cases} \tag{24}$$

where *a* is a leaning rate constant.

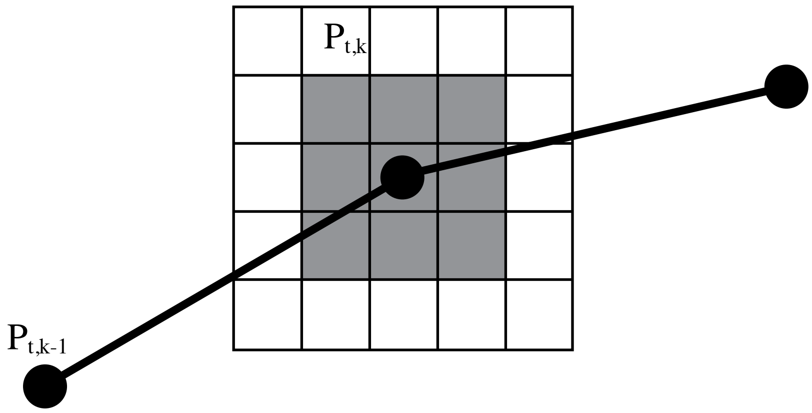

Fig. 1. The searching window (gray pixels) and the region of interest (gray and white pixels) of a vertex $P_{t,k}$.

6. Experimental Results

For testing the accuracy of the algorithms, a video sequence is used. The video is taken from a CCTV camera of a CCTV system in Ngamwongwan parking building 1, Kasetsart University. The camera is installed for taking video of cars entering the second floor of the parking building.

In the video, cars and motorcycles come toward to the camera and they can turn either left or right. Since the entrance has a barrier and a speed hump, vehicles usually slow down at that area. Even if most vehicles move slowly, sometime motorcycles can take a bypass and move faster. For some fast moving cars and motorcycles, they may appear on only a single frame before they pass by.

Even if vehicles go toward to the camera, the vehicles which turni left may have their shapes and sizes changed rapidly. Big cars can instantly change the intensity in the scene and cause shadow problems. Since the building is open-air, the lighting condition outsides the building can suddenly changes intensity in a scene.

The video is a 300×222 pixel color video. The whole video is long, but only 600 frames are tested and observed. These 600 frames consist of background only frames which can be used to identify the accuracy of the background subtraction algorithms, moving and stopping cars, motorcycles, and people walking and standing in the scene. Moreover, the light conditions change due to clouds and there are tree leaves moving in the scene. These details can affect the accuracy of the algorithm.

Moreover, to make sure that the algorithm can provide good updated background images, 31 background-only images are manually selected from the video. Assume that the background does not change much, these background images are compared to the updated background from the proposed algorithm. Comparing results from the conventional and simple background subtraction algorithms, moving average (MA) and moving average with selective (MAS), the benefits of the proposed algorithm can be observed.

The results of the algorithm are contours which can separate non-background area from background area and the updated background, which is adapted to the changes in the scene but does not contain non-background objects. To specify the accuracy of the algorithm, the results are checked by observes. The results of the proposed algorithm are shown and described by cases.

Fig. 2 shows converged contours in the case that there is a non-background object in the scene and the variation between non-background and background area is high. Moreover, in this case, the intensity does not change much and a shadow problem does not occur. This case is expected since it matches assumptions in the problem statement. With high variation in the subtraction image at the non-background area and low variation at the background area, the probability energies are highly different between correct and incorrect contours. The force from this energy term will guide the contour to extract the non-background object from the background quickly and correctly. In addition, the correct contour results in a good constructed background image.

Since the camera is pointing to the entrance of a parking lot, sunlight from outside can directly change the intensity of the video. Moreover, the movement of cars results in moving shadows, which increase the variation in a scene. These conditions change variation in a scene and can cause problems to simple background subtraction algorithms, especially moving average with selective.

Since the algorithm detects the variation of the video sequence and assumes that high variation areas contain parts of non-background objects, the energy of these high variation areas can expand the contour to cover them. So, in this case, the contour is guided to extract both foreground object and its shadow on the background. Shown in Fig. 3, shadows are included in the non-background area.

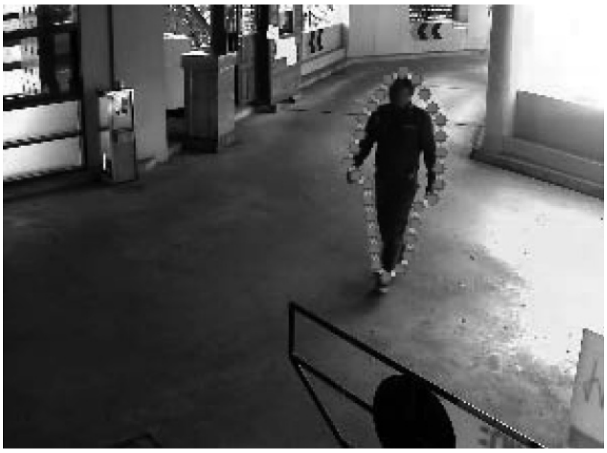

Fig. 2. Converged contour in the case of single objects.

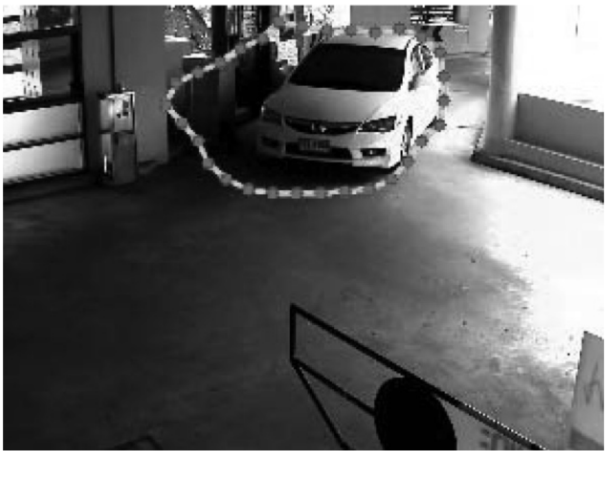

Fig. 3. Converged contour in the case of single objects with shadow.

For the simplicity of the algorithm, the proposed method is designed for single non-background areas and the implementation does not deal with multiple object problems. In the case that there are multiple non-background objects in the observed image, shown in Fig. 4, the contour cannot correctly separates them from background area, but it will cover all non-background objects as a single area.

Due to the design for single non-background area, the initial contour generator will draw the initial contour around all estimated non-background areas, and the algorithm will separate them from the background scene in one piece. The converged contours may not perfectly cover all objects and usually cover an area between them. Moreover, the converged contours cannot be used to estimate the initial contour for the next frame. These unsatisfied situations will poorly decelerate the algorithm. Fortunately, the incorrect contours can cover almost all non-background areas, which finally results in correct updated background images.

Ghosts or trailing objects usually appear in the updated background images of the MA algorithms. The same as the MA algorithm, static foregrounds result in unrecoverable mistakes of the MAS algorithm. However, the proposed algorithm can eliminate these problems. As shown in Fig. 5, the constructed background image is clear from non-background objects and is updated to changes in background area.

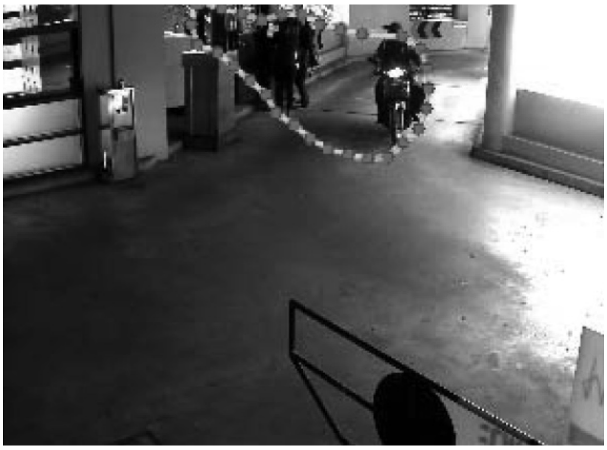

Fig. 4. Converged contour in the case of multiple objects.

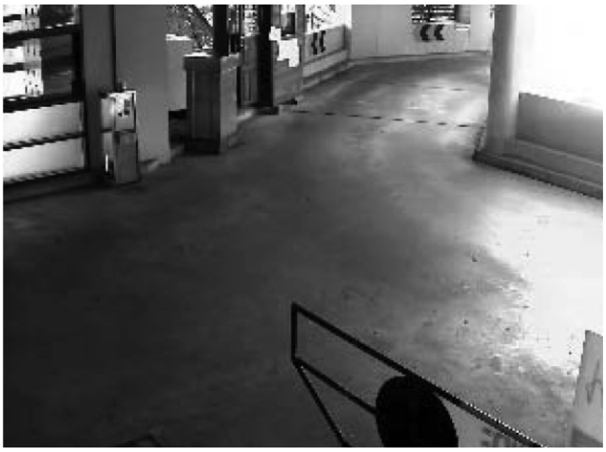

Fig. 5. Background constructed by the proposed algorithm.

Since there is no technique that properly determines the accuracy of background subtraction algorithms [3], 31 background-only images are selected from the video sequence and used instead. We assume that the background changes slowly along the video sequence, and changes of background in the video are linearly adapted from one background frame to the next background frame. So, reference images are defined from the combination of these background-only images. Weighting values are computed from the distances between the two nearest background images to the observed image. Then a reference image is constructed by summation of two background images with weighting values. The reference images are used instead of the manually defined background images. The technique may not be proper for determining the accuracy of these three background subtraction algorithms since the distances between a observed image and its two nearest background image may be too far. In this study, a background image appears on frame number 12 and the nearest background image appears on the frame number 134. In this region, reference images may not be satisfied. Moreover, changes in the observed images may not be similar to the selected background images. Results of the comparison are shown in terms of root mean square in Fig. 6.

From Fig. 6, some frames (such those of 281, 289, 430, etc.) are frames which the reference background images are constructed by a new pair of selected background images, and intensity in these images is changing. At these frames, discontinuity can occurs in the graph. The MAS algorithm results in low RMS values on frame numbers 1 to 160. Since these frames consist of moving foreground objects and the intensity of those frames does not change much, the low average RMS shows that the MAS algorithm is proper for these cases (see table 1). However, the intensity starts to change after the frame number 160, which results from an unrecoverable problem of the MAS algorithm. With this problem, the MAS algorithm cannot provides good background images anymore, and it results in high RMS values compared to the proposed algorithm and the MA algorithm.

The graph clearly shows that the proposed algorithm is better than the MA since the RMS values of the proposed algorithm are, mostly, lower then the RMS values of the MA algorithm. From the figure, static foregrounds appearing on frame numbers 70 to 140 cause high mean square error of the MA algorithm. The same problem occurs on frame numbers 200 to 230 and frame numbers 325 to 340. Unlike the MA algorithm, the proposed algorithm can reduce this problem, and its RMS values are lower on these frames. These results show that the background images constructed by the proposed algorithm are more similar to the real background scene than the background image from the MA and the MAS algorithm.

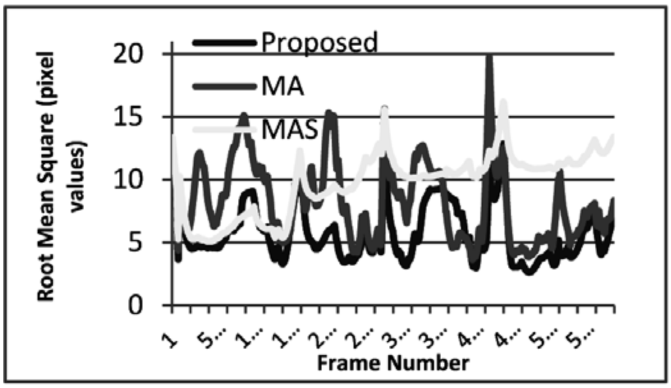

Fig. 6. Root mean square of constructed background images, compared to reference images.

Table 1 shows the average of RMS values along the video frames. In this table, the average of RMS is separated into two parts at frame number 160 since the MAS algorithm cannot construct good background images after the intensity has changed. Without the intensity problem, the RMS of frame numbers 1 to 160 clearly show the benefit of the MAS algorithm over the MA algorithm. The average RMS shows that the proposed algorithm can construct background images which are more similar to the reference background images. Moreover, with the intensity problem, the lower average RMS shows the benefit of the proposed algorithm over the MA and the MAS algorithms.

7. Conclusion

Differences in background and non-background areas in videos can be a feature which can separate them from each other. In this study, the idea of applying the differences between an observed image and a background image to separate non-background objects from background scene is introduced and discussed. From the statement of problem, the variation from a subtracted image can be use as a feature to extract a non-background object from background image.

Shown in Fig. 6 and Table 1, The proposed algorithm can successfully extract non-background area from the background area and, it can construct updated background, which is adaptive to changes in the video sequence, especially the light intensity. The algorithm is tested with real video sequence from CCTV installed in Ngamwongwan parking building 1, Kasetsart University. With an initial contour generator which allows this system to be automatically executed without any human intervention, the results successfully show the improvement from conventional background subtraction algorithms. In all cases, the algorithm clearly shows that the algorithm successfully extracts non-background areas from background areas. Even if the contour does not perfectly fit the boundary between these areas due to high variations on some area such as shadow areas, the algorithm can work in cases that detection and extraction fails.

Table 1 Average of root mean square of frames in the video sequence comparing to reference images (Frame number.):

Frame No.	Proposed	MA	MAS
1 - 160	5.72	9.32	6.36
161 - 600	5.62	7.83	10.94
1 - 600	5.65	8.23	9.72

8. References

[1] C. Stauffer and W.E.L. Grimson, “Adaptive background mixture models for real-time tracking,” in Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, Fort Collins, p. 2, 1999.

[2] Alan M. Mcivor, “Background Subtraction Techniques,” in In Proc. of Image and Vision Computing, Auckland, 2000.

[3] M. Piccardi, “Background subtraction techniques: a review,” in Systems, Man and Cybernetics, 2004 IEEE International Conference on, Vol. 4, The Hague, pp. 3099 - 3104, 2004.

[4] Y. Tian. (2008, April) Andrewsenior. [Online]. www.andrewsenior.com/technical/surveillanceclass/02_MovingObjectDetection_part1.pdf

[5] Boo Hwan Lee, Il Choi, and Gi Joon Jeon, “Motion-Based Moving Object Tracking Using an Active Contour,” in Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, p. 2, 2006.

[6] Jianjun Xu, Duyan Bi, and Zifu Wei, “A Novel Robust Contour Tracking Algorithm,” in Information Science and Management Engineering (ISME), 2010 International Conference of, pp. 130 - 134, 2010.

[7] Chih-Yuan Chung and Homer H. Chen, “Video Object Extraction via MRF-Based Contour Tracking,” in Circuits and Systems for Video Technology, IEEE Transactions on, pp. 149 - 155, 2010.

[8] Michael Kass, Andrew Witkin, and Demetri Terzopoulos, “Snakes: Active contour models,” International Journal of Computer Vision, pp. 321 - 331, 1988.

[9] Charles C. H. Lean, Alex K. B. See, and S. Anandan Shanmugam, “An Enhanced Method for the Snake Algorithm,” in Innovative Computing, Information and Control, 2006. ICICIC ‘06. First International Conference on, pp. 240 - 243, 2006.

[10] Yulan Liu and Silong Peng, “A New Motion Detection Algorithm Based on Snake and Mean Shift,” in Image and Signal Processing. CISP ‘08. Congress on, 2008, pp. 140 - 144, 2008.

[11] Chenyang Xu and J.L. Prince, “Gradient vector flow: a new external force for snakes,” in Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on, pp. 66 - 71, 1997.

[12] Chenyang Xu and J.L. Prince, “Snakes, shapes, and gradient vector flow,” Image Processing, IEEE Transactions on, Vol. 7, No. 3, pp. 359 - 369, 1998.

[13] Dong Joong Kang, “A fast and stable snake algorithm for medical images,” Pattern Recognition Letters, Vol. 20, No. 5, pp. 507 - 512, 1999.

[14] Tim McInerney and Demetri Terzopoulos, “T-snakes: Topology adaptive snakes,” Medical Image Analysis, Vol. 4, No. 2, pp. 73 - 91, 2000.

[15] An Guocheng, Wu Zhenyang, and Chen Jianjun, “A fast external force model for snake-based image segmentation,” in 9th International Conference on Signal Processing, 2008. ICSP 2008, pp. 1128 - 1131, 2008.

[16] Rui Li, Yirong Guo, Yujuan Xing, and Ming Li, “A Novel Multi-Swarm Particle Swarm Optimization Algorithm Applied in Active Contour Model,” in Intelligent Systems, 2009. GCIS ‘09. WRI Global Congress on, pp. 139 - 143, 2009.

[17] Hua Fang, Jeong Woo Kim, and Jong Whan Jang, “A Fast Snake Algorithm for Tracking Multiple Objects,” Journal of Information Processing Systems, Vol. 7, No. 3, pp. 519 - 530, 2011.

[18] Sundaramoorthi, G, Yezzi, A, and Mennucci, A.C, “Coarse-to-Fine Segmentation and Tracking Using Sobolev Active Contours,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, Vol. 30, No. 5, pp. 851 - 864, May 2008.