Detecting Moving Objects Using a Camera on a Moving Platform

Chung-Ching Lin and Marilyn Wolf
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA, USA
Email: cclin@gatech.edu, wolf@ece.gatech.edu

***Abstract*—This paper proposes a new ego-motion estimation and background/foreground classification method to effectively segment moving objects from videos captured by a moving camera on a moving platform. Existing methods for moving-camera detecting impose serious constraints. In our approach, ellipsoid scene shape is applied in the motion model and a complicated ego-motion estimation formula is derived. Genetic algorithm is introduced to accurately solve ego-motion parameters. After motion recovery, noisy result is refined by motion vector correlation and foreground is classified by pixel level probability model. Experiment results show that the method demonstrates significant detecting performance without further restrictions and performs effectively in complex detecting environment.**

***Keywords*-moving camera, background subtraction**

I. Introduction

Nowadays, more and more moving cameras are used for different applications, such as unmanned aerial vehicle, driving assistant and wide-area video surveillance.

Many sophisticated methods for detecting moving objects have been developed for static cameras (e.g. [1]). However, only fewer work relate to detecting moving objects from a moving platform (e.g.[2]). It is difficult to apply existing methods to effectively subtract image background from videos captured by a gray-level mono camera in a vehicle. Under normal conditions, the environment is complex. Plane shape model cannot represent the scene. In this situation, using plane shape model will cause incorrect ego-motion estimation or inaccurate current frame prediction. Besides, during foreground detection, there are usually noises caused by model imperfection. The other source of noise is from new appearing background when it is wrongly recognized as foreground. Moreover, only edge portion of moving objects could be detected in each adjacent frame because there is only slight movement when comparing adjacent frames. The purpose of this paper is to develop an effective detecting algorithm to overcome such difficulties.

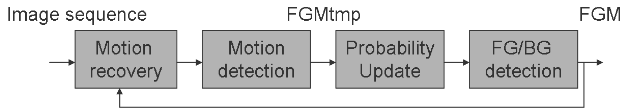
An adaptive particle filter and EM algorithm approach is proposed to detect moving objects based on assuming the scene can be approximated by a plane [3]. In the simple environment, it can be assumed that scene is plane shape. This assumption can only be applied for simple motion and scene. Yamaguchi et al. [4] utilize feature points and epipolar lines to detect moving objects. It assumes that there is no moving obstacle in the initial frame and that the road region in the initial frame is decided according to the height of the camera that is measured when the vehicle is stationery. However, when these assumptions are violated due to presence of moving obstacles in the initial frame or change of camera height, the application of this method would be restricted. Kang et al. [5] propose a method which is capable of detecting moving object by using multiview geometric constrains. However, the approach requires future information which cannot be known at current time. In order to overcome such problems and generate effective results without these restrictions, a new approach of ego-motion estimation and background/foreground classification is developed in this paper. This approach can be generally applied to detect moving objects from videos captured by a camera in a moving platform. Next, the algorithm of proposed method and experiment results are presented.

II. Overview of Proposed Method

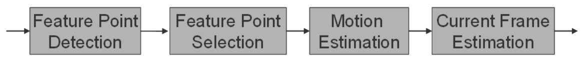
The process flow of proposed method is shown in figure 1(a). The proposed method contains two major parts: motion recovery and segmentation. First of all, estimation model should be established. In order to perform background subtraction in a complex environment, ellipsoid is used as the shape of a scene to develop estimation model. Then, the estimation model can be derived. With the estimation model, motion recovery can be performed. The detailed block diagram of motion recovery is shown in figure 1(b), The first step of motion recovery is feature point detection. Feature points are found by SURF algorithm [6] in every frame. Next, the corresponded feature points between the previous and current frame are matched to obtain feature motion vectors. Only useful feature points are selected for motion estimation. Then, the estimation model utilizes the motion vectors of selected pixels to estimate motion parameters. For solving the complicated estimation model, a genetic algorithm is used to search initial values of motion parameters. After estimating ego-motion parameters, the parameters are then used to estimate current frame.

Following motion recovery, estimated frame, current frame and feature motion vector are utilized to generate foreground template. Finally, probabilities update and foreground/background detection are processed.

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.121

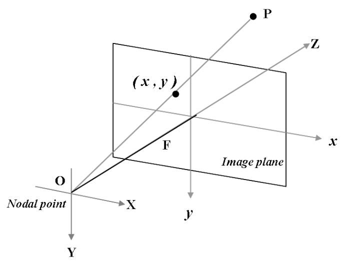


(a) Flow of proposed method

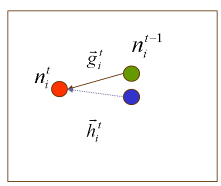


(b) Detailed process of motion recovery

Figure 1. Overview of proposed method



(a) Projection diagram



(b) Image plane and feature motion vector

Figure 2. Image plane projection

III. Motion Recovery

The purpose of motion recovery is to estimate the ego-motion parameters which can be utilized to calculate the corresponding movement of every pixel between adjacent frames. With ego-motion parameters, the estimation of current frame obtained from previous frame is used to compare with the current frame.

A. Estimation Model

Utilizing an ellipsoid to represent the environment is a flexible choice because ellipsoid shape has three degree freedom (length, width and height) and can adapt to different complex environments. Projection model is shown in figure 2(a). After projecting silent scene on the image plane, the relation between camera motion and basic motion vector of every pixel is

$$\begin{cases} \frac{T_Z x - T_X F}{Z} - \omega_Y F + \omega_Z y + \frac{\omega_X}{F} xy - \frac{\omega_Y}{F} x^2 = u(x,y) \\ \frac{T_Z y - T_Y F}{Z} + \omega_X F - \omega_Z x - \frac{\omega_Y}{F} xy + \frac{\omega_X}{F} y^2 = v(x,y) \end{cases} \tag{1}$$

where x, y are known pixel position, u, v are known horizontal and vertical feature point movement. T_X,T_Y and T_Z are translational movement of camera, ω_X,ω_Y and ω_Z are rotational movement of camera, F is focal length and Z is real distance between every point and the camera. These camera motion parameters are unknown and need to be estimated.

It is assumed that camera is at the center of ellipsoid model and its focal length lies on Z axis. After derivation, we can replace Z with following equation.

$$\frac{1}{Z} = \sqrt{\frac{x^2}{F^2A^2} + \frac{y^2}{F^2B^2} + \frac{1}{C^2}} \tag{2}$$

In order to do motion recovery, we use Equation (1) (2) and feature motion vectors (u, v) to estimate camera motion parameters. An accurate set of camera motion parameters can be used to calculate right quantities of every pixel movement. Then, current frame can be estimated by camera motion parameters.

Figure 2(b) is an image plane. n_i^{t-1} is a feature point on previous frame and n_i^t is the corresponded feature point in the current frame. $\vec{g}_i^t = (u, v)$ is the feature motion vector of feature point n_i^t. Since we want to estimate camera motion parameters, we generate an initial guess to calculate feature point movement $\vec{h}_i^t$. Ideally, by using non-linear least square method, we can get accurate camera motion parameters when $\vec{h}_i^t$ approximates to $\vec{g}_i^t$ with iterations.

B. Motion Estimation

As shown in Equation (1) and (2), solving ego-motion parameters would be complicated. It is difficult to estimate ten degrees of freedom (3 rotational, 3 translational, focal length and 3 parameters from ellipsoid model). Plus, the existing square root in estimation model causes the estimation even more difficult. A non-linear least square method does not consistently provide real solutions and reach global minimum. However, the other optimization method, genetic algorithm (GA), can provide better solutions for highly complex search spaces and can be easily processed in parallel. Thus, we generate initial guesses first. Next, by inputting the initial guesses, GA is used to update camera motion parameters to approximate optimization solutions. Then, the GA solutions are provided as initial values to do non-linear least square optimization. The objective function of GA and non-linear least square method is to minimize the sum of squared residuals between selected feature motion vectors $\vec{g}$ and estimated feature motion vectors $\vec{h}$.

$$S = \sum r^2 = \sum \left\| \vec{g}_i^t - \vec{h}_i^t \right\|. \tag{3}$$

C. Update Background Information

After motion estimation, camera motion parameters and previous frame, I^{t-1}, can be used to estimate current frame. Current frame estimation, I_{est}^t, supplies benchmark for every pixels of current frame, I^t. To do current frame estimation, the movement of every pixel is calculated by using camera motion parameters. Pixel values of current frame estimation are interpolated by the pixel values of previous frame. The current frame estimation is taken as a reference frame and is compared with current frame. The difference values between I_{est}^t and I^t provide information of foreground and background. However, if we just estimate current frame from previous frame, only small amount of foreground information is obtained. That is because only little difference exists between I^t_{est} and I^t, and most of differences appear at edges. In order to get better foreground information, we perform interpolation by using updated frame, I^{t-1}_{upd}, instead of previous frame. The I^{t-1}_{upd} is updated by the feedback of previous foreground mask FGM^{t-1}. The detail of update process shows below.

$$I^{t-1}_{upd}(\zeta) = \begin{cases} I^{t-1}_{est}(\zeta),\ \forall\zeta \in \{FGM^{t-1}(\zeta)=1\} \\ I^{t-1}(\zeta),\ otherwise \end{cases} \tag{4}$$

$$I^t_{est} = Interpolate(I^{t-1}_{upd}, \psi) \tag{5}$$

where ζ is pixel position, ψ is a set of camera motion parameters. By repeating this process, it can make accumulative results effectively.

IV. Information Update and Detection

After motion recovery, feature motion vectors and difference between I^t and I^t_{est} are used to update the probability model for every pixel. The probability model is used for performing foreground and background classification. However, there are many sources of noise existing in the difference of I^t and I^t_{est}: interpolation errors, estimation model imperfections or object movements. We use information obtained from motion recovery to filter out noise sources other than object movements.

A. Refinement

In this section, feature motion vectors are used to select foreground pixel candidates and delete noise. After deleting noise, the result of foreground template, FGM^t_{tmp}, is utilized to update the probability model. First of all, threshold is applied to delete small interpolation errors, which is a source of noises.

$$I^t_{diff\_th} = Threshold_1(I^t - I^t_{est}) \tag{6}$$

Next, for every pixel when its $I^t_{diff\_th}(\zeta)$ is one, we search feature motion vectors within a window around the pixel. Then correlation, γ, of $\vec{g}$ and $\vec{h}$ is calculated for all the feature points within the window. When the expectation of correlation is higher than a threshold, $I^t_{diff\_th}(\zeta)$ of those pixels are treated as noises from model imperfection, and then are discarded.

$$FGM^t_{tmp}(\zeta) = I^t_{diff\_th}(\zeta)\cdot\{1 - Threshold_2(E[\gamma|\tau,\zeta,\psi])\}, \tag{7}$$

where τ is a set of feature motion vectors. Generally, both of noises and feature points appear around edges. Therefore, $E[\gamma]$ is a reasonable choice to provide margin for ellipsoid model imperfection. Every pixel with $FGM^t_{tmp} = 1$ is taken as a foreground candidate and probability model is updated based on FGM^t_{tmp}.

B. Probability Update and Background Detection

A pixel level probability model and its update procedures are presented in this section. Classification rule in [1] is derived for moving object detection by a static camera. With adding camera motion parameters and different probabilities update method, the following relation is developed for moving object detecting by a moving camera.

$$P(b|\nu_\rho,\zeta,\psi) + P(f|\nu_\rho,\zeta,\psi) = 1 \tag{8}$$

where b is background, f is foreground and ν_ρ is pixel value. By using Bayesian decision rule, the pixel is classified as background if probability values satisfy

$$P(b|\nu_\rho,\zeta,\psi) > P(f|\nu_\rho,\zeta,\psi). \tag{9}$$

Using Bayes rule on the posterior probability of background, it follows

$$P(b|\nu_\rho,\zeta,\psi) = \frac{P(\nu_\rho|b,\zeta,\psi)P(b|\zeta,\psi)}{P(\nu_\rho|\zeta,\psi)}. \tag{10}$$

With these equations, we can obtain the classification rule for moving object detection by a moving camera.

$$Backgound \Rightarrow 2P(\nu_\rho|b,\zeta,\psi)P(b|\zeta,\psi) > P(\nu_\rho|\zeta,\psi) \tag{11}$$

The above three probability set need to be updated in each pixel. Because every pixel is moving in every frame, it is required to predict probabilities of every pixel by interpolation. The benefit of interpolation is that the probabilities of pixels contain information of adjacent pixels. During classifying the background with (11), it makes system tolerate small errors caused by model imperfection. To implement Equation (11), we use quantized histogram to replace probabilities. H^t_ν, $H^t_{\nu b}$ and P^t_b present $P(\nu_\rho|\zeta,\psi)$, $P(\nu_\rho|b,\zeta,\psi)$ and $P(b|\zeta,\psi)$ respectively.

$$H^t_\nu(\rho,\zeta) = \begin{cases} H^t_{\nu\_pre}(\rho,\zeta)+1,\ if\ Q[I^t(\zeta)] = \rho \\ H^t_{\nu\_pre}(\rho,\zeta),\ otherwise \end{cases} \tag{12}$$

$$H^t_{\nu b}(\rho,\zeta) = \begin{cases} H^t_{\nu b\_pre}(\rho,\zeta)+1,\ if\ FGM^t_{tmp}(\zeta)=0, \\ \qquad\qquad Q\left[I^t(\zeta)\right] = \rho \\ \alpha\cdot H^t_{\upsilon b\_pre}(\rho,\zeta),\ if\ FGM^t_{tmp}(\zeta)=1, \\ \qquad\qquad Q\left[I^t_{est}(\zeta)\right] = \rho \\ H^t_{\nu b\_pre}(\rho,\zeta),\ otherwise \end{cases} \tag{13}$$

$$P^t_b(\zeta) = \begin{cases} \alpha\cdot P^t_{b\_pre}(\zeta),\ if\ FGM^t_{tmp}=1 \\ \alpha\cdot P^t_{b\_pre}(\zeta)+(1-\alpha),\ otherwise \end{cases} \tag{14}$$

$H^t_{\nu\_pre}$, $H^t_{\nu b\_pre}$ and $P^t_{b\_pre}$ are predicted from H^{t-1}_ν, $H^{t-1}_{\nu b}$ and P^{t-1}_b by interpolation. $Q[\cdot]$ is quantization and α is a scale factor. With updated probabilities and classification rule, background is classified and moving objects can be detected.

V. Experiments

This section presents experiment results obtained from the proposed method and compare them with results of the existing method using multiview geometric constrains [5], which demonstrated significant detecting results on videos in presence of strong parallax. The video streams were captured by a camera in a forward-moving car and the camera was held by a human hand. Because the road is uneven and human hand is unstable, the captured video streams have a lot of sudden irregular movements. The relative movements between objects and camera are complex and change rapidly.

In these two experiments, $threshold_1$ is 50, $threshold_2$ is 0.7, and search window size is 15. The quantization level of histogram is 64, and scale factor α is 0.8. γ is the cosine of the angle between the two vectors.

In video 1, a pedestrian in front of the camera is moving from right to left. In video 2, a car in front of camera is moving forward. Figure 3(a)(c)(e)(g) and Figure 3(b)(d)(f)(h) show experiment results of video 1 and 2 respectively. By applying the proposed method, moving objects in video 1 and 2 are detected and shown in the bounding boxes of Figure 3(a) and 3(b) respectively. Figure 3(c) shows a pedestrian is accurately detected as foreground in video 1 and Figure 3(d) shows a vehicle is detected in video 2. Figure 3(e) and 3(f) show corresponding compared results of the method using multiview geometric constrains. They are results before applying threshold. As the figures show, no matter what threshold is applied (e.g. Figure 3(g) and 3(h)), there will exist more noises in steady background than Figure 3(c) and Figure 3(d). As one can see, experiment results show that the proposed method can accurately detect moving objects and also successfully filters out background noise. After moving objects are detected, the moving objects can be accurately tracked by using the proposed method in [7].

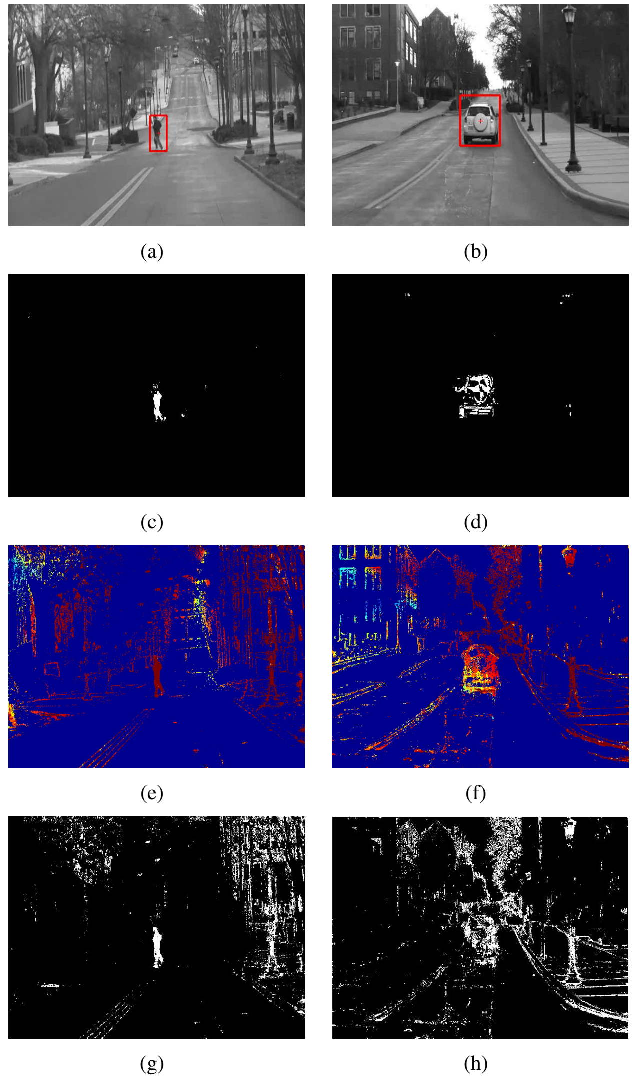


Figure 3. Detecting results of experiments

VI. Conclusions and Discussion

In this paper, we have proposed a novel method to effectively detect moving objects from videos captured by a camera on a moving platform. Experiment results show the propose method has effective detecting performance. And, there is no need to impose initial assumptions or to apply future frame information in the detecting algorithm. Thus, the proposed method could be generally applied to detect moving objects with irregular camera movement and in complex environment. Future research is aimed at detecting moving objects from multiple moving cameras.

References

[1] L. Li, W. Huang, I. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proceedings of the ACM international conference on Multimedia*, 2003, pp. 2–10.

[2] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "Robust Multi-Person Tracking from a Mobile Platform," *Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1831–1846, 2009.

[3] B. Jung and G. Sukhatme, "Detecting moving objects using a single camera on a mobile robot in an outdoor environment," in *International Conference on Intelligent Autonomous Systems*, 2004, pp. 980–987.

[4] K. Yamaguchi, T. Kato, and Y. Ninomiya, "Vehicle ego-motion estimation and moving object detection using a monocular camera," in *IEEE International Conference on Pattern Recognition*, vol. 4, 2006.

[5] J. Kang, I. Cohen, G. Medioni, and C. Yuan, "Detection and tracking of moving objects from a moving platform in presence of strong parallax," in *IEEE International Conference on Computer Vision*, 2005.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision and Image Understanding*, vol. 110, 2008, pp. 346–359.

[7] C. Lin and W. Wolf, "MCMC-based Feature-guided Particle Filtering for Tracking Moving Objects from a Moving Platform," in *IEEE Workshop on Embedded Computer Vision*, 2009.