ISSN 2229-5518

Analysis on Adaptive Moving Objects via Robot Vision Implementations by Detection Techniques

Deepa Kumari, Shamik Tiwari, Deepika Gupta, Raina

Abstract— Building a robot is not only a passion but also a dream for most budding engineers. It is essential to make the concept of visual sensor system used in the field of robotics for identification and tracking of the objects. Identifying moving objects from a video sequence is a fundamental and critical task in many robot-vision applications. A common approach is to perform background subtraction, which identifies moving objects from the portion of a video frame that differs significantly from a background model. There are many challenges in developing a good background subtraction algorithm. First, it must be robust against changes in illumination. Second, it should avoid detecting non-stationary background objects such as moving leaves, rain, snow, and shadows cast by moving objects. Finally, its internal background model should react quickly to changes in background such as starting and stopping of vehicles.

As the name suggests, background subtraction is the process of separating out foreground objects from the background in a sequence of video frames. Background subtraction is used in many emerging video applications, such as video surveillance, traffic monitoring, and gesture recognition for human-machine interfaces. Many methods exist for background subtraction, each with different strengths and weaknesses in terms of performance and computational requirements.

Index Terms— Adaptive human-motion tracking, Background subtraction methods, Detection techniques, Frame difference, Mixture of Gaussians, Robot vision

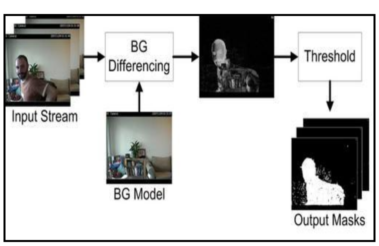
1 Introduction

Numerous BGS algorithms and a number of post-processing techniques that aim to improve the results of these algorithms have been proposed. In this paper, we evaluated several popular, state-of-the-art BGS algorithms and examine how post-processing techniques affect their performance. The experimental results demonstrate that post-processing techniques can significantly improve the foreground segmentation masks produced by a BGS algorithm.

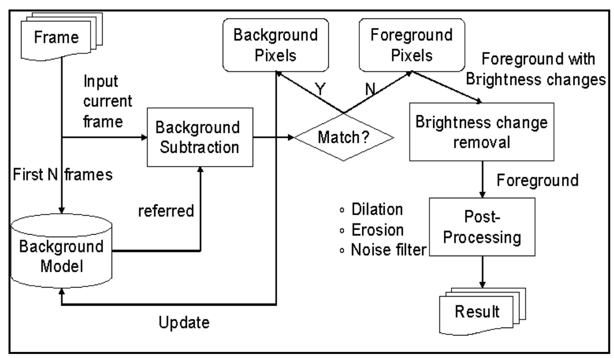
Let F0 be the initial frame and Fi be the consecutive frames, where i= 1 to n. The psuedocode is given below:

If (F0 - Fi) > Th
{
Then accept and process the frame
Else
Reject the frame
}
Where Th refers to Threshold value

Highlight a section that you want to designate with a certain style, then select the appropriate name on the style menu. The style will adjust your fonts and line spacing. Use italics for emphasis; do not underline.



For this evaluation, our goal is to implement three methods that were Computationally efficient enough to make the leap from MATLAB to commercial application, and good representation of background subtraction implementations in today's video applications.



Since background subtraction is being implemented on a wide range of hardware—and thus within a wide range of computational budgets—we chose to implement methods of varying complexity;

- Low-complexity, using the frame difference method,
- Medium complexity, using the approximate median method, and
- High-complexity, using the Mixture of Gaussians method

2.1 Frame Difference

Frame difference is arguably the simplest form of background subtraction. The current frame is simply subtracted from the previous frame, and if the difference in pixel values for a given pixel is greater than a threshold Ts, the pixel is considered part of the foreground.

ISSN 2229-5518

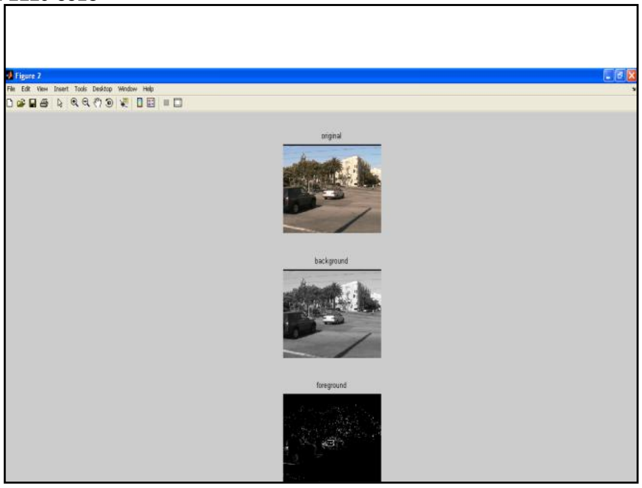


Fig 3. The frame difference method applied to the test video. Non-black pixels are foreground pixels.

1.2 Approximated Median

The approximate median method works as such: if a pixel in the current frame has a value larger than the corresponding background pixel, the background pixel is incremented by one. Likewise, if the current pixel is less than the background pixel, the background is decremented by one. In this way, the background eventually converges to an estimate where half the input pixels are greater than the background, and half are less than the background—approximately the median (convergence time will vary based on frame rate and amount movement in the scene.)

As you can see, the approximate median method does a much better job at separating the entire object from the background. This is because the more slowly adapting background incorporates a longer history of the visual scene, achieving about the same result as if we had buffered and processed N frames. Some trails behind the larger objects (the cars) can be seen. This is due to updating the background at a relatively high rate (30 fps). In a real application, the frame rate would likely be lower (say, 15 fps)

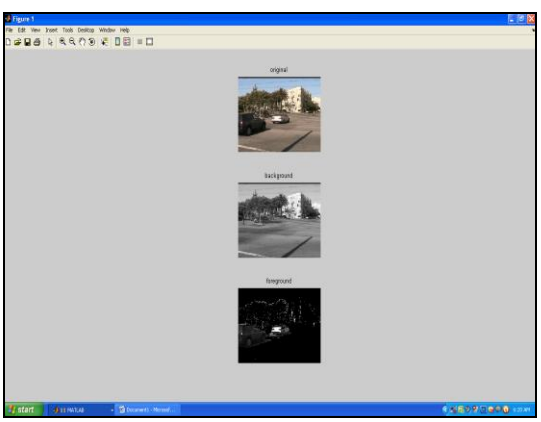
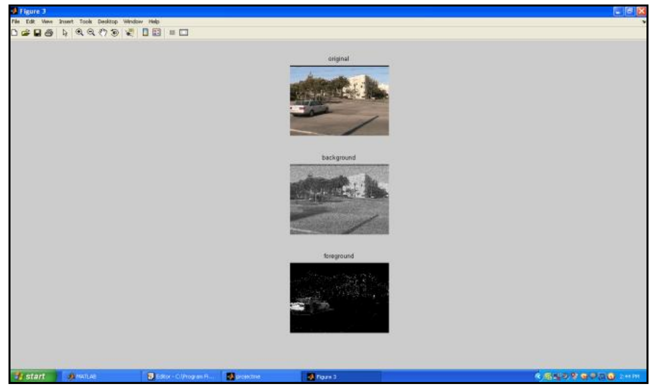


Fig 4. The approximate median method at work on the test video.

1.3 Mixture of Gaussians

Among the high-complexity methods, two methods dominate the literature; Kalman filtering and Mixture of Gaussians (MoG). Both have their advantages, but Kalman filtering gets slammed in every paper for leaving object trails that can't be eliminated. As this seems like a possible deal breaker for many applications, We went with MoG. Also, MoG is more robust, as it can handle multi-modal distributions. For instance, a leaf waving against a blue sky has two modes—leaf and sky. MoG can filter out both. Kalman filters effectively track a single Gaussian, and are therefore unimodal: they can filter out only leaf or sky, but usually not both.



In MoG, the background isn't a frame of values. Rather, the background model is parametric. Each pixel location is represented by a number (or mixture) of Gaussian functions that sum together to form a probability distributions function F.

The mean u of each Gaussian function can be thought of as an educated guess of the pixel value in the next frame—we assume here that pixels are usually background. The weight and standard deviations of each component are measures of our confidence in that guess (higher weight & lower σ = higher confidence). There are typically 3-5 Gaussian components per pixel—the number typically depending on memory limitations.

To determine if a pixel is part of the background, we compare it to the Gaussian components tracking it. If the pixel value is within a scaling factor of a background component's standard deviation σ, it is considered part of the background. Otherwise, it's foreground

2. Implementations

Recursive techniques do not maintain a buffer for background estimation. Instead, they recursively update a single background model based on each input frame. As a result, input frames from distant past could have an effect on the current background model. Compared with non-recursive techniques, recursive techniques require less storage, but any error in the background model can linger for a much longer period of time. Most schemes include exponential weighting to discount the past, and incorporate positive decision feedback to use only background pixels for updating. Some of the representative recursive techniques are described below:

ISSN 2229-5518

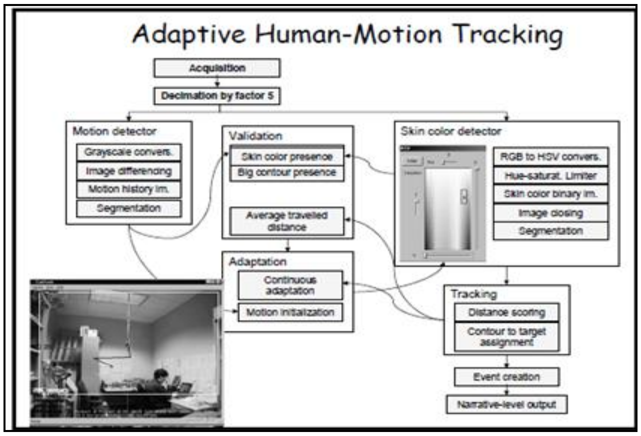


Fig 6. The processes involved in adaptive human-motion tracking.

a) Approximated median filter

Due to the success of non-recursive median filtering, McFarlane and Schofield propose a simple recursive filter to estimate the median . This technique has also been used in back-ground modeling for urban trac monitoring . In this scheme, the running estimate of the median is incremented by one if the input pixel is larger than the estimate, and decreased by one if smaller. This estimate eventually converges to a value for which half of the input pixels are larger than and half are smaller than this value, that is, the median.

b) Kalman filter

Kalman filter is a widely-used recursive technique for tracking linear dynamical systems under Gaussian noise. Many different versions have been proposed for background modeling, differing mainly in the state spaces used for tracking. The simplest version uses only the luminance intensity Karmann and von Brandt use both the intensity and its temporal derivative , while Koller, Weber, and Malik use the intensity and its spatial derivatives . The internal state of the system is described by the background intensity Bt and its temporal derivative Bt′ , which are recursively updated as follows:

c) Foreground Detection

Foreground detection compares the input video frame with the background model, and identifies candidate foreground pixels from the input frame. Except for the non-parametric model and the MoG model, all the techniques introduced previously use a single image as their background models. The most commonly- used approach for foreground detection is to check whether the input pixel is significantly different from the corresponding background estimate:

d) Data Validation

We define data validation as the process of improving the candidate foreground mask based on information obtained from outside the background model.Three main limitations: first, they ignore any correlation between neighboring pixels; second, the rate of adaption may not match the moving speed of the foreground objects; and third, non-stationary pixels from moving leaves or shadow cast by moving objects are easily mistaken as true foreground objects.

The first problem typically results in small false-positive or false-negative regions distributed randomly across the candidate mask. The most common approach is to combine morphological filtering and connected component grouping to eliminate these regions. Applying morphological filtering on foreground masks eliminates isolated foreground pixels and merges nearby disconnected foreground regions. Many applications assume that all moving objects of interest must be larger than a certain size. Connected-component grouping can then be used to identify all connected foreground regions, and eliminates those that are too small to correspond to real moving objects.

When the background model adapts at a slower rate than the foreground scene, large areas of false foreground, commonly known as ghosts", often occur . If the background model adapts too fast, it will fail to identify the portion of a foreground object that has corrupted the background model. A simple approach to alleviate these problems is to use multiple background models running at different adaptation rates, and periodically cross-validate between different models to improve performance . Sophisticated vision techniques can also be used to validate foreground detection. Computing optical ow for candidate foreground regions can eliminate ghost objects as they have no motion . Color segmentation can be used to grow foreground regions by assuming similar color composition throughout the entire object . If multiple cameras are available to capture the same scene at di_erent angles, disparity information between cameras can be used to estimate depth. Depth information is useful as foreground objects are closer to the camera than background . The moving-leaves problem can be addressed by using sophisticated background modeling techniques like MoG and applying morphological filtering for cleanup. On the other hand, suppressing moving shadow is much more problematic, especially for luminance-only video.

4 Conclusion

Although It presents a comparative study of several state-of-the-art background subtraction methods. Approaches ranging from simple background subtraction with global thresholding to more sophisticated statistical methods have been implemented and tested on different videos with ground truth. The time taken to complete an average frame of the data set is shown. The time taken varies from 0.0004 seconds to 12.7196 seconds per frame. Each of the algorithms were run 100 to calculate the average time for each frame to ensure that the operating system did not interfere or influence the speed results, apart from the Mixture of Gaussian.

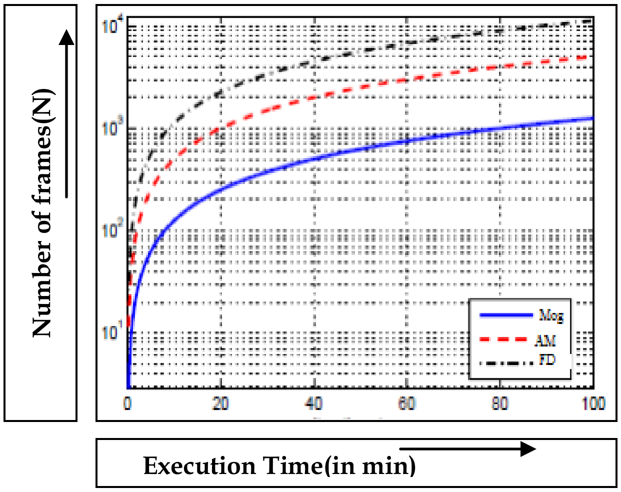


Fig 7. The graph between number of frames and execution time for the analysis of all three approached methods of background subtraction .

The only algorithm capable, within this experiment, of removing a complex background is the frame difference algorithm. This is because the model updates each frame, and checks for movement or motion through a frame, allowing for the algorithm to ignore background motion and identify foreground elements. The problem with this is that it does not handle slow moving foreground objects well. The goal is to provide a solid analytic ground to underscore the strengths and weaknesses of the most widely implemented motion detection methods. The methods are compared based on their robustness to different types of video, their memory requirements, and the computational effort they require. Most of the videos used come from state-of-the-art benchmark databases and represent different challenges such as poor SNR, multimodal background motion, and camera jitter. Overall, it helps to better understand for which type of videos each method best suits but also estimate how, sophisticated methods are better compared to basic background subtraction methods.

REFERENCES

[1] D. Gavrila. Pedestrian detection from a moving vehicle. In ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II, pages 37–49, London, UK, 2000. Springer-Verlag.

[2] Wren C, Azabayejani A, Darrell T.and Pentland A. Pfinder: Real-time Tracking of the Human Boby.IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997 : 780-785

[3] Haritaoglu I, Harwood D and Davis L. W4: Real-Time Surveillance of People and their Activities, IEEE Trans. Pattern Analysis and Machine Intelligence, 2000 : 809-830.

[4] Wang Junqing, Shi Zelin, and Huang Shabai, "Detection of Moving Targets in Video Sequences". (Opto-Electronic Engineering, Dec 2005, pp. 5-8).

[5] Ren Mingwu, and Sun Han, "A Practical Method for Moving Target Detection Under Complex Background".(Computer Engineering, Oct 2005, pp. 33-34).

[6] Milan Sonka, Vaclav Hlavac, and Roger Boyle, "Image Processing, Analysis, and Machine Vision (Second Edition)", Posts & Telecom Press, Beijing, Sep 2003.

[7] Zhang Yunchu, Liang Zize, Li En, and Tan Min, "A Background Reconstruction Algorithm Based on C-means Clustering for Video Surveillance", (Computer Engineering and Application, 2006, pp. 45-47).

[8] H.G. Barrow. Parametric correspondence and chamfer matching: two new techniques for image matching. In International Joint Conference on Artificial Intelligence, pages 659–663, 1977.

[9] D. Cunado, M. S. Nixon, and J. N. Carter, "Using gait as a biometric: via phase-weighted magnitude spectra," in Proc. Int. Conf. Audio- and Video-Based Biometric Person Authentication, 1997, pp. 95–102.

[10] P. S. Huang, C. J. Harris, and M. S. Nixon, "Human gait recognition in canonical space using temporal templates," Proc. Inst. Elect. Eng. (IEE) Vision Image and Signal Processing, vol. 146, no. 2, pp. 93–100, 1999.

[11] A. Johnson and A. Bobick, "A multi-view method for gait recognition using static body parameters," in Proc. Int. Conf. Audio- and Video-Based Biometric Person Authentication, 2001, pp. 301–311.