An Efficient Moving Object Detection using Pixelization Dilate and Different Threshold Setting

Sampath kumar A[1], Saravanan N[2], Sudharsan G[3] , Thangarasu R[4]

[1] Assistant Professor,Department of Computer Science and Engineering,Sri Krishna College of Technology, Coimbatore

[2,3,4] UG Scholor,Department of Computer Science and Engineering,Sri Krishna College of Technology, Coimbatore

Abstract— **Motion tracking is the process of locating a moving object (or several ones) in time. An algorithm analyses the video frames and outputs the location of moving targets within the video frame. The main difficulty in video tracking is to associate target locations in consecutive video frames, especially when the objects are moving fast relative to the frame rate. Here, video tracking systems usually employ a motion model which describes how the image of the target might change for different possible motions of the object to track. In existing system foreground objects with frequent and/or infrequent motions. The feature-matching based local motion stabilization algorithm to identify frequent local motions in the background for reducing false positives in the detected foreground. The proposed system pixelization dilate and different threshold. the motion tracking algorithm is to be able to take video frames as input, and determine the locations of moving objects in the video. The moving objects will have a numeric id, and a bounding rectangle. Threshold of difference because of noise in the video ("fuzziness", "static", "blurriness", etc.), there will be minute difference between almost all of the pixels of the frame and background. We are really only interested in a large difference, which is much more likely to be an actual foreground object. The experimental result shows best performance compare with existing object detection algorithm.**

Index Terms—**pixelization dilate,threshold**

I. Introduction

Videos are actually sequences of images, each of which called a frame, displayed in fast enough frequency so that human eyes can percept the continuity of its content. It is obvious that all image processing techniques can be applied to individual frames. Besides, the contents of two consecutive frames are usually closely related.

Visual content can be modeled as a hierarchy of abstractions. At the first level are the raw pixels with color or brightness information.Further processing yields features such as edges, corners, lines, curves, and color regions. A higher abstraction

layer may combine and interpret these features as objects and attributes. At the highest level are the human level concepts involving one or more objects and relationships among them. Object detection in videos involves verifying the presence of an object in image sequences and possibly locating it precisely for recognition. Object tracking is to monitor an objects spatial and temporal changes during a video sequence, including its presence, position, size, shape, etc. This is done by solving the temporal correspondence problem, the problem of matching the target region in successive frames of a sequence of images taken at closely-spaced time intervals. These two processes are closely related because tracking usually starts with detecting objects, while detecting an object repeatedly in subsequent image sequence is often necessary to help and verify tracking.

Tracking multiple objects in videos is an important problem in computer vision which has wide applications in various video analysis scenarios, such as visual surveillance, sports analysis, robot navigation and autonomous driving. In cases where objects in a specific category are to be tracked, such as people or cars, a category detector can be utilized to facilitate tracking. Recent progress on Multi- Object Tracking (MOT) has focused on the tracking-by detection strategy, where object detections from a category detector are linked to form trajectories of the targets. In order to resolve ambiguities in associating object detections and to overcome detection failures, most of these recent works process video sequences in a batch mode in which video frames from future time steps are also utilized to solve the data association problem. However, such non-causal systems are not suitable for online tracking applications like robot navigation and autonomous driving.

For tracking-by-detection in the online mode, the major challenge is how to associate noisy object detections in the current video frame with previously tracked objects. The basis for any data association algorithm is a similarity function between object detections and targets. To handle ambiguities in association, it is useful to combine different cues in computing the similarity, such as appearance, motion, and location. Most previous works rely on heuristically selected parametric models for the similarity function and tune these parameters by cross-validation, which is not scalable to the number of features and does not necessarily guarantee generalization power of the model.

II. EXISTING SYSTEM

A. *Feature-based object detection*

In feature-based object detection, standardization of image features and registration (alignment) of reference points are important. The images may need to be transformed to another space for handling changes in illumination, size and orientation. One or more features are extracted and the objects of interest are modeled in terms of these features. Object detection and recognition then can be transformed into a graph matching problem.

- Shape-based approaches

Shape-based object detection is one of the hardest problems due to the difficulty of segmenting objects of interest in the images. In order to detect and determine the border of an object, an image may need to be preprocessed. The preprocessing algorithm or filter depends on the application. Different object types such as persons, flowers, and airplanes may require different algorithms. For more complex scenes, noise removal and transformations invariant to scale and rotation may be needed. Once the object is detected and located, its boundary can be found by edge detection and boundary-following algorithms. The detection and shape characterization of the objects becomes more difficult for complex scenes where there are many objects with occlusions and shading.

- Color-based approaches

Unlike many other image features (e.g. shape) color is relatively constant under viewpoint changes and it is easy to be acquired. Although color is not always appropriate as the sole means of detecting and tracking objects, but the low computational cost of the algorithms proposed makes color a desirable feature to exploit when appropriate.

B. *Template-based object detection*

If a template describing a specific object is available, object detection becomes a process of matching features between the template and the image sequence under analysis. Object detection with an exact match is generally computationally expensive and the quality of matching depends on the details and the degree of precision provided by the object template. There are two types of object template matching, fixed and deformable template matching.

- Fixed template matching

Fixed templates are useful when object shapes do not change with respect to the viewing angle of the camera. Two major techniques have been used in fix template matching.

- Image subtraction

In this technique, the template position is determined from minimizing the distance function between the template and various positions in the image. Although image subtraction techniques require less computation time than the following correlation techniques, they perform well in restricted environments where imaging conditions, such as image intensity and viewing angles between the template and images containing this template are the same.

- Correlation

Matching by correlation utilizes the position of the normalized cross-correlation peak between a template and an image to locate the best match. This technique is generally immune to noise and illumination effects in the images, but suffers from high computational complexity caused by summations over the entire template. Point correlation can reduce the computational complexity to a small set of carefully chosen points for the summations.

- Deformable template matching

Deformable template matching approaches are more suitable for cases where objects vary due to rigid and non-rigid deformations. These variations can be caused by either the deformation of the object per se or just by different object pose relative to the camera. Because of the deformable nature of objects in most video, deformable models are more appealing in tracking tasks.

C. *Motion detection*

Detecting moving objects, or motion detection, obviously has very important significance in video object detection and tracking. A large proportion of research efforts of object detection and tracking focused on this problem in last decade. Compared with object detection without motion, on one hand, motion detection complicates the object detection problem by adding objects temporal change requirements, on the other hand, it also provides another information source for detection and tracking.

A large variety of motion detection algorithms have been proposed. They can be classified into the following groups approximately.

- Thresholding technique over the inter frame difference these approaches rely on the detection of temporal changes either at pixel or block level. The difference map is usually binarized using a predefined threshold value to obtain the motion/no-motion classification.
- Statistical tests constrained to pixel wise independent decisions these tests assume intrinsically that the detection of temporal changes is equivalent to the motion detection. However, this assumption is valid when either large displacement appears or the object projections are sufficiently textured, but fails in the case of moving objects that preserve uniform regions. To avoid this limitation, temporal change detection masks and filters have also been considered. The use of these masks improves the

efficiency of the change detection algorithms, especially in the case where some a priori knowledge about the size of the moving objects is available, since it can be used to determine the type and the size of the masks. On the other hand, these masks have limited applicability since they cannot provide an invariant change detection model (with respect to size, illumination) and cannot be used without an a priori context-based knowledge.

- Global energy frameworks The motion detection problem is formulated to minimize a global objective function and is usually performed using stochastic (Mean-field, Simulated Annealing) or deterministic relaxation algorithms (Iterated Conditional Modes, Highest Confidence First). In that direction, the spatial Markov Random Fields have been widely used and motion detection has been considered as a statistical estimation problem. Although this estimation is a very powerful, usually it is very time consuming.

D. *Object tracking using motion information*

Motion detection provides useful information for object tracking. Tracking requires extra segmentation of the corresponding motion parameters. There are numerous research efforts dealing with the tracking problem. Existing approaches can be mainly classified into two categories: motion-based and model-based approaches. Motion-based approaches rely on robust methods for grouping visual motion consistencies over time. These methods are relatively fast but have considerable difficulties in dealing with non-rigid movements and objects. Model-based approaches also explore the usage of high-level semantics and knowledge of the objects. These methods are more reliable compared to the motion-based ones, but they suffer from high computational costs for complex models due to the need for coping with scaling, translation, rotation, and deformation of the objects.

- Boundary-based approaches

Also referred to as edge-based, this type of approaches relies on the information provided by the object boundaries. It has been widely adopted in object tracking because the boundary-based features (edges) provide reliable information which does not depend on the motion type, or object shape. Usually, the boundary-based tracking algorithms employ active contour models, like snakes and geodesic active contours. These models are energy-based or geometric-based minimization approaches that evolve an initial curve under the influence of external potentials, while it is being constrained by internal energies.

- Geodesic active contour models

These models are not parameterized and can be used to track objects that undergo non-rigid motion. In [CC96], a three step approach is proposed which start by detecting the contours of the objects to be tracked. An estimation of the velocity vector field along the detected contours is then performed. At this step, very unstable measurements can be obtained. Following this, a partial differential equation is designed to move the contours to the boundary of the moving objects. These contours are then used as initial estimates of the contours in the next image and the process iterates. More recently, in [BSR99], a front propagation approach that couples two partial differential equations to deal with the problems of object tracking and sequential segmentation was proposed. Additionally, in [GKRR99], a new, efficient numerical implementation of the geodesic active contour model has been proposed which was applied to track objects in movies.

- Region-based approaches

These approaches rely on information provided by the entire region such as texture and motion-based properties using a motion estimation/segmentation technique. In this case, the estimation of the target's velocity is based on the correspondence between the associated target regions at different time instants. This operation is usually time consuming (a point-to-point correspondence is required within the whole region) and is accelerated by the use of parametric motion models that describe the target motion with a small set of parameters. The use of these models introduces the difficulty of tracking the real object boundaries in cases with non-rigid movements/objects, but increases robustness due to the fact that information provided by the whole region is exploited.

III. Proposed Approach

The goal of the motion tracking algorithm is to be able to take video frames as input, and determine the locations of moving objects in the video. The moving objects will have a numeric id, and a bounding rectangle. For example, suppose we have a video whose fifth through twentieth frames shows a person walking across the scene. For each of those frames, the algorithm should give us a rectangle with an ID of 1 (the first and only moving object). In the first frame including the object (frame 5), the bounding rectangle is at the left side of the screen (supposing that the person is walking from left to right). Then in each successive frame, the rectangle coordinates will be further toward the right of the screen.

The main idea of the image analysis is to first know what the scene looks like when there are no moving objects around (we call this the "background" image), and then compare each new frame with the background image to see if there are any foreground objects.

A. *Video Frame Extraction*

Java Media Framework (JMF) provides a unified architecture and messaging protocol for managing the acquisition, processing, and delivery of time-based media data. JMF is designed to support most standard media content types, such as MPEG, QuickTime, and WAV. The video capture with RGB 24bit format only to extract each frame based on JMF code.

B. Video Frame Difference Calculation

The difference between the current video frame and the background image. An example image is shown below. You can see that two people are readily distinguishable. The pixels that are almost exactly the same we will assume to be part of the background; the pixels that are very different are probably part of moving objects. We will look later at how the background image is maintained. The difference is literally calculated as the absolute value of the difference between the red, green, and blue components of the frame's pixel and the background's pixel.

C. Threshold

Threshold of difference because of noise in the video ("fuzziness", "static", "blurriness", etc.), there will be some minute difference between almost all of the pixels of the frame and background. We are really only interested in a large difference, which is much more likely to be an actual foreground object. Therefore, we apply a threshold. A threshold simply says, if A > B then we care about it, otherwise completely ignore the pixel. The resulting image is purely black and white. The white pixels are foreground pixels, and the black are background.

D. Pixelization Dilate

At this point we have our moving objects as blobs of white pixels. But a bunch of white pixels still doesn't quite give us our rectangles yet. We need to group the white pixels together into complete objects, with rectangular bounds. However, most of the time the object's foreground (white) pixels will be rather scattered. For example, notice in the threshold image above the separation between the people and their shadows, and even between their feet and legs. Pixelization Dilate Pixelization Dilate The result of grouping neighboring pixels would be separate objects for shadows, feet, torso, and maybe head. It'd be really nice to keep these poor people together in one piece. Thus our next step in the image processing chain is dilation. Dilation basically makes each pixel spread into the pixels around it so that white pixels close to each other end up touching, and forming a single blob of pixels. There are several ways to do this. I chose a less traditional way because it is probably the fastest method of dilation there is: pixelization. Pixelization, or down-sampling, granulates the image into larger pixels. The particular implementation of pixelization i wrote is specifically for dilation, in that it favors white pixels. For example, for a 4x4 pixel square that will become one large pixelated pixel, if any of the 24 pixels in the box are white, we make the entire large pixel white. Normal pixelization would just average the 24 pixels, or arbitrarily choose one of the pixels for the color.

E. Update Background

The coherent pixel blobs representing the foreground objects. Before we group them into individual objects, we need to take care of our background image. During the course of a video stream, the scene is likely to change slightly. For example, in an outdoor scene, clouds moving overhead can change the overall lighting of the scene. We'd rather not detect cloud shade as giant moving objects. Our background image needs to adapt to these slow or global changes. This is a difficult problem to get working well. For now, chose a simple solution. First, before any motion tracking can be performed, the initial background image must be captured (when no foreground objects are in the scene). That is the purpose of the "Recapture Background" button in my motion tracking program. After that, each pixel of the background is updated slightly according to the new frame's pixel colors, but only the background pixels are updated. That is, only the pixels with corresponding black pixels in the threshold image above will be adjusted. If we included the foreground pixels, then a red object would cause our background image to start turning red wherever the object moved. So instead we ignore whatever pixels are influenced by foreground objects. That we must utilize the threshold image is why the "Update Background" process block comes after the threshold.

F. Object Detector Algorithm

Object Detector (labeling) now that we've taken care of our background image, we are finally ready to locate the foreground objects! The method is commonly called "image labeling." It is the process of labeling each pixel with a number identifying which object it belongs to. In the example image, the purple pixels are all labeled with the same number, and likewise the yellow pixels have their own shared number ID. As these "blobs" are found, their bounding rectangles are also calculated. This gives us our goal output: rectangles with numeric IDs. Labeling is not a trivial algorithm to implement (as I well found out), nor is it widely discussed. specifically, the class FourNeighborBlobDetector. The basic idea is to do a flood-fill, much like that of a basic paint program (you know, the can of paint that's spilling all over the place). A traditional flood-fill algorithm is not meant for real-time video processing, though, so I used a much more efficient algorithm.

In case of BLOB identification process those pixel value which is greater than threshold value is checked for their adjacency pixels. To combine detected pixels to Blobs a test of pixel adjacency needs to be performed. There are two common methods to evaluate adjacent Pixels that is 4-pixel neighborhood. The 4-pixel neighborhood checks for adjacent pixel only on vertical and horizontal axis of the current pixel. 4-pixel neighborhood check for adjacent pixel on vertical, horizontal and diagonal axis of the current pixel. To realize BLOB detection 4-pixel neighborhood is used for adjacency check due to 4-pixel neighborhood is more reliable than 4-pixel neighborhood. The BLOBs with blur depends on the direction of the objects movement. If the motion of the object goes along the horizontal or vertical axis of the screen, the BLOB can show an elliptical shape. In the detection approach, only the elliptical shape has been taken into account.

IV. Experimental Results

The proposed system implementation uses the Java Media Framework (JMF). Each image processing step implements the JMF Effect interface. No attempt was made to use languages and tools that would provide high-performance. In fact, my laptop was not able to keep up real-time for a 640x480 video at 30 frames per second. (This is eight times as much data to process as a 320x240 video at 15 fps.) Here is a list of some future improvements that could be made, and that I hope to make in the future as time permits:

The proposed method and the other methods in comparison are evaluated quantitatively in terms of Recall, Precision, and F1-measure. In this work, we define Recall as the ratio of the overlapped area between the ground truth bounding boxes (or foreground regions) and the detected foreground regions to the area of the ground truth bounding boxes (or foreground regions); and define Precision as the ratio of the overlapped area between the ground truth bounding boxes (or foreground regions) and the detected foreground regions to the area of the detected foreground regions.

Table 1. Comparison of object detection existing vs proposed system

Algorithms	Recall	Precision	F1-Measure
LBP	0.4756	0.5211	0.4973
Mean-shift	0.5673	0.0649	0.1165
ViBe	0.4445	0.5425	0.4886
GRASTA	0.2738	0.6545	0.3861
SC-SOBS	0.5238	0.5935	0.5565
Existing (w/o C-hist)	0.5259	0.6437	0.5789
Existing (w/o LC)	0.5601	0.6662	0.6086
Existing (w/o stabilization)	0.5718	0.6555	0.6108
Background Modeling With Visual Attention Analysis	0.5701	0.6877	0.6234
Proposed	0.6845	0.8156	0.7635

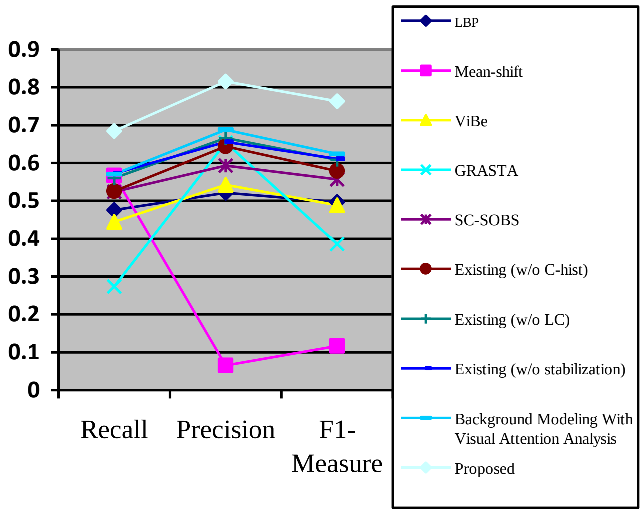

Fig 2. Performance comparison existing vs proposed system

V. Conclusion

The proposed algorithm an efficient tracking moving object targets within the video frame. In existing system foreground objects with frequent and/or infrequent motions. The feature-matching based local motion stabilization algorithm to identify frequent local motions in the background for reducing false positives in the detected foreground. The proposed system pixelization dilate and different threshold. the motion tracking algorithm is to be able to take video frames as input, and determine the locations of moving objects in the video. The moving objects will have a numeric id, and a bounding rectangle. Threshold of difference because of noise in the video ("fuzziness", "static", "blurriness", etc.), there will be minute difference between almost all of the pixels of the frame and background. We are really only interested in a large difference, which is much more likely to be an actual foreground object. The experiment result give better accuracy of proposed system.

REFERENCES

[1] G. Kim and A. Torralba, “Unsupervised detection of regions of interest using iterative link analysis,” in Adv. Neural Inf. Process. Syst., 2009.

[2] X. Liu, L. Lin, S. Yan, H. Jin, and W. Jiang, “Adaptive object tracking by learning hybrid template online,” IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 11, pp. 1588–1599, 2011.

[3] L. Li, W. Huang, I. Y. Gu, and Q. Tian, “Statistical modeling of complex backgrounds for foreground object detection,” IEEE Trans. Image Processing, vol. 13, no. 11, pp. 1459–1472, 2004.

[4] Z. Kim, “Real time object tracking based on dynamic feature grouping with background subtraction,” in IEEE Conf. Comput. Vision Pattern Recognit., 2008.

[5] S. Brutzer, B. Hoferlin, and G. Heidemann, “Evaluation of background subtraction techniques for video surveillance,” in IEEE Conf. Comput. Vision Pattern Recognit., 2011.

[6] T. Bouwmans, “Recent advanced statistical background modeling for foreground detection: A systematic survey,” Recent Patents on Computer Science, vol. 4, pp. 147–176, 2011.

[7] Y. Sheikh and M. Shah, “Bayesian modeling of dynamic scenes for object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 11, pp. 1778–1792, 2005.

[8] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, “Static and moving object detection using flux tensor with split gaussian models,” in CVPR Workshops, 2014.

[9] H. Han, J. Zhu, S. Liao, Z. Lei, and S. Z. Li, “Moving object detection revisited: Speed and robustness,” IEEE Trans. Circuits Syst. Video Technol., vol. 25, no. 6, pp. 910–921, 2015.

[10] S. Kim, D. Yang, and H. Park, “A disparity-based adaptive multihomography method for moving target detection based on global motion compensation,” IEEE Trans. Circuits Syst. Video Technol., 2015.

[11] K. Lin, S.-C. Chen, C.-S. Chen, D.-T. D. Lin, and Y.-P. Hung, “Abandoned object detection via temporal consistency modeling and backtracing verification for visual surveillance,” IEEE Trans. Information forensics and Security, 2015.

[12] X. Shen and Y. Wu, “A unified approach to salient object detection via low rank matrix recovery,” in IEEE Conf. Comput. Vision Pattern Recognit., 2012.

[13] Y. Lin, Y. Y. Tang, B. Fang, Z. Shang, Y. Huang, and S. Wang, “A visual-attention model using earth mover’s distance based saliency measurement and nonlinear feature combination,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 2, pp. 314–328, 2013.