A Pathline-Based Background Subtraction Algorithm

Reinier Oves García[1(✉)], Luis Valentin[1], Carlos Pérez Risquet[2], and L. Enrique Sucar[1]

[1] Computer Science Department, Instituto Nacional de Astrofísica Óptica y Electrónica, Sta. María Tonantzintla, 72840 Puebla, Mexico
{ovesreinier,luismvc,esucar}@inaoep.mx
[2] Universidad Central "Marta Abreu" de Las Villas, Santa Clara, Cuba
cperez@uclv.edu.cu

Abstract. Background subtraction is an important task in video processing and many algorithms are developed for solving this task. The vast majority uses the static behavior of the scene or texture information for separating foreground and background. In this paper we present a novel approach based on the integration of the unsteady vector field embedded in the video. Our method does not learn from the background and neither uses static behavior or texture for detecting the background. This solution is based on motion extraction from the scene by plane-curve intersection. The set of blobs generated by the algorithm are equipped with local motion information which can be used for further image analysis tasks. The proposed approach has been evaluated with a standard benchmark with competitive results against state of the art methods.

Keywords: Background subtraction · Motion detection · Optical flow · Vector field integration

1 Introduction

Video analysis has become an active research topic in Computer Science with application in robotics, video surveillance, pose estimation, human computer interaction, etc. One of the first step in all these video applications is the background/foreground subtraction of the scene. Nevertheless, performing this task, in an automatic way, remains an important and difficult challenge.

A common way to perform background subtraction is to train a model with images while their are appearing, and then use the last one for testing if it is adjusted to the model. Another simpler strategy is to compute the difference between the current frame and the last one. Modern change detection algorithms are generally split into three parts [16]: first, a background model of the scene is created and periodically updated by analyzing frames from the video sequence. Then, preliminary foreground/background segmentation labels *(or probabilities)* are assigned to all pixels of every new frame based on their similarity to the

J.A. Carrasco-Ochoa et al. (Eds.): MCPR 2017, LNCS 10267, pp. 179–188, 2017.
DOI: 10.1007/978-3-319-59226-8_18

model. Finally, regularization is used to combine information from neighboring pixels and to make sure uniform regions are assigned homogeneous labels. Because of the wide range of possible scenarios and the parameters that control model sensitivity, the foreground/background segmentation can be very difficult to obtain in some cases, especially when illumination variations, dynamic background elements and camouflaged objects are all present in a scene at the same time.

In this paper we present a novel background subtraction algorithm based on motion change detection through optical flow integration. Our algorithm computes a set of trajectories from the velocity field described by the optical flow. The resulting set of trajectories are used to extract the background/foreground of the video stream. Additionally, they provide a model of the foreground objects characterized by their punctual *velocity, curvature and torsion* which could represent useful features for several applications, such as video surveillance.

2 Vector Fields

Vector fields are classified by its time dependency in two main groups, steady and unsteady vector fields. Steady vector fields represent time-independent flows *(e.g. laminar flows)* while the unsteady or time-dependent *(e.g. turbulent flows)* case are the most complex and represent the changes of the flow over the space-time domain.

An unsteady vector field in $\mathbb{R}^n$ is mathematically represented by a continuous vector-valued function $V_{time}(X, t)$ on a manifold M where the spatial component is represented by $X \in \mathbb{R}^{n-1}$ and temporal dimension is given by $t \in \mathbb{R}$. For such a reason V_{time} is modeled as a system of Ordinary Differential Equations *(ODE)* $\frac{dX}{dt} = V_{time}(X, t)$. In other words V_{time} is a map $\varsigma : \mathbb{R}^{n-1} \times \mathbb{R}$ with initial conditions $X(t_0)$ where its solutions are called characteristic curves, tangent curves or orbits [18,24].

On the other hand a video can be seen like a map $V : \mathbb{R}^2 \times \mathbb{R}$ where all frames $F_i \in \mathbb{R}^2$ evolve in time $T \in \mathbb{R}$. The color information represented by each frame F_i simulates an apparent movement over the temporal domain from F_i to F_{i+1}. The pattern of motion at each pixel on the scene is computed through the partial derivatives of F [1,4]. The optical flow is known as a 2-dimensional steady vector field densely sampled over the frame space.

Massless particles trajectories are computed integrating V_{time} in space as well as in the time-space dimension. These trajectories are classified in four different types, depending on the integration space; all of them differ for the unsteady case, while for the steady case all trajectories coincide. For our purpose we only have to consider *pathlines* which represent the movement of massless particles over the space–time domain. The arc-length parametrization of a pathline P starting at point $p(x(t_0))$ is defined as:

$$p(x(t_0)) = x_0 + \int_{t_0}^{t_n} V(x(t), t)dt \qquad (1)$$

As trajectories model the motion of the flow, other local characteristics are defined over V_{time}. Curvature, torsion and instantaneous velocity are intrinsic magnitudes of the vector field which are defined at each non critical point. These properties can be assigned to each curve in every visited point [18,23].

3 Related Work

Background Subtraction (BS) is the first step applied to detect regions of interest in a video stream. It consists on creating a background model, so that it is possible to discriminate between the static elements and the moving objects in a video sequence. The simplest way to do this is to subtract the current image from a reference image. However, this method is susceptible to illumination changes. Finding a good reference image for BS is complicated due to the dynamic nature of real-world scenes.

Instead of using a fixed reference image for *BS*, some of the first adaptive methods used pixel intensity average and filtering to create a parametric background model. This kind of approach is robust to noise and can slowly adapt to global illumination variations, but is generally inadequate against shadows. Gaussian mixture models [5,17] were introduced to solve the latter problem, this approach can handle a dynamic background by using a mixture of Gaussian probability density functions over the color intensities of the image. This approach remains to this day a very popular solution.

BS based on non-parametric models have also been proposed [3]. Unlike parametric models, these rely directly on the local intensity of observations to estimate background probability density functions at individual pixel locations.

Another approach that deals with a multimodal background model is the so-called codebook [8,25]. This method assigns to each background pixel a series of key color values (codewords) stored in a codebook. These codewords will take over particular color in a certain period of time. *BS* has also been achieved by other methodologies. In [15] the authors improve the subtraction by superpixels and Markov Random Fields; [7] proposes an approach based on region-level, frame-level [12,20] or hybrid frame/region-level [11,19] comparisons to explicitly model the spatial dependencies of neighboring pixels.

The use of methods based on artificial neural network have achieved good results [9,10] on different scenarios without prior knowledge. However, this kind of approach requires a very large training period. Hybrid approaches have also been proposed. For instance, in [22] the authors combine flux tensor-based motion detection and classification results from a Split Gaussian Mixture Model, and use object-level processing to differentiate static foreground objects from ghost artifacts.

In contrast with previous work, the proposed approach focuses on detecting the foreground based on motion detection through optical flow; once the foreground is detected, the rest of the image is the background. This does not require a training stage, and as a byproduct gives motion information on the moving blobs, which could be used for further analysis.

4 A Pathline-Based Approach for Background Subtraction

The core idea of this method is based on the computation of pathlines behind the unsteady vector field embedded in the video. The flow field construction works over the optical flow extracted in a few continuous sequence of frames. The methodology used in this work is shown in Fig. 1 and described below.

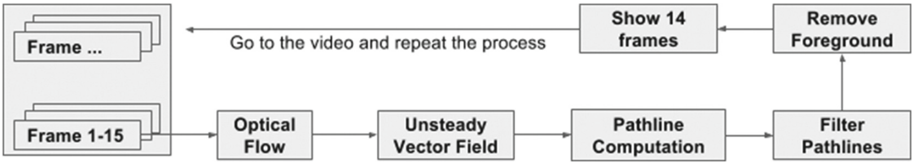


Fig. 1. Methodology of the pathline-based background subtraction strategy over online and offline videos.

Optical Flow: In order to compute the optical flow [4] we take a segment of 15 frames from the video. We select only 15 frames for reducing computational time and drifting [21], but the algorithm is able to work with as many frames as required. From the optical flow we compute the steady vector field *(non-time-dependent vector field)* over each pair of consecutive frames [4] obtaining a list of N steady vector fields.

Unsteady Vector Field: Over the list of steady vector field is generated in the previous step we compute a 2-dimensional unsteady vector field. This field is transformed into a 3-dimensional steady vector field with time implicitly in the *z-direction*. To accomplish this task we generate a grid in R^3 and put in each position $p(x, y, z)$ of the grid the vector $\boldsymbol{v}(u, v, 1)$ [24]. This dimensionality augmentation in Eq. 1 facilitates the integration of the vector field and increases the accuracy and speed of the numerical integration schema. The new dimension incorporated to each vector $(z = 1)$ integrates over t which represents the temporal component of the data set.

Pathline Computation: The integration of the vector field with initial conditions produces a set of trajectories called pathlines. These pathlines describe the motion of the massless particle over the space-time domain. Here we sample the grid in a regular way for producing a set of curves parametrized by time. This set of curves represents the motion of color information *(pixels)* over the selected frames (See Subsect. 5.1).

Pathline Filtering: At the same time that pathlines are being generated, a filtering process is taking place. The orientation of this kind of trajectories is important for its later discrimination. Background pixels move more in *z-direction* than in the other two. For that reason we seed a pathline anywhere in our sampled grid but we restrict its travel over the *space-time* taking into account its orientation (See Subsect. 5.2).

Removing Foreground: The filtered pathline bundle contains only trajectories associated with the motion of the scene represented by the set of frames. These trajectories were generated over a continuous space and hence they store inter-pixel information as well inter-frame information. A discretization process of the curve takes place in this step as well as the plane-curve intersection for detecting motion in each frame. Each curve intersects at least one frame in only one point. These intersection points are taken as foreground points (See Sect. 6 for more details).

In each frame, the coordinates of moving blobs (foreground) is given by the intersection points of the pathline bundle with each frame. Once a set of (15) frames (chunk) is processed, the next video segment is analyzed. It is important to note that N frames produce only $N-1$ steady vector fields. For this reason we have to start the next set taking one value before the previous set ends. This consideration guaranties that motion never ends between each pair of video segments. It is possible that the last set does not fulfill the size of our sample and we recommend for that cases to join the residual set with the previous one so that no motion information is lost.

5 Tracking Motion over an Unsteady Vector Field

Considering pixels as massless particles moving over the flow described by the embedded unsteady vector field we can track their information by means of integrating the ODE system V_{time} with initial conditions over the sampled grid. For our purpose we select Runge-Kutta [2] of 4^{th} order to integrate the ODE system. This method is applied for CFD by the *Computational Fluids Dynamics* community for its accuracy and speed. Our image grid is equipped with a trillinear interpolation schema [14]. This interpolation algorithm allow us to reach a good approximation of trajectories in points where the field is not explicitly defined.

5.1 Dense Pathline Computation

For the sake of capturing the motion in more detail a dense set of pathlines is computed [13]. The dense characteristic of the set is given by the seeding strategy used by the integration algorithm. The grid that contains V_{time} is sampled in the x and y directions by a step factor that separates one spatial initial condition from another. For the temporal domain, we sample the grid one-to-one, so that no temporal information is lost.

Once the initial condition is set, the stop condition must be defined. Given a pathline P starting at $p_0(x_0, y_0, t_0)$ we advect the flow until the next computed point fulfills one of these three conditions: p_n goes out of the domain, p_n reaches a critical point or the vector from p_n to p_{n-1} is almost normal to the XY plane. For instance, suppose we have a video of 340 × 240 and a duration of 10 frames. A video like this produces a *3-dimensional* steady vector field V_{time} over the domain 240 × 480 × 10. (It is evident that beyond that limits V_{time} it is not defined as well as in critical points ($||V_{time}(p)|| = 0$) [6].

5.2 Filtering Pathlines by Their Orientation

In the previous section it is omitted the third stop rule of the integration algorithm because of its importance. The first two rules only guaranty that every evaluated point in V_{time} is defined and are mandatory to use them while in vector field integration. For our context we add a third rule to isolate automatically background and foreground.

As we mention earlier, background information can be approximated by a straight line parallel to the *z-axis*. For such a reason our third rule states that given two consecutive points, p_i and p_{i+1}, of the same curve, the distances are defined as: $x_{distance} = |x_i - x_{i+1}|$, $y_{distance} = |y_i - y_{i+1}|$ and $z_{distance} = |z_i - z_{i+1}|$. The orientation condition is reached when the $z_{distance}$ is larger than a threshold (three in our experiments).

Identifying this kind of points while integrating reduces dramatically the execution time of the algorithm and avoids regions where the texture of the objects in the scene coincides with the background texture.

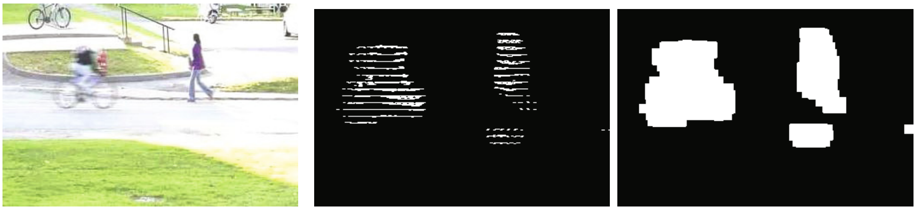
6 Background Subtraction

At this point all the pathlines are filtered and the resulting bundle only represents the motion of the objects over a continuous space. A good representation of the motion at each frame is achieved by means of a discretization of the curve space in the *z-direction.* This process is given by moving a plane over the frame space and computing the intersections of the curves with the plane for each movement.

Given the plane equation $F = (1, 0, t) + \lambda(0, 1, t) + \beta(0, 0, t)$ and a pathline $P(s)$ parametrized by its *z-component* we can compute the intersection point $I(x, y, t)$ where P crosses F, $t \in [1..15]$. The foreground in frame F at time t is defined by the set of points I generated by the intersection of the pathline bundle with the image plane.

The point set generated by the intersection of pathlines with the list of frames conforms the motion of the scene but it is a sparse representation of the motion. To eliminate "holes" in the frame a morphological operation –dilatation– is applied. This operation is accomplished by means of drawing a square at each foreground point in each frame with a size of $GridSpaceX/2$. This strategy guaranties the connection between all points in each moving blob producing a continuous region (see Fig. 2(c)).

It is evident that blobs generated by this algorithm are larger than the original moving objects. For reducing this information we compute the frame difference between F_t and F_{t-1}, in order to generate a mask that contains the original moving objects but also noise. Using a bitwise *AND* operation between the resulting image from our approach and the image produced by frame difference it is possible to eliminate noise and outliers, resulting in a more robust blob *(See Fig. 3 last column).*



(a) Original

(b) Point intersect

(c) Morphological dilatation

Fig. 2. Visual representation of the plane-curve intersection in the *Change Detection Dataset* (frame 475): (a) Original image. (b) Points where the pathline bundle intersects the frame. (c) Morphological dilatation applied over the point set

7 Experimental Results

To evaluate our approach, we use the benchmark introduced for the 2012 CVPR Workshop on Change Detection. This data set offers different types of realistic condition scenarios, along with an accurate ground truth data. From this benchmark, we are using four categories: highway, office, pedestrians and PETS2006; evaluating our method with 6049 images. This data set has been used to test several state-of-the-art background subtraction methods. Figure 3 illustrates an example from each category, with the results of the proposed background subtraction method as well as another state-of-the-art method.

For evaluation we used two sets of outdoor and two sets of indoor image sequences. Considering that our method is based on motion detection, the outdoor sets present a significant challenge due to the moving background elements. Experimental results are presented using precision and recall *(see Fig. 4)*. Besides, we are comparing these results with another two state of the art methods [16,17], and also with the simplest background subtraction, frame difference (FD). All our experiments were done on an Intel i7 CPU at 3.3 GHz, using the OpenCV library.

Figure 4(a) depicts a plot with the precision metrics for each category of the data set. We observe that our method obtains *high* values for every category. Although it does not have the highest precision value, it is about three times faster than the one that gets the best results. In Fig. 4(c) we show the time needed to process each one of the data set categories.

On the other hand, we don't get the best results in the recall metric, as it is shown in Fig. 4(b). This is because our approach is based on motion detection, and when an object stops is not possible to detect it. However, this background subtraction strategy not only separates moving objects from the background, but also provides relevant information about the moving objects, for instance, velocity, curvature and torsion. These local properties are presented intrinsically in the unsteady vector field embedded in the video and are computed during integration. Our resulting image is modeled as a *3-channel* matrix where the

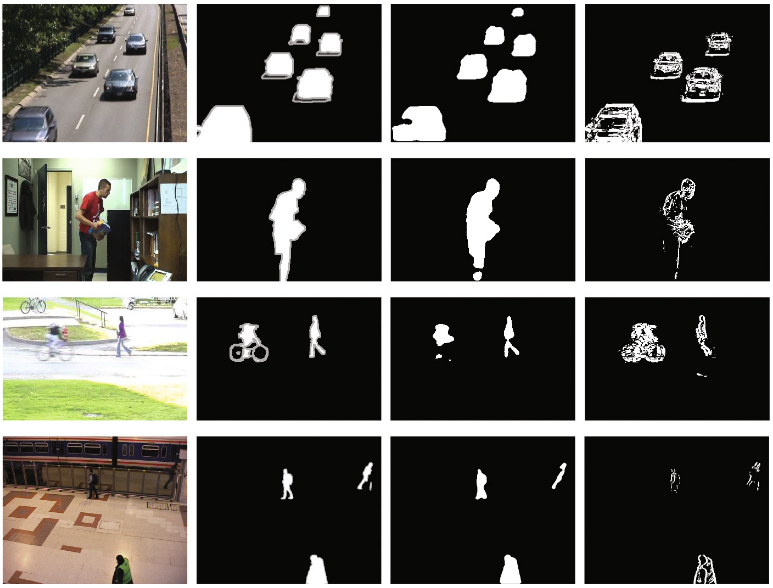
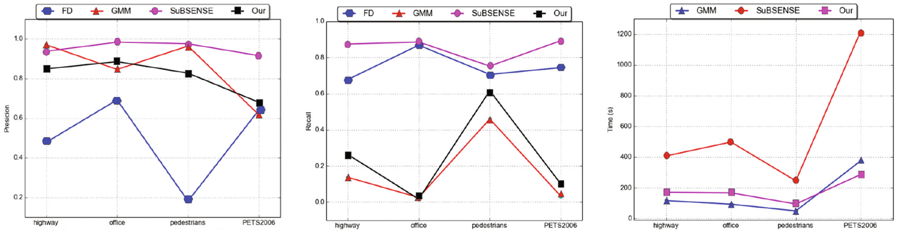


Fig. 3. Background subtraction results. The first column shows a set of images from the data set. In the second column we can see the ground truth. The third column depicts the results of SuBSENSE [16], and in the last column we can see the results of our method.



(a) Precision plot.

(b) Recall plot.

(c) Time.

Fig. 4. Precision (a) and Recall (b) plots for each category in the data set. Blue (Hexagon), red (Triangle) and magenta (Circle) lines represents the Frame Difference, Gaussian Mixture Model and SuBSENSE methods, respectively; while the black (Square) line is for our method. We also show the time (c) required by every method to process each category. (Color figure online)

first channel stores the instantaneous velocity, the second channel stores the curvature and the last one stores the torsion. In that way, our blob is more informative than classical blobs.

8 Conclusions

In this paper we presented a novel approach for background subtraction based on the integration of the unsteady vector field embedded in the video. Our method is not based on the steadiness of the scene and neither on texture information; but on motion detection based on optical flow. For such a reason this proposal is particularly useful for scenarios where the background change constantly. Experimental results in a benchmark data set that includes dynamic scenarios, demonstrate that our method is efficient and competitive with other state-of-the-art techniques; in particular it is able to capture all moving blobs no matter how small they are. Besides the blob identification, our method is able to compute local scalar quantities *(velocity, curvature, torsion)* that increase the blob information, providing useful features for further processing, such as object tracking and classification.

Acknowledgments. This work was supported in part by FONCICYT (CONAYT and European Union) Project SmartSDK - No. 272727. Reinier Oves García is supported by a CONACYT Scholarship No.789638

References

1. Beauchemin, S.S., Barron, J.L.: The computation of optical flow. ACM Comput. Surv. (CSUR) **27**(3), 433–466 (1995)
2. Butcher, J.: Runge-Kutta method. Scholarpedia **2**(9), 3147 (2007)
3. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 751–767. Springer, Heidelberg (2000). doi:10.1007/3-540-45053-X_48
4. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Heidelberg (2003). doi:10.1007/3-540-45103-X_50
5. Friedman, N., Russell, S.: Image segmentation in video sequences: a probabilistic approach. In: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, UAI 1997, San Francisco, CA, USA, pp. 175–181. Morgan Kaufmann Publishers Inc. (1997)
6. Helman, J.L., Hesselink, L.: Visualizing vector field topology in fluid flows. IEEE Comput. Graph. Appl. **11**(3), 36–46 (1991)
7. Iodoin, J.-P., Bilodeau, G.-A., Saunier, N.: Background subtraction based on local shape. CoRR, abs/1204.6326 (2012)
8. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground-background segmentation using codebook model. Real-Time Imaging **11**(3), 172–185 (2005)
9. Maddalena, L., Petrosino, A.: A self-organizing approach to background subtraction for visual surveillance applications. IEEE Trans. Image Process. **17**(7), 1168–1177 (2008)
10. Maddalena, L., Petrosino, A.: The SOBS algorithm: what are the limits? In: CVPR Workshops, pp. 21–26. IEEE Computer Society (2012)
11. Nonaka, Y., Shimada, A., Nagahara, H., Taniguchi, R.: Evaluation report of integrated background modeling based on spatio-temporal features. In: CVPR Workshops, pp. 9–14. IEEE Computer Society (2012)

12. Oliver, N.M., Rosario, B., Pentland, A.P.: A Bayesian computer vision system for modeling human interactions. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 831–843 (2000)
13. Peng, Z., Laramee, R.S.: Higher dimensional vector field visualization: a survey. In: TPCG, pp. 149–163 (2009)
14. Rajon, D.A., Bolch, W.E.: Marching cube algorithm: review and trilinear interpolation adaptation for image-based dosimetric models. Comput. Med. Imaging Graph. **27**(5), 411–435 (2003)
15. Schick, A., Bäuml, M., Stiefelhagen, R.: Improving foreground segmentations with probabilistic superpixel Markov random fields. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, pp. 27–31. IEEE, June 2012
16. St-Charles, P.-L., Bilodeau, G.-A., Bergevin, R.: Subsense: a universal change detection method with local adaptive sensitivity. IEEE Trans. Image Process. **24**(1), 359–373 (2015)
17. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: CVPR, pp. 2246–2252. IEEE Computer Society (1999)
18. Theisel, H., Weinkauf, T., Hege, H.-P., Seidel, H.-P.: Topological methods for 2d time-dependent vector fields based on stream lines and path lines. IEEE Trans. Vis. Comput. Graph. **11**(4), 383–394 (2005)
19. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. In: ICCV, pp. 255–261 (1999)
20. Tsai, D.-M., Lai, S.-C.: Independent component analysis-based background subtraction for indoor surveillance. IEEE Trans. Image Process. **18**(1), 158–167 (2009)
21. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3551–3558 (2013)
22. Wang, R., Bunyak, F., Seetharaman, G., Palaniappan, K.: Static and moving object detection using flux tensor with split Gaussian models. In: Proceedings of IEEE CVPR Workshop on Change Detection (2014)
23. Weinkauf, T., Theisel, H.: Curvature measures of 3d vector fields and their applications (2002)
24. Weinkauf, T., Theisel, H.: Streak lines as tangent curves of a derived vector field. IEEE Trans. Vis. Comput. Graph. **16**(6), 1225–1234 (2010)
25. Mingjun, W., Peng, X.: Spatio-temporal context for codebook-based dynamic background subtraction. AEU Int. J. Electron. Commun. **64**(8), 739–747 (2010)