Research Article

Tracking Animated Object in Video via Image Content Modeling

Mehdi Taghizadeh [a], Maryam Mahnaie [b*], Amir Peikar [b]

[a] Faculty member of Electrical Engineering, Kazerun Branch, Islamic Azad University, Kazerun, Iran.
[b] Department of Food and Drug, Yasuj University of Medical Sciences, Yasuj, Iran.

Article History: Received: 5 April 2021; Accepted: 14 May 2021; Published online: 22 June 2021

Abstract: Tracking and recognizing the target using a film or a sequence of consecutive images is one of the most significant fields of research in machine vision. In the current research, a spatial-temporal content model is used to track images in the video. In this method, at first, a Spatio-temporal content model between the target object and the surrounding spatial background is learned based on the spatial relationships of a scene. The next time stage is performed using an assurance mapping design in tracking that can integrate Spatio-temporal content information and estimate the location of the target object by maximizing the assurance mapping. Likewise, content is used to help track moving objects in complex scenes and to try to reduce side effects and background interference. The simulation results are offered for fixed and moving camera modes, and lastly, the range of parameters designed to track the moving object in both modes is expressed. The simulation results reveal the speed, power, and accuracy of the suggested algorithm.

Keywords: Video processing, Moving object tracking, Spatial-temporal content, Assurance mapping

1. Introduction

Due to the wide range of applications such as motion analysis, activity detection, monitoring, and human-computer interaction, and many others, image tracking is one of the most prevalent research topics [1]. In image tracking, by receiving consecutive images of a scene during the movement time, the visual system seeks to detect objects in the scene and determine their directions [2]. Motion recognition and separation of objects from the background is the first step in tracking moving objects. Separation of objects from the background can be done based on spatial, temporal, or Spatio-temporal information. At large, there are two types of visual information presentation, which are object-based presentation and image-based presentation. Object-based presentation depends on the separation of the object from the context, and presentation based on the whole image is directly derived from the image [3].

Research has also been done in this area. In [4], the Kalman filter is used to track vehicles in an adaptive framework. In [5], the extended Kalman filter was used to estimate the three-dimensional motion path of an object using two-dimensional image motions. In [6] they used a particle filter to track faces and cars. In [7], the authors constructed the object model by averaging the colors and pixels within the rectangular area and to reduce the computational complexity in the next frame, searched for the object in the eight neighborhoods around the object. In [8], the authors used a calculated weighted histogram of an elliptical area to represent the object, and instead of conducting a global search, they used the mean shift process [9] to locate the object. In another paper, he used a common distance-color histogram (instead of a single color histogram) with the mean shift approach.

The authors tried to solve this problem with a new definition of object tracking. In this way, the authors considered the problem of object tracking as the separation of an object from a field close to the object (which is located around the object and not the whole field). Some researchers have considered object tracking as a two-class classification problem, and by constructing a time-varying discriminant function, they have separated the object and background property distributions in each frame. Some authors first used the color approach and properties to separate the object from its surroundings, then used two separate neural networks. Some researchers have used Support Vector Machine classifiers to track vehicles backward.

Researchers have also used information on light flux and contour velocity to determine a range for active contour activity in each frame. In conventional content-based methods, to find reliable regions, key points around the target are first extracted to help locate the main target. Then, a descriptor sample is used to show these consistent regions. Consequently, heavy computational operations are required to display and find consistent regions. In addition, regarding the sparse nature of key points (points with distance and far from each other), some consistent regions that are useful for finding the target position may be discarded. This study tries to inspect these problems and seek to address them in such a way that all local regions around the target are considered as potentially consistent regions and the correlation of motion between objects and local content are learned in consecutive frames, by the Spatio-temporal content model.

According to what was mentioned, the need for a way is evident to track moving objects in a sequence of images. In this research, a spatial-temporal content model is used to track images in the video. In this model, all local regions around the target are considered as potential and consistent regions, and the correlation of motion between objects and their local contents in successive frames is learned by the spatial-temporal content model.

Then the Spatio-temporal relationships between the target and its local contents, stimulated by the human visual system, are modeled, and content is used to help resolve ambiguities in complex scenes efficiently and effectively.

2. Proposed tracking algorithm

The tracking problem is posed by calculating an assurance map, and the best target location is obtained by maximizing the potential target location function. Fast Fourier transform was used for fast learning and diagnosis. Implementation is done in MATLAB. In visual tracking, a local texture consists of a target object and the instantaneous background around it in a specific area. Therefore, there is a robust spatial and temporal relationship between local scenes containing the object in successive frames. The following figure shows the proposed algorithm.

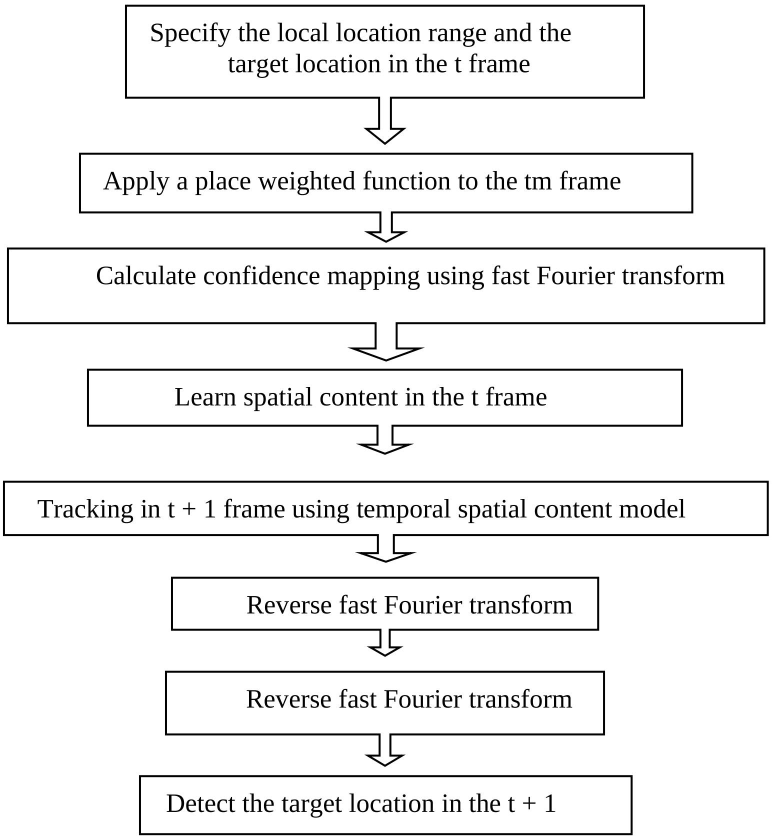

Specify the local location range and the target location in the t frame
Apply a place weighted function to the tm frame
Calculate confidence mapping using fast Fourier transform
Learn spatial content in the t frame
Tracking in t + 1 frame using temporal spatial content model
Reverse fast Fourier transform
Reverse fast Fourier transform
Detect the target location in the t + 1

Figure 1: Block diagram of the proposed tracking algorithm

It is supposed that the target location in the first frame is quantified either manually or by some target detection algorithms. In the t*th* frame, the spatial content model $h_t^{sc}(x)$ is learned and it is used to update the Spatio-temporal content model, $h_{t+1}^{stc}(x)$ and to detect the target location in the t + 1 M. When the frame (T + 1) is reached the local content area $\Omega_c(x_t^*)$ based on the tracked location x_t^*in the t frame is removed, and thus the corresponding content attribute set $h_{t+1=}^{c}\{c(z) = (I_{t+1}(z), z) | z \in \Omega_c(x_t^*)\}$ is formed. The target location x_{t+1}^* * is determined in the t + 1 frame by maximizing the assurance mapping:

$$x_{t+1}^* = \text{argmax } c_{t+1}(x), x \in \Omega_c(x_t^*) \qquad (1)$$

Where $c_{t+1}(x)$ is displayed as follows:

$$c_{t+1}(x) = F^{-1}(F\big(\, H_{t+1}^{stc}(x)\big) \odot F\big(I_{t+1}(x)\omega_{\sigma_t}(x - {x_t}^*)\big)) \qquad (2)$$

Which is calculated from the above formulas.

The spatial-temporal content model is updated as follows:

$$H_{t+1}^{stc} = (1 - \rho)H_t^{stc} + \rho h_t^{sc} \qquad (3)$$

Where ρ is the learning parameter and $h_t^{sc}$is the spatial content model calculated in the t frame. This equation is a time filtering method that can be easily observed in the frequency domain.

Where $H_\omega^{stc} \triangleq \int H_t^{stc} e^{-j\omega t} dt$ is the Fourier transform of time H_t^{stc} and similar to h_ω^{sc}. The time filter F_ω is formulated as follows:

$$F_\omega = \frac{\rho}{e^{j\omega} - (1-\rho)} \qquad (4)$$

Where j represents an imaginary unit. F_ω is a low-pass filter. Consequently, the proposed spatial and temporal content model can effectively filter the image noise introduced by the appearance changes, thus leading to more stable results.

The target location in the current frame is found by maximizing the assurance mapping obtained from the weighted content area around the target location in the preceding step. Though, the target scale often changes over time. Consequently, the scale parameter σ in the weight function ω_σ needs to be updated. The proposed scale update plan is as follows:

$$\begin{cases} s'_t = \sqrt{\frac{c_t(x_t^*)}{c_{t-1}(x_{t-1}^*)}}, \\ \bar{s}_t = \frac{1}{n}\sum_{i=1}^{n} s'_{t-i}\,, \\ s_{t+1} = (1-\lambda)s_t + \lambda \bar{s}_t, \\ \sigma_{t+1} = s_t \sigma_t \end{cases} \qquad (5)$$

Where $c_t(.)$ is the assurance mapping previously calculated, and s'_t is the estimated scale between two consecutive frames. To avoid oversensitive adaptation and noise reduction introduced by the estimation error, the scale parameter value is obtained for the estimated target, s_{t+1} through filtering, where $\bar{s}_t$ is the mean scale estimated from N consecutive frames, and λ> 0 is a constant filter parameter.

3. Simulation results

First, the video to be reviewed must be converted into multiple and consecutive photos and saved. Then, regarding the size of the pixel space in each image, the initial value was considered as the location of the moving object and a window for the surrounding space. An assurance map should be drawn based on the image within the specified range. As can be seen in the following formula, the two parameters α and β are used to weigh the window content. The following figure shows the image of a normalized window weighted with different αs and βs.

$$c(x) = P(x|o) = be^{-\left|\frac{x-x^*}{\alpha}\right|^{\beta}} \qquad (6)$$

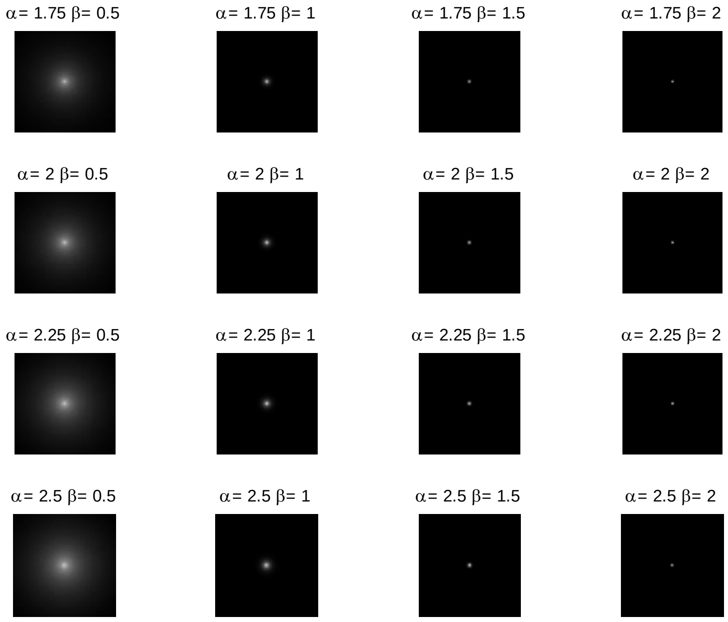


Figure 2: Display of a Gaussian window with different αs and βs in the location domain

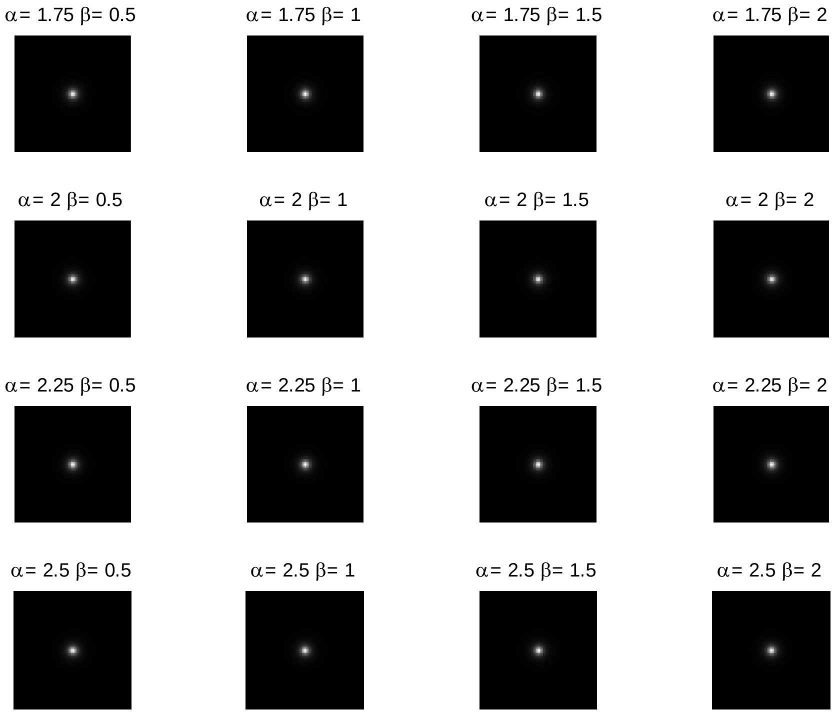


Figure 3: Gaussian window display with different αs and βs in the frequency domain

It is observed that more *β*s cause more distribution around the center and put more points in the assurance range. The variable α represents the speed of distancing from the center. Based on the above figures and the values suggested in the paper [10], a value of 2.25 is appropriate for α and a value of 1 for β. Normalization operations have been performed in the drawing and calculations related to the assurance map windows. To reduce the effects of the side pixels of the window on the calculations and weighting in the following equation, a Hamming window is used, which is also shown in the figure below.

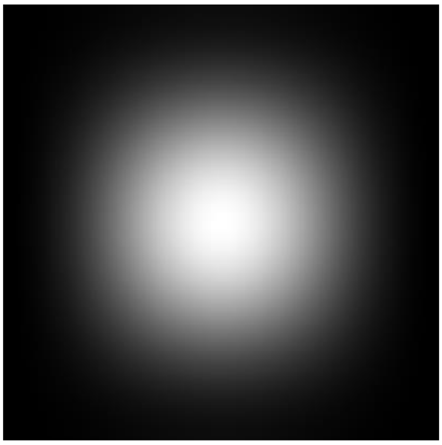


Figure 4: Hamming window used

Via the following formula, you can change the scale value of the Hamming window at any stage, but due to the lack of specific events in the videos under review, keeping this parameter constant for 5-10 frames seems reasonable. This parameter is updated according to the following formula. We set the value of λ to 0.25. Extremely increasing or decreasing this variable causes σ to be estimated using previous or current data.

$$\begin{cases} s'_t = \sqrt{\frac{c_t(x_t^*)}{c_{t-1}(x_{t-1}^*)}}, \\ \bar{s}_t = \frac{1}{n}\sum_{i=1}^{n} s'_{t-i}\,, \\ s_{t+1} = (1-\lambda)s_t + \lambda\bar{s}_t, \\ \sigma_{t+1} = s_t\sigma_t \end{cases} \qquad (7)$$

Now the previous probability for the texture content model is calculated regarding the following formulas.

$$P(c(z)|o) = I(z)\omega_\sigma(z - x^*) \qquad (8)$$

$$\omega_\sigma(z) = ae^{-\frac{|z|^2}{\sigma^2}} \qquad (9)$$

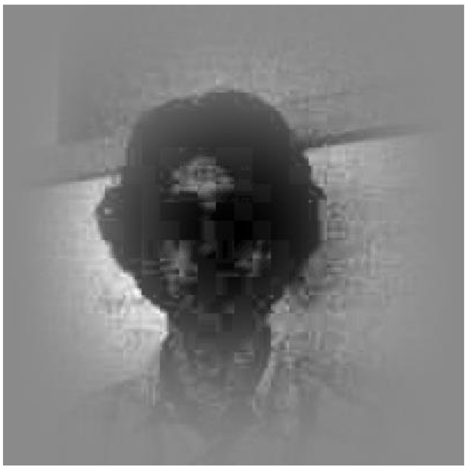


Figure 5: Texture content model

In this step, the assurance map is calculated according to the following formula.

$$c_{t+1}(x) = F^{-1}(F\big(\,H_{t+1}^{stc}(x)\big) \odot F\big(I_{t+1}(x)\omega_{\sigma_t}(x - x_t{}^*)\big)\big) \qquad (10)$$

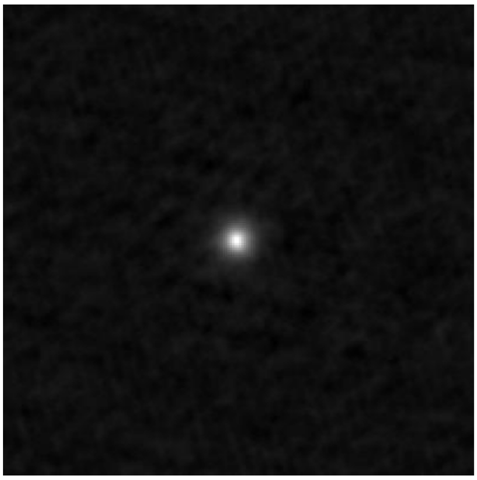


Figure 6: Assurance map

The h^sc value is then updated according to the following formula.

$$h^{sc}(x) = F^{-1}\left(\frac{F\left(be^{-\left|\frac{x-x^*}{\alpha}\right|^{\beta}}\right)}{F(I(x)\omega_\sigma(x-x^*)}\right) \qquad (11)$$

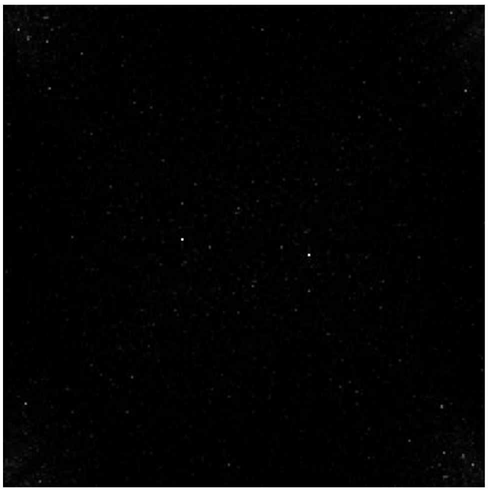


Figure 6: Time-space content in the frequency domain

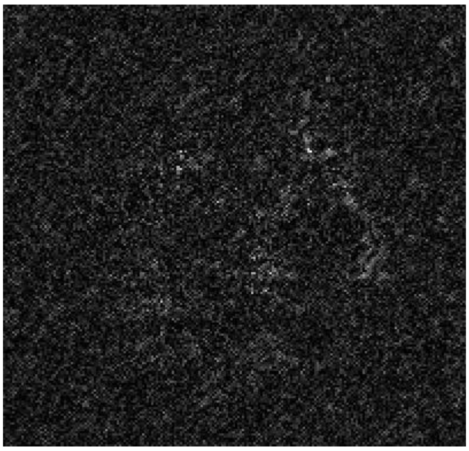


Figure 7: Content of time-space in the location domain

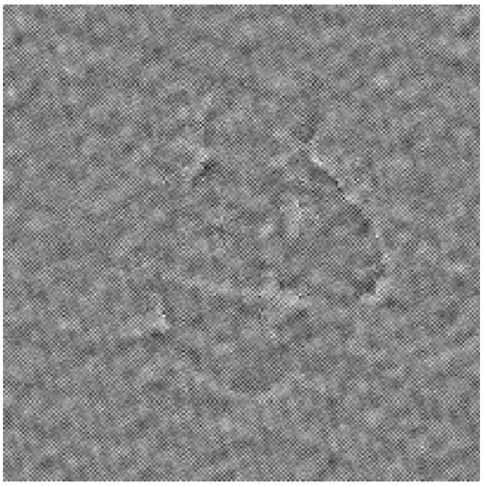


Figure 8: Updated location-time content in the location domain

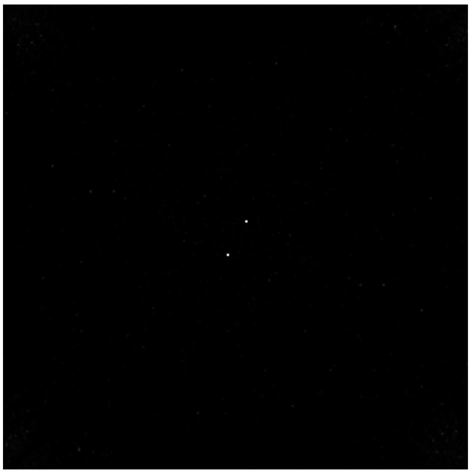
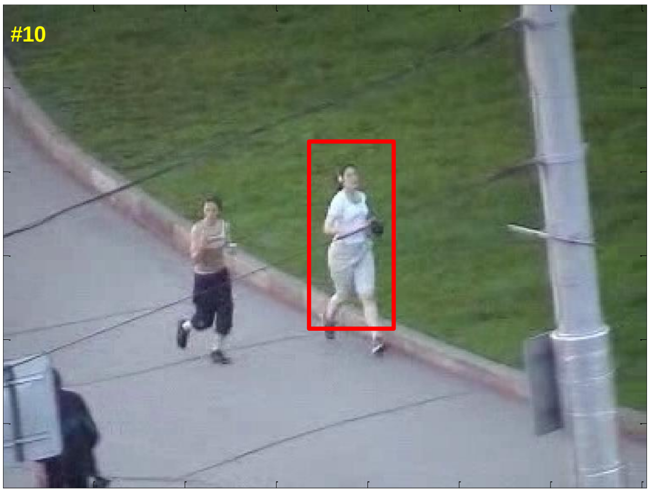


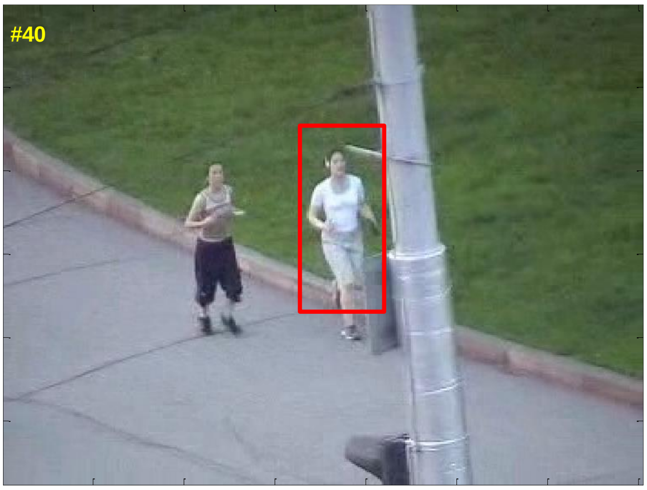
Figure 9: Updated temporal content in the frequency domain

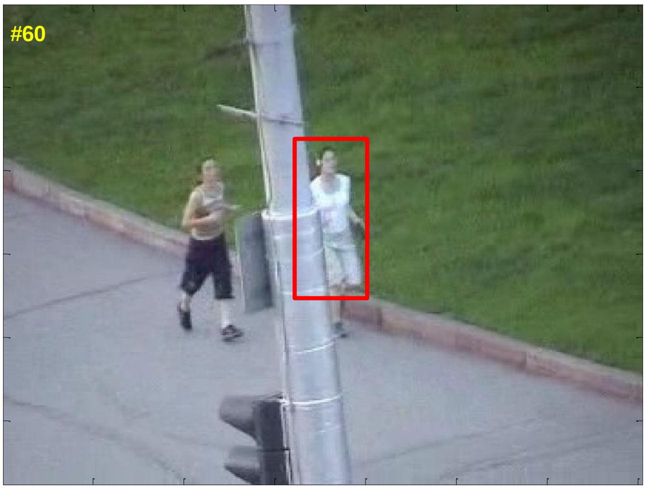
The content of space-time is updated via the following formula.

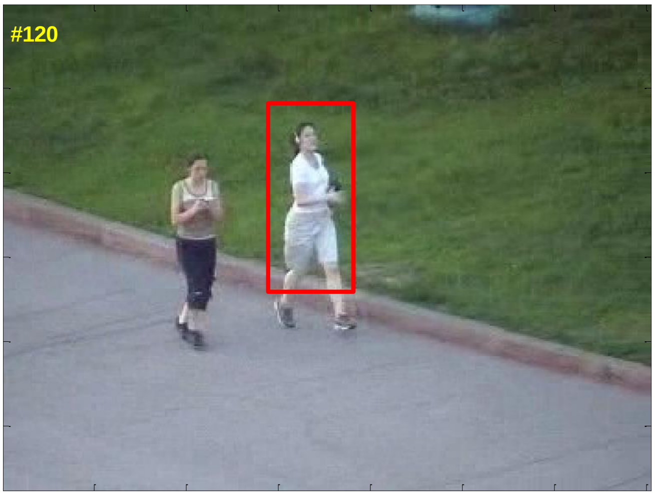
$$H_{t+1}^{stc} = (1 - \rho)H_t^{stc} + \rho h_t^{sc} \qquad (12)$$

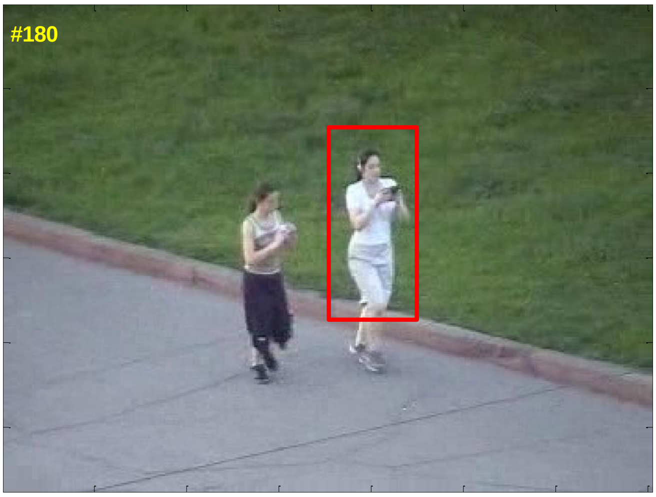
Then the area marked with a rectangle is drawn in the main image. The number of frames is above them. The person's movement can be noticed according to the background image, the brightness of the image, and the state of the person's face.

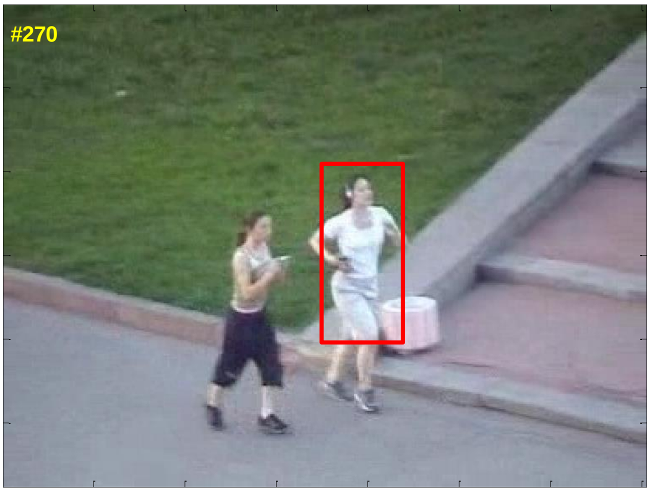





#60

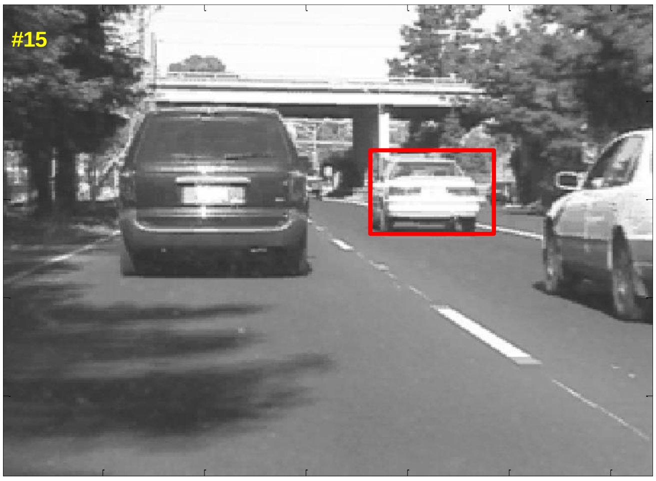

#120

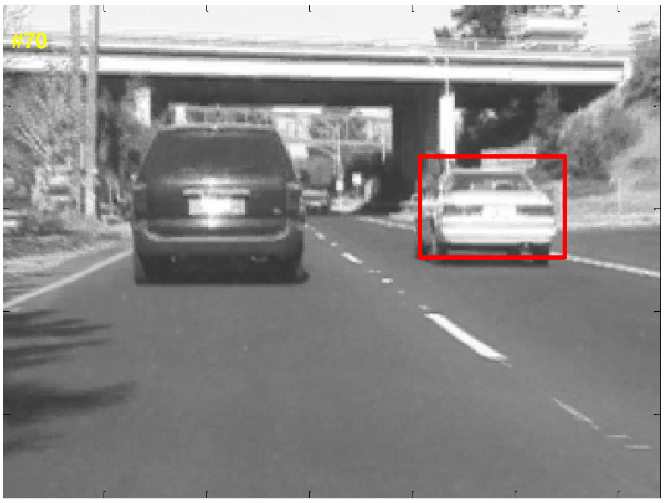

#180

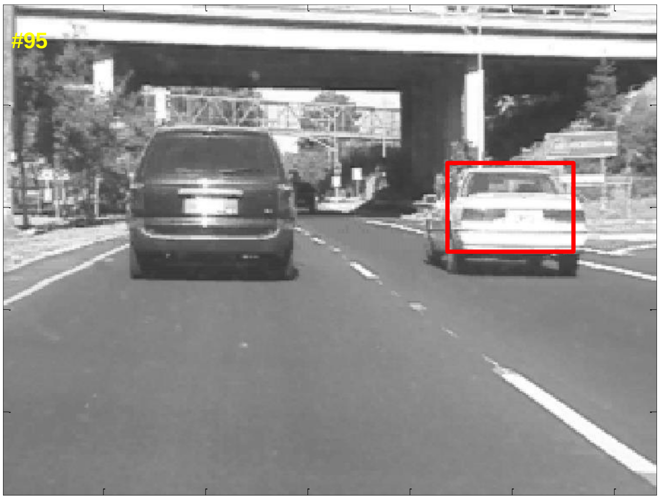


Test results of data on vehicle motion data 1:

To get better results from the algorithm in this video, the value of the variable α is set to 0.25. This makes the window dynamic. It can be seen that in paths where the distance from the camera to the target changes and the perspective phenomenon is affected, the value of this variable should be less than in cases where the distance from the camera to the target changes less.

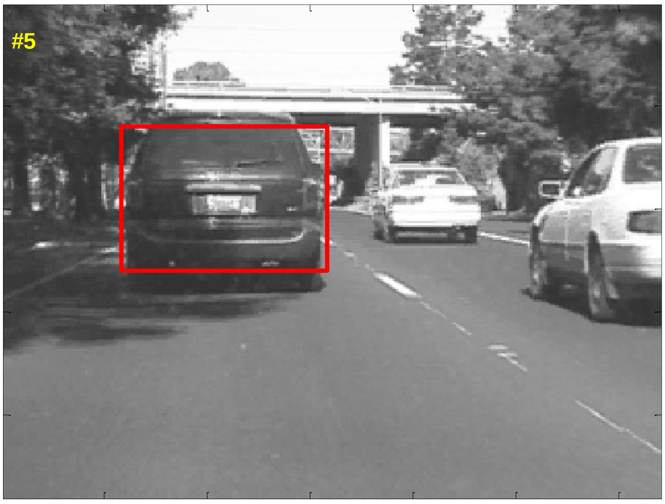


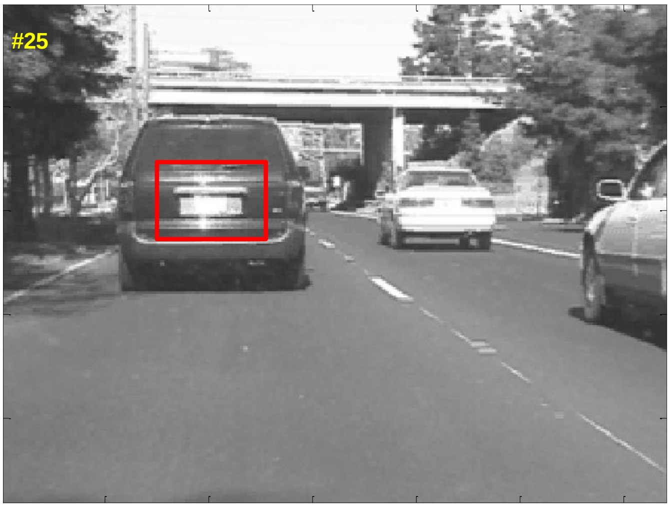


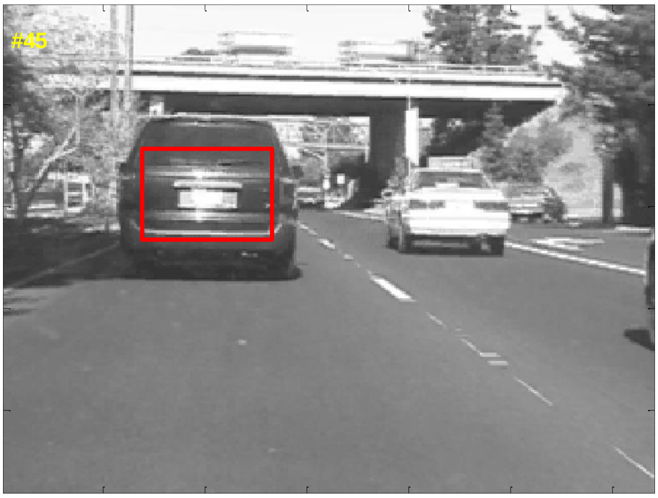


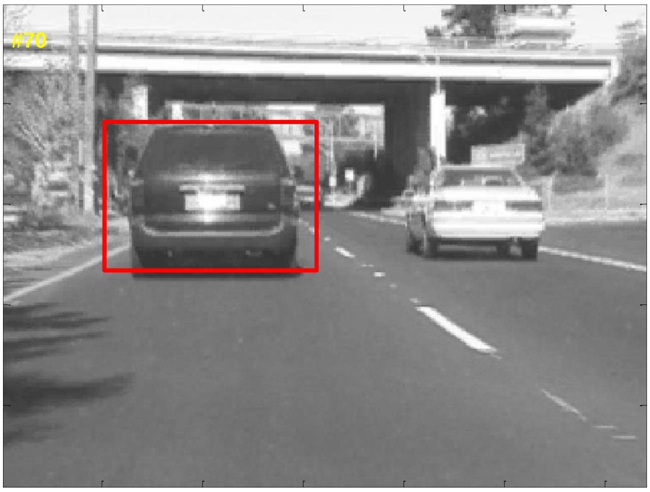
Data test results on vehicle motion database 2:

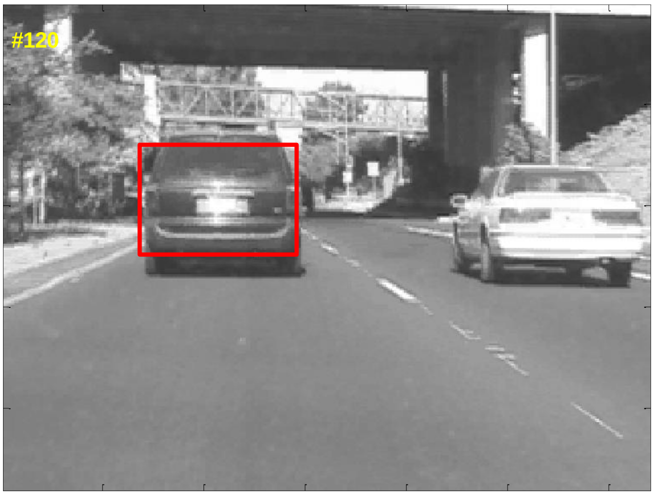
The value of α for this car is set to 0.5. Because in 300 frames and later, an obstruction occurs for this car and the image of another moving car comes next to it.

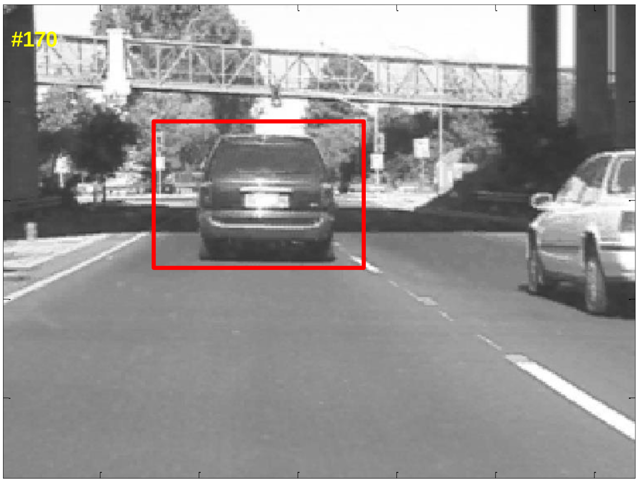

#5

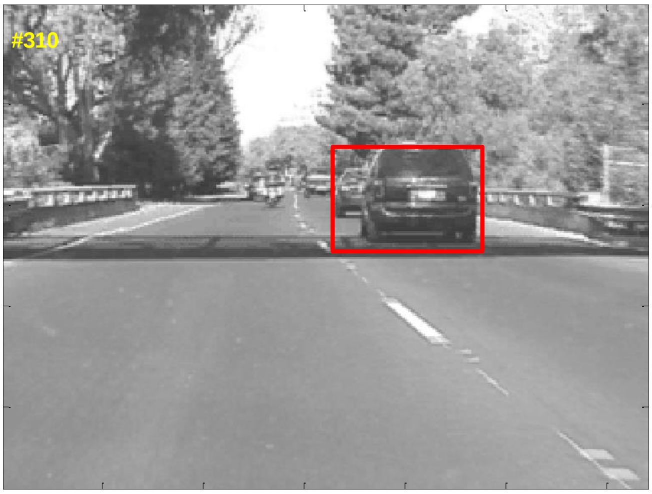

#25

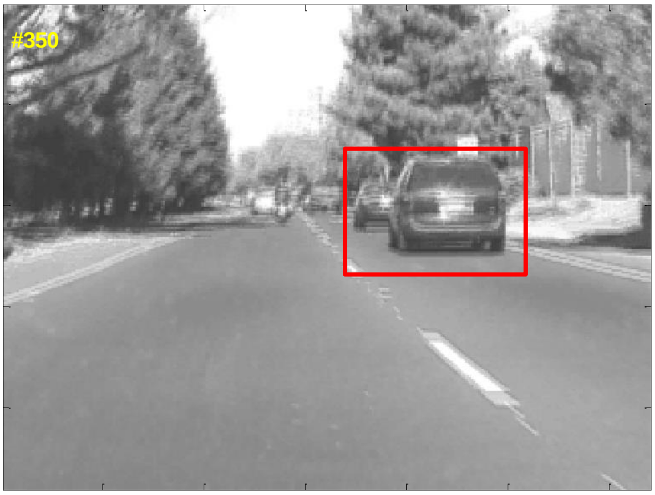

#45

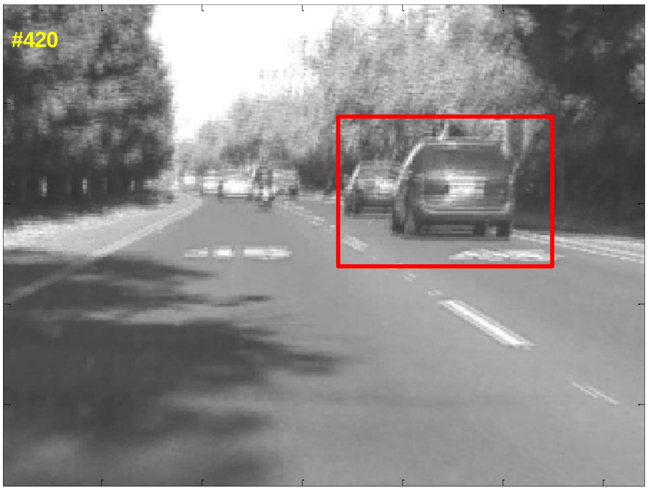

#70


#120


#170







Based on the simulation results, it is evident that objects can be tracked in different modes. Low computational complexity is one of the main features of the proposed algorithm in which FFT operations are performed to process a frame, including learning the spatial content model and calculating the assurance mapping. The computational complexity for calculating each FFT is only of the order of o (MN log (MN)) for local regional content with M × N pixels, thus leading to a fast method. Most importantly, the proposed algorithm achieves the strong results discussed.

Regarding the experiments done on different videos, motion scenarios in the video can be divided into two categories. Movable camera mode and still camera mode. When the camera is moving, such as the video of David in the room or the video of chasing two cars, the movement of the moving object in consecutive frames is relatively low, so there is no need for excessive window dynamics to track the target. It causes objects in the background to be accepted as moving objects, and in some cases may lead content to background data. Consequently, we set the learning coefficient λ between 0.03 and 0.06. Likewise, the variable α can be changed between 0.2 and 1. In cases where the camera is still, the movement of the moving object in consecutive frames is relatively greater, and therefore there is no need for excessive window dynamics or readiness to jump to the target location. Similarly, objects and backgrounds do not interfere continuously, and changing content in the margin points is not as annoying as in the case of a moving camera. Background disturbance occurs when a moving object passes behind or inside the background object. Consequently, we set the learning coefficient λ between 0.05 and 0.07. The α variable can also be changed between 0.5 and 1.5.

To quantitatively compare the proposed algorithm with other algorithms, it is essential to do tracking operations for all different videos with all algorithms and then calculate the tracking error rate. The tracking error is equivalent to the difference between the center of the tracking window and the actual location stated for the target position. Compared methods include mean shift, tracking using L1 software, least-squares total algorithm, intensive tracking, and local-global tracker.

Table 1: Quantitative comparison of the proposed algorithm with other algorithms

Videos reviewed	Mean Shift Algorithm	Tracking algorithm using L1 norm	Total least squares algorithm	Compact tracking algorithm	local-global tracker	Proposed algorithm
Car4	144	16	117	63	47	11
Car 11	76	8	8	9	16	7
davidindoor	176	86	78	28	12	8

It can be seen that the proposed algorithm has a much better and more acceptable performance than other algorithms.

4. Conclusion

In the current study, a fast yet simple algorithm of Spatio-temporal content information was offered for visual tracking. Two spatial content models (e.g., spatial content and spatial-temporal content models) were presented that are stout against the apparent changes introduced by an obstruction, light changes, and positional changes. The fast Fourier transform algorithm was used in online learning and detection. The result was an efficient tracking method implemented with MATLAB. Several experiments on the algorithms of preceding studies reveal that the proposed algorithm leads to more favorable results in terms of accuracy, power, and speed. Lastly, the range of changes of the parameters was stated that should be set.

The Spatio-temporal content of the proposed tracking algorithm is considerably different from the content-based method, which uses FFT for efficient computing. All of the content-based methods mentioned use strategies to find regions that are consistent with the target motion correlation. To find consistent regions, key points around the target are first extracted to help determine the position of the main target. Next, the SIRF or SURF descriptor is used to indicate these consistent regions. Consequently, heavy computational operations are required to display and find consistent regions. Furthermore, due to the sparse nature of the key points, some consistent regions that are useful for locating the target may be discarded. In contrast, the proposed algorithm does not have these problems because all local regions around the target are considered as potentially consistent regions and the correlation of motion between objects and their local contents is learned by the Spatio-temporal content model in successive frames that have been effectively calculated by FFT.

The proposed algorithm differs meaningfully from other algorithms in various respects. First, our algorithm models the Spatio-temporal relationship between the target and its local content, which is stimulated by the human visual system and uses the content to help resolve ambiguities in complex scenes efficiently and effectively. Second, our algorithm focuses on regions that require careful analysis, thus effectively reducing the side effects of background clutter and leading to stronger results. Third, our algorithm controls the problem of ambiguity of the target location using an assurance mapping with the previous proper distribution, thus achieving a more stable and accurate performance for more visual tracking. Lastly, our algorithm solves the scale adaptation problem, but other FFT-based tracking methods only track fixed-scale objects and have less accurate results than our method.

Regarding the results of practical implementation of the designed algorithms and comparison of the results with the simulated values, analysis of the data processing mechanism designed in specific conditions including incorrect camera performance and data processing design for a network of 2D and 3D surveillance sensors is proposed. Becomes. Likewise, the conversion of simulations to comprehensive software of data processing machine with the ability to change the effective parameters in each sub-block through a suitable user interface and design and simulation of fast-tracking algorithms with high computational load should be considered.

References

[1] Benezeth, Y., Jodoin, P.M., Emile, B., Laurent, H., and Rosenberger, C., "Review and evaluation of commonly-implemented background subtraction algorithms", International Conference on Pattern Recognition (ICPR 2008), 19th Publication Date: 8-11, pp. 1-4(2014).

[2] Comaniciu, D., Meer, P., "Mean shift: A robust approach toward feature space pp. analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, 619(2014).

[3] Guang Zheng. Moving Object Tracking and 3D Measurement Using Two PTZ Cameras, 5th International Conference on Intelligent Networking and Collaborative Systems, (2013).

[4] H. Grabner, J. Matas, L. Van Gool, and P. Cattin, "Tracking the invisible: Learning Where the object might be," in CVPR, pp. 1285–1292, (2010).

[5] H.Oike, H, Wu, C, Hua and T, Wada, High-Speed Binocular Active Camera System for Capturing Good Image of a Moving Object, IEICE, D VOL.J91-D No.5, pp.1393-1405, (2008).

[6] K.Deguchi, S.Kagami, S.Saga and H.Hontani, Real-Time Object Tracking and Reconstruction by Active Camera, Computer and Image media, P111-8, (2007).

[7] K. Muhlmann, D. Maier, J. Hesser, and R. Manner: "Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation", Proc. IEEE Workshop Stereo and Multi-Baseline Vision, pp. 30-36, (2010).

[8] M. Juengel, H. Mellmann, and M. Spranger, "Improving vision-based distance measurements using reference objects," RoboCup, Lecture Notes in Artificial Intelligence (2007).

[9] M.Z. Brown, D. Burschka, G.D. Hager: "Advances in computational stereo", IEEE Transactions on Pattern Analysis and Machine Intelligence, 25 (8) 993, 1008 (2012).