Literature Review

Topic : Background Subtraction with a Freely Moving Camera

Air Force Island

Guanfang Dong, Yingnan Ma

Introduction

In the real world, background subtraction plays an important role in many areas. For a freely moving camera, it is challenging to implement background subtraction. Due to the movement of the camera, the whole background is changed. In order to avoid recognizing the moving background as foreground, we need a more accurate algorithm, which can recognize the moving objects from the “moving” background.

In recent years, different methods and algorithms are designed to solve the problem of background subtraction for a freely moving camera. Employing particle trajectory is a widely used method for detecting moving objects. Particle trajectory “is the moving path of a particle across a video clip” [1]. By computing the optical flow, [6] we can capture a particle trajectory that involves finding dense trajectories. After we get the particle trajectory, we can detect the moving object from the background. In addition to the particle trajectory, the Bayesian framework [12] can also be used for detecting moving objects by estimating dense optical flow over time. Besides the above algorithms, the analysis of optical flow can also be used in other methods. For example, [2] optical flow methods can be combined with Gaussian Mixture Model (GMM). With the support of temporal differencing, the moving objects can be extracted successfully from the video.

Another popular method used to detect moving objects is analyzing texture of the video. Texture based methods use discriminative texture features to capture background statistics. By calculating the histograms over a circular region of each pixel, [6] we can model pixels as a group of adaptive local binary pattern (LBP) texture operator. In addition, [9] texture can also be used to solve the background subtraction difficulties in highly dynamic scenes.

Other methods can also solve the background subtraction for a freely moving camera. Instead of formulating the representation of the background, we can achieve the goal by implementing deep learning. With the use of deep pixel distribution learning (DPDL) model, [13] the distribution of observations can be obtained. With the analysis of the distribution by using convolutional neural network (CNN), we can recognize the foreground from the background.

In real life, the videos recorded by a freely moving camera may involve some unavoidable accidents, such as camera jitter, which can lead to bad performance of background subtraction results. In order to design a single system that can achieve the background subtraction in different environments, [7] the Multiple Background Model is introduced. Multiple Background Model can analyze the changes during the process and choose optimal color space dynamically, so that the bad performance caused by these accidents can be fixed.

For the above methods and algorithms, some are similar to what we will use in the project. In our project, we will start at the point of particle trajectory by computing optical flow. Based on the particle trajectory, we will use probabilistic graphical model to receive pixel wise foreground and background labeling so that we can obtain a coarse foreground. By editing this coarse foreground in detail, we can get a more accurate foreground.

However, we will implement some creative ideas to utilize and improve above methods. Previous methods involved Otsu's thresholding, [1] which could lead to bad performance for changed-velocity moving objects. To improve performance, we will remove the optical flow after we obtain the coarse foreground. By analyzing the texture of the video, we can avoid the negative effect caused by optical flow when recognizing foreground from background. In addition to the method of removing optical flow, we will also implement discrete cosine transform (DCT) in our project so that we can cancel the noise on each frame of the video. By combining the above methods, the accuracy of background subtraction result will be improved dramatically.

Review of particle trajectory methods

Capturing particle trajectory involves finding [6] dense trajectories based on advecting optical flow. However, the analysis of optical flow may be influenced by the quality of video. For example, too much light or overexposed videos may cause offset for optical flow. Thus, we cannot assert that the particle trajectory is captured perfectly. Based on current drawbacks, we will propose a new method to avoid aforementioned difficulties. We were inspired by [14] JPEG picture compression. The compression uses [11] discrete cosine transformation (DCT) to cancel the noisy spots in pictures, which makes pictures look smoother. DCT can be computed by fast Fourier transform, which will not lead to too much computational waste during the image processing. This method can dramatically improve the accuracy of the particle trajectory captured during the process.

Apart from the method we just mentioned, Wu, Oreifej and Shah proposed a method for unaligned videos. They [6] "decompose the trajectories into camera-induced and object-induced components" based on low rank optimization. After having those relevant object motion trajectories, they computed a "compact set of chaotic invariant features," which captured the characteristics of the trajectories. Finally, they learned and recognized the human actions using the computed motion features. However, this approach had some significant drawbacks. Firstly, this method could only precisely detect the human movements such as boxing and waving. It is assumed that the human was not disabled because this method would take more care of rigid body movements and articulated motions. Secondly, this method is hard to use for a freely moving camera or even for a dynamic scene. For example, it is hard to apply this method on a snowy day because snowflakes would cause a lot of noisy spots for capturing and distinguishing the camera-included or object-included components. Therefore, due to these drawbacks, we would not use their methods in our project.

Different from previous particle trajectory capturing methods, Sheikh, Javed and Kanade proposed a method that could avoid the above drawbacks. Due to the fact that [3] "image motion is induced by a confluence of camera motion, independent object motion, and the 3D structure of the scene", the motion of objects can be recognized by geometric constrains. After that, these sparse locations [3] "can be used to build foreground and background appearance models." With the use of above models, moving objects can be segmented.

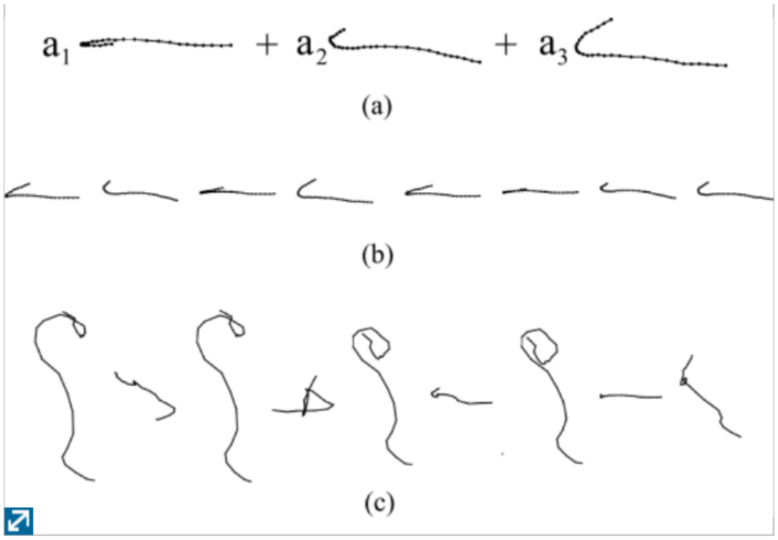

Figure 1. Difference between background and foreground particle trajectory

From our perspective, this method is much more reliable than the previous one. The most important reason is that the background particle trajectory as shown in Figure 1, which is caused by camera motion is totally different from the object motion. Therefore, applying a reasonable constraint could easily distinguish a coarse foreground and background. In fact, humans use a very similar method to detect moving objects from what they see. Humans observe the differences of moving patterns between objects and background, according to the heuristic knowledge and 3d structure. Therefore, this method is a very good approach to find a coarse foreground area.

Review of thresholding methods

After applying particle trajectories and using a reasonable constraint to find the coarse foreground, we could find a suitable thresholding value to obtain a fine foreground. For previous methods, Otsu's thresholding selection method was a good approach to solve this problem. Specifically, we could find a histogram that [5] "has a deep and sharp valley between two peaks." This histogram represents moving objects and background respectively. Therefore, the thresholding could be chosen at the bottom of the valley. However, Wu, He and Nguyen proved that Otsu's threshold method only worked well in [1] "videos with a single moving object or multiple moving objects in consistent motion." For the velocity-changed objects, Otsu's thresholding would misclassify those objects as background.

To deal with this problem, Wu, He and Nguyen [1] suggested normalizing the histogram first. They applied Min-Max, Arctan and Log normalizations to find the features of the origin data values, scatter the small values and cluster the large values respectively. Then, they combined them and chose the highest value of "clustered-ness" as the optimal thresholding value. Leeham, Yan, Takru, Tan and Mian [4] also concluded other popular thresholding methods such as Niblack's Method, Mean-Gradient Technique, Quadratic Integral Ratio and Yanowitz and Bruckstein's Method. Basically, some methods might have very good performance, but the computational cost was expensive. For example, the computational cost of Yanowitz and Bruckstein's method was O(N^3), due to the fact that it applied interpolation for threshold surface to smooth image and a post-processing method to validate the segmented image. Therefore, the above methods were not suitable for computing adaptive thresholding for a whole video in our project.

Review of optical flow methods

Due to the fact that thresholding methods are not suitable or not creative to find a fine foreground, we will propose a new method called background optical flow cancellation. After background optical flow is successfully canceled in each frame, we can reliably decrease the bad influence that caused by background optical flow. Therefore, a fine foreground can be easily captured.

In order to compute optical flow, Zhou and Zhang [2] suggested that we could applying Lucas-Kanada's gradient-based method because this method had been proven as the most accurate and computationally efficient method. Specifically, [15] they first computed velocity from spatiotemporal derivatives of image intensity or filtered versions of the image. Then, they constrained the 2-D velocity by a second-order differential method. Finally, they combined local estimates of component velocity and 2-D velocity through space and time, thereby producing a robust estimate of optical flow. After the optical flow was estimated, Zhou and Zhang suggested [2] building a Gaussian pyramid for "both source (at time t-1) and the target images (at time t)". Zhou and Zhang used motion estimate and an optical flow constraint equation to find a coarse area that contains foreground, which is relatively similar to our first approach. But, our method would compute the optical flow value based on the coarse foreground that we had discovered. Therefore, the union of the two methods could generate an accurate fine foreground.

Apart from the aforementioned methods, Yalcin, Collins and Black [12] also mentioned another reasonable method to find layered models of optical flow by using the Bayesian framework. In order to achieve this, they only applied the optical flow algorithm on stabilized frames because the "stabilization of the frames compensates for gross affine background motion prior to running robust optical flow to compute dense residual flow." As a result, further details of the approach and videos could be found. Considering the differences in video qualities, the improvement of this method would be different. We would check the accuracy of the method that used optical flow algorithm only. If the results are not as expected, we would implement further optimization.

Review of texture-based methods

Many state-of-the-art methods could be implemented to solve for background subtraction with a freely moving camera. In addition to the above methods, another popular method that can be used to detect moving object is analyzing texture features of the video.

The idea of the texture-based method is "modeling the background and detecting moving objects from a video sequence" [6] by analyzing the texture features of the video. M. Heikkila and M. Pietikainen [6] mentioned an approach that "uses discriminative texture features to capture background statistics." They used a modified local binary pattern as the texture operator because of “its tolerance against illumination changes and its computational simplicity” [6]. By calculating the histograms over a circular region of each pixel, they modeled pixels as a group of adaptive local binary pattern (LBP) histograms.

Besides the above texture-based method, texture could also be used to solve the background subtraction difficulties in highly dynamic scenes. Vijay Mahadevan and Nuno Vasconcelos [9] used center-surround computations to define saliency locally. They proposed a discriminant formulation and [9] “the saliency of a location is the discriminant power of a set of features with respect to the binary classification problem which opposes center to surround.” They provided an unsupervised algorithm to implement background subtraction. Without learning the parameters of the background, [9] they only focused on the differences of motion between the center and surround, which makes the algorithm more effective.

In our project, we also involved the texture of the video. After we obtain a coarse foreground, we will remove the optical flow by analyzing the texture of the video. By analyzing the texture in each frame, the background optical flow can be successfully removed, which would avoid poor performance resulting from optical flow.

Review of using Gaussian Mixture Model

As mentioned before, the Gaussian Mixture Model was widely used in solving background subtraction problem. According to Ren, Chua and Ho [10], the reason is that spatial distribution of Gaussians allows each background pixel modeled temporally and spatially. Therefore, it is easy to classify the pixel ass the foreground or background. However, we still need to update the parameters of the Gaussian Mixture Model in each iteration for better performance. To deal with this problem, a narrow Gaussian for the background and a flat Gaussian for the moving objects should be obtained at the same time. With the help of those two models, “the background Gaussian distribution of every pixel compose a background map, which is adapted frame by frame to each new incoming frame.” Therefore, better results can be obtained.

Although this method is very efficient and accurate, we decided to use our creative idea to generate a fine foreground by removing and canceling undesired background optical flow.

Review of deep learning methods

Many methods solve the background subtraction for a freely moving camera by comparing backgrounds in different frames. However, instead of formulating the representation of the background, we can also achieve the goal by implementing deep learning.

Chenqiu Zhao and his teammates solved the background subtraction by analyzing [13] “the classification of a pixel's current observation in comparison to historical observations, and propose a Deep Pixel Distribution Learning (DPDL) model for background subtraction.” With the use of Deep Pixel Distribution Learning model, the distribution of observations could be obtained. They represented the distribution of past observations by Random Permutation of Temporal Pixels

(RPoTP) [13]. With the use of pixel-wise representation, enough Random Permutation of Temporal Pixels features could be promised, which lays a solid foundation for the analyzing steps. After obtaining the distribution, they learned the distribution using a neural net. They implemented a convolutional neural network to recognize the foreground and the background. With the use of random permutation, the framework could be designed to focus on distribution of observations [13].

The DPDL model is an effective approach to solve the background subtraction problem. However, our project will focus on utilizing state-of-the-art methods and making improvements by implementing creative ideas, which will also enhance the performance significantly.

Review of other methods

In real life, the videos recorded by freely moving camera may involve some unavoidable accidents. In order to design a single system that can achieve the background subtraction in different environment, Hasan Sajid and Sen-Ching Samson Cheung [7] introduced the Multiple Background Model. The Multiple Background Model analyzed the changes during the video process and chose the optimal color space dynamically, so that bad performance caused by these accidents could be fixed. This method "allowed the algorithm's applicability to a wide variety of challenges associated with change detection including camera jitter, dynamic background, Intermittent Object Motion, shadows, bad weather, thermal, night videos etc" [7].

References

[1]Wu, Y., He, X., & Nguyen, T. Q. (2017). Moving object detection with a freely moving camera via background motion subtraction. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(2), 236-248.

[2]Zhou, D., & Zhang, H. (2005, October). Modified GMM background modeling and optical flow for detection of moving objects. In *2005 IEEE international conference on systems, man and cybernetics* (Vol. 3, pp. 2224-2229). IEEE.

[3]Sheikh, Y., Javed, O., & Kanade, T. (2009, September). Background subtraction for freely moving cameras. *In Computer Vision, 2009 IEEE 12th International Conference on* (pp. 1219-1225). IEEE.

[4]Leedham, G., Yan, C., Takru, K., Tan, J. H. N., & Mian, L. (2003, August). Comparison of some thresholding algorithms for text/background segmentation in difficult document images. In *null* (p. 859). IEEE.

[5]Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, *9*(1), 62-66.

[6]Wu, S., Oreifej, O., & Shah, M. (2011, November). Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *Computer Vision (ICCV), 2011 IEEE International Conference* on (pp. 1419-1426). IEEE.

[7]Sajid, H., & Cheung, S. C. S. (2015, September). Background subtraction for static & moving camera. In *Image Processing (ICIP), 2015 IEEE International Conference* on (pp. 4530-4534). IEEE.

[8]Heikkila, M., & Pietikainen, M. (2006). A texture-based method for modeling the background and detecting moving objects. *IEEE transactions on pattern analysis and machine intelligence*, *28*(4), 657-662.

[9]Mahadevan, V., & Vasconcelos, N. (2008, June). Background subtraction in highly dynamic scenes. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*(pp. 1-6). IEEE.

[10]Ren, Y., Chua, C. S., & Ho, Y. K. (2003). Motion detection with nonstationary background. *Machine Vision and Applications*, *13*(5-6), 332-343.

[11]Ahmed, N., Natarajan, T., & Rao, K. R. (1974). Discrete cosine transform. *IEEE transactions on Computers*, *100*(1), 90-93.

[12]Yalcin, H., Hebert, M., Collins, R., & Black, M. J. (2005, June). A flow-based approach to vehicle detection and background mosaicking in airborne video. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 2, pp. 1202-vol). IEEE.

[13]Zhao, C., Cham, T. L., Ren, X., Cai, J., & Zhu, H. (2018, July). Background subtraction based on deep pixel distribution learning. In *2018 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1-6). IEEE.

[14]Wallace, G. K. (1992). The JPEG still picture compression standard. *IEEE transactions on consumer electronics*, 38(1), xviii-xxxiv.

[15]Barron, J. L., Fleet, D. J., & Beauchemin, S. S. (1994). Performance of optical flow techniques. *International journal of computer vision*, *12*(1), 43-77