Video Analysis By Object Detection And Tracking Methods

Raghavendra T.K [1], Vinutha M [2]
P.G Student, Department of Computer Science & Engineering,
AIT Chickmagalur, VTU Belgaum.
[1]raghu.kit.tip@gmail.com, [2]vinuthaaam@gmail.com

Abstract – Now a day object tracking and object detection is a major challenging problem. Tracking objects can arise due to abrupt object motion, changing appearance patterns of object and the scene, non-rigid object structures, object-to-object and object-to-scene occlusions, and camera motion. Tracking is usually performed in the higher-level applications that require the location and shape of the object in all frames. We have presented the important issues related to tracking and detection including the use of appropriate image features and detection of objects.

Keywords – **Object tracking, Point tracking, Kernel tracking, Silhouette tracking, Object detection etc.**

I. INTRODUCTION

Object tracking is a technique or method used to track the number and direction of objects traversing a certain passage or entrance per unit time. The resolution of the measurement is entirely dependent on the sophistication of the technology employed. The device is often used at public places such as railway stations, shopping malls; air ports etc so that the movement of each individual object can be analyzed. Many different technologies are used in tracking devices, such as infrared beams, computer vision and thermal imaging.

There are various reasons for object tracking. One such usage is people counting. In retail stores, counting is done as a form of intelligence-gathering. The use of people counting systems in the retail environment is necessary to calculate the Conversion Rate, i.e. the percentage of a store's visitors that makes purchases. This is the key performance indicator of a store's performance and is far superior to traditional methods, which only take into account sales data. The most important use of this technique is to track a particular object of our interest and to maintain a record of status of that object which can be analyzed for further information like for example in case of suspicious left language detection in public places like railway stations etc. and in video surveillance systems to keep track of the movements of suspicious activities.

Effective detection and tracking require accurate measurements of object position and motion, even when the sensor itself is moving.

1. OBJECT DETECTION

Every tracking method requires an object detection mechanism either in every frame or when the object first appears in the video. A common approach for object detection is to use information in a single frame. However, some object detection methods make use of the temporal information computed from a sequence of frames to reduce the number of false detections. This temporal information is usually in the form of frame differencing, which highlights changing regions in consecutive frames. Given the object regions in the image, it is then the tracker's tasks to perform object correspondence from one frame to the next to generate the tracks. Although the object detection itself requires a survey of its own, here we outline the popular methods in the context of object tracking for the sake of completeness.

2.1 Point Detectors

Point detectors are used to find interest points in images which have an expressive texture in their respective localities. Interest points have been long used in the context of motion, stereo, and tracking problems. A desirable quality of an interest point is its invariance to changes in illumination and camera viewpoint.

2.2 Background Subtraction

Object detection can be achieved by building a representation of the scene called the background model and then finding deviations from the model for each incoming frame. Any significant change in an image region from the background model signifies a moving object. The pixels constituting the regions undergoing change are marked for further processing. Usually, a connected component algorithm is applied to obtain connected regions corresponding to the objects. This process is referred to as the *background subtraction.*

Background subtraction became popular following the work of Wren et al. [1997]. In order to learn gradual changes in time, Wrenetal. Propose modeling the color of each pixel, $I(x, y)$, of a stationary background with a single 3D (Y, U, and V color space) Gaussian, $I(x, y) \sim N(\mu(x, y), \_(x, y))$. The model parameters, the mean $\mu(x, y)$ and the covariance $\_(x, y)$, are learned from the color observations in several consecutive frames. Once the background model is derived, for every pixel (x, y) in the input frame, the likelihood of its color coming from $N(\mu(x, y), \_(x, y))$ is computed, and the pixels that deviate from the background model are labeled as the foreground pixels. A substantial improvement in background

modeling is achieved by using multimodal statistical models to describe per-pixel background color. In this method, a pixel in the current frame is checked against the background model by comparing it with every Gaussian in the model until a matching Gaussian is found. If a match is found, the mean and variance of the matched Gaussian is updated; otherwise a new Gaussian with the mean equal to the current pixel color and some initial variance is introduced into the mixture. Each pixel is classified based on whether the matched distribution represents the background process. Moving regions, which are detected using this approach with the background models also.

Another approach is to incorporate region-based (spatial) scene information instead of only using color-based information. Lgammal & Davis [2000] use nonparametric kernel density estimation to model the per-pixel background. During the subtraction process, the current pixel is matched not only to the corresponding pixel in the background model, but also to the nearby pixel locations. Thus, this method can handle camera jitter or small movements in the background. Li and Leung [2002] fuse the texture and color features to perform background subtraction over blocks of 5 × 5 pixels. Since texture does not vary greatly with illumination changes, the method is less sensitive to illumination.

An alternate approach for background subtraction is to represent the intensity variations of a pixel in an image sequence as discrete states corresponding to the events in the environment. For instance, for tracking cars on a highway, image pixels can be in the background state, the foreground (car) state, or the shadow state. Rittscheretal. [2000] use Hidden Markov Models (HMM) to classify small blocks of an image as belonging to one of these three states. In the context of detecting light on and off events in a room, Stenger et al. [2001] use HMMs for the background subtraction. The advantage of using HMMs is that certain events, which are hard to model correctly using unsupervised background modeling approaches.

Instead of modeling the variation of individual pixels, Oliver et al. [2000] propose a holistic approach using the eigenspace decomposition. For k input frames, I_i: $i = 1 \cdot \cdot \cdot k$, of size $n \times m$, a background matrix $\mathbf{B}$ of size $k \times l$ is formed by cascading m rows in each frame one after the other, where $l = (n \times m)$, and eigenvalue decomposition is applied to the covariance of $\mathbf{B}$, $\mathbf{C} = \mathbf{B}_T\mathbf{B}$. The background is then represented by the most descriptive η eigenvectors, $\mathbf{u}_i$, where $i < \eta < k$, that encompass all possible illuminations in the field of view (FOV). Thus, this approach is less sensitive to illumination. The foreground objects are detected by projecting the current image to the eigenspace and finding the difference between the reconstructed and actual images.

One limitation of the aforementioned approaches is that they require a static background. This limitation is addressed by Monnet et al. [2003], and Zhong and Sclar off [2003]. Both of these methods are able to deal with time-varying background (e.g., the waves on the water, moving clouds, and escalators). These methods model the image regions as autoregressive moving average (ARMA) processes which provide a way to learn
and predict the motion patterns in a scene. An ARMA process is a time series model that is made up of sums of autoregressive and moving-average components, where an autoregressive process can be described as a weighted sum of its previous values and a white noise error. The most important limitation of background subtraction is the requirement of stationary cameras.

2.3 Segmentation

The aim of image segmentation algorithms is to partition the image into perceptually similar regions. The recent segmentation techniques are relevant to object tracking.

2.3.1 Mean-Shift Clustering: the mean-shift approach to find clusters in the joint spatial+color space, [l , u, v, x, y], where [l , u, v] represents the color and [x, y] represents the spatial location. Mean-shift clustering is scalable to various other applications such as edge detection and tracking. Mean-shift based segmentation requires fine tuning of various parameters to obtain better segmentation, for instance, selection of the color and spatial kernel bandwidths, and the threshold for the minimum size of the region considerably effect the resulting segmentation.

2.3.2. Image Segmentation Using Graph-Cuts: Image segmentation can also be formulated as a graph partitioning problem, where the vertices (pixels), $\mathbf{V} = \{u, v, \ldots\}$, of a graph (image), $\mathbf{G}$, are partitioned into N disjoint sub graphs (regions), Ai , $\_N\ i = 1\ Ai = \mathbf{V}$,$Ai \cap Aj = \emptyset$, $i \_= j$, by pruning the weighted edges of the graph. The total weight of the pruned edges between two sub graphs is called a *cut.* The weight is typically computed by color, brightness, or texture similarity between the nodes. Wu and Leahy [1993] use the minimum cut criterion, where the goal is to find the partitions that minimize a cut. In their approach, the weights are defined based on the color similarity. One limitation of minimum cut is its bias toward over segmenting the image. This effect is due to the increase in cost of a cut with the number of edges going across the two partitioned segments.

Shi and Malik [2000] propose the *normalized cut* to overcome the over segmentation problem. In their approach, the cut not only depends on the sum of edge weights in the cut, but also on the ratio of the total connection weights of nodes in each partition to all nodes of the graph. For image-based segmentation, the weights between the nodes are defined by the product of the color similarity and the spatial proximity. Once the weights between each pair of nodes are computed, a weight matrix $\mathbf{W}$ and a diagonal matrix $\mathbf{D}$, where $\mathbf{D}i,i = \_Nj = 1\mathbf{W}i, j$,

are constructed. The segmentation is performed first by computing the eigenvectors and the eigenvalues of the generalized eigensystem $(\mathbf{D}-\mathbf{W})\mathbf{y} = \lambda\mathbf{D}\mathbf{y}$, then the second-smallest eigenvector is used to divide the image into two segments. For each new segment, this process is recursively performed until a threshold is reached.

In normalized cuts-based segmentation, the solution to the generalized eigensystem for large images can be expensive in terms of processing and memory requirements. However, this method requires fewer manually selected parameters, compared to mean shift segmentation. Normalized cuts have also been used in the context of tracking object contours [Xu and Ahuja 2002].

2.3.3 Active Contours*:*

In an active contour framework, object segmentation is Achieved by evolving a closed contour to the object's boundary, such that the contour tightly encloses the object region. Evolution of the contour is governed by an energy functional which defines the fitness of the contour to the hypothesized object region.

Energy functional for contour evolution has the following common form:

$$E(\_) = \_ \, 10 \, E_{int}(\mathbf{v}) + E_{im}(\mathbf{v}) + E_{ext}(\mathbf{v})ds \qquad (1)$$

, where *s* is the arc-length of the contour _, *Eint* includes regularization constraints, *Eim* includes appearance-based energy, and *Eext* specifies additional constraints. *Eint* usually includes a curvature term, first-order ($\nabla\mathbf{v}$) or second-order ($\nabla 2\mathbf{v}$) continuity terms to find the shortest contour. Image-based energy, *Eim*, can be computed locally or globally. Local information is usually in the form of an image gradient and is evaluated around the contour [Kass et al. 1988; Caselles et al. 1995]. In contrast, global features are computed inside and outside of the object region. Global features include color [Zhu and Yuille 1996; Yilmaz et al. 2004; Ronfard 1994] and texture [Paragios and Deriche 2002].

An important issue in contour-based methods is the contour initialization. In image gradient-based approaches, a contour is typically placed outside the object region and shrunk until the object boundary is encountered [Kass et al. 1988; Caselles et al. 1995]. This constraint is relaxed in region-based methods such that the contour can be initialized either inside or outside the object so that the contour can either expand or shrink, respectively, to fit the object boundary. However, these approaches require prior object or background knowledge [Paragios and Deriche 2002]. Using multiple frames or a reference frame, initialization can be performed without building region priors. For instance, in Paragios and Deriche [2000], the authors use background subtraction to initialize the contour.

Besides the selection of the energy functional and the initialization, another important issue is selecting the right contour representation. Object contour, _, can be represented either explicitly (control points, **v**) or implicitly (level sets, φ). In the explicit representation, the relations between the control points are defined by spline equations. In the level sets representation, the contour is represented on a spatial grid which encodes the signed distances of the grids from the contour with opposite signs for the object and the background regions. The contour is implicitly defined as the zero crossings in the level set grid. The evolution of the contour is governed by changing the grid values according to the energy computed using Equation (1), evaluated at each grid position. The changes in the grid values result in new zero crossings, hence, a new contour position. The source code for generic level sets, which can be used for various applications by specifying the contour evolution speed, for instance, segmentation, tracking, heat flow etc., is available at LevelSetSrc. The most important advantage of implicit representation over the explicit representation is its flexibility in allowing topology changes (split and merge).

2.4. Supervised Learning

Object detection can be performed by learning different object views automatically from a set of examples by means of a supervised learning mechanism. In context of object detection, the learning examples are composed of pairs of object features and an associated object class where both of these quantities are manually defined. Selection of features plays an important role in the performance of the classification; hence, it is important to use a set of features that discriminate one class from the other. In addition to the features, it is also possible to use other features such as object area, object orientation, and object appearance in the form of a density function, for example, histogram. Once the features are selected, different appearances of an object can be learned by choosing a supervised learning approach. These learning approaches include, but are not limited to, neural networks [Rowleyet al. 1998], adaptive boosting [Viola et al. 2003], decision trees [Grewe and Kak 1995],and support vector machines [Papageorgiou et al. 1998]. These learning methods compute a hypersurface that separates one object class from the other in a high dimensional space. Following we will discuss the adaptive boosting and the support vector machines due to their applicability to object tracking.

2.4.1. Adaptive Boosting: Boosting is an iterative method of finding a very accurate classifier by combining many base classifiers, each of which may only be moderately accurate [Freund and Schapire 1995]. In the training phase of the Adaboost algorithm, the first step is to construct an initial distribution of weights over the training set. The boosting mechanism then selects a base classifier that gives the least error, where the error is proportional to the weights of the misclassified data. Next, the weights associated with the data misclassified by the selected base classifier are increased. Thus the algorithm encourages the selection of another classifier that performs better on the misclassified data in the next iteration.

2.4.2. Support Vector Machines: As a classifier, Support Vector Machines (SVM) are used to cluster data into two classes by finding the maximum marginal hyperplane that separates one class from the other [Boser et al. 1992]. The margin of the hyperplane, which is maximized, is defined by the distance between the hyperplane and the closest data points. The data points that lie on the boundary of the margin of the hyperplane are called the support vectors. In the context of object detection, these classes correspond to object class and nonobject class. SVM can also be used as a nonlinear classifier by applying the kernel trick to the input feature vector extracted from the input. The kernels used for kernel trick are polynomial kernels or radial basis functions.

In the context of object detection, Papageorgiou et al. [1998] use SVM for detecting pedestrians and faces in images. In order to reduce the space, temporal information is utilized by computing the optical flow field in the image. Particularly, the discontinuities in the optical flow field are used to initiate the search for possible objects resulting in a decreased number of false positives.

2. Object Tracking: Object tracking is a major event within the field of computer vision.

We are using high-powered computers, the availability of high quality and inexpensive video cameras. There are three main steps in video analysis: detection of interesting moving objects, tracking of such objects from frame to frame, and analysis of object tracks to recognize their behavior.

We can simplify tracking by motion and/or appearance of objects. Every tracking method requires an object detection mechanism either in all frames or when the object first appears in the video.

The main tracking categories are:

1. **Point Tracking:** Objects detected in consecutive frames are represented by points, and the association of the points is based on the previous object state which can include object position and motion. This approach requires an external mechanism to detect the objects in every frame.

2. **Kernel Tracking:** Kernel refers to the object shape and appearance. For example, the kernel can be a rectangular template or an elliptical shape with an associated histogram. Objects are tracked by computing the motion of the kernel in consecutive frames. This motion is usually in the form of a parametric transformation such as translation, rotation, and affine.
   The kernel trackers can be obtained based on
   —tracking single or multiple objects,
   —ability to handle occlusion,
   —requirement of training,
   —type of motion model, and
   —requirement of a manual initialization.

3. **Silhouette Tracking:** Tracking is performed by estimating the object region in each frame. Silhouette tracking methods use the information encoded inside the object region. This information can be in the form of appearance density and shape models which are usually in the form of edge maps. Given the object models, silhouettes are tracked by either shape matching or contour evolution. Both of these methods can essentially be considered as object segmentation applied in the temporal domain using the priors generated from the previous frames.

CONCLUSION

In this, we present an extensive survey of object tracking and object detection methods .We includes a short discussion on popular object detection methods. The survey on object tracking and object detection with rich bibliography content can give valuable insight into this important research topic and encourage new research.

FUTURE WORK

The Significant progress has been made in object tracking and detection during the last few years. Several trackers have been developed which can track objects simple scenarios. Thus, tracking and detection associated problems are very active areas of research and new solutions are continuously being proposed.

REFERENCES

[1] Alper Yilmaz, Omar Javed, and Mubarak Shah "Object Tracking and Object Detection methods", Dec2006.
[2] Aggarwal, J. K. and CAI, Q. 1999 "Human motion analysis: A Review Compute Vision Image Understand".*73*, 3, 428–440.
[3] Aggarwal, "Segmentation and recognition of continuous human activity". In IEEE Workshop on Detection and Recognition of Events in Video. June 20012, 8–35.
[4] Bar-Shalom Y and Foreman T "Tracking and Data Association", Academic Press Inc, 1988.
[5] Chandrasekhar D.Badgujar1, Dipali P.Sapkal, "Object Detect, Track and Identify Using Video Surveillance" Volume 2, Issue 10, October 2011, PP 71-76.