Proficient Use of Naïve Bayes Classifier in Object Tracking

Varsha S. Futane
PG Student (ME-CSE) Computer Sciences and Engineering Department TPCT's College of Engineering, Osmanabad, (M.S.), India

Anilkumar N. Holambe
Computer Sciences and Engineering Department
TPCT's College of Engineering, Osmanabad, (M.S.), India

***Abstract :-* Real time object tracking is becoming a challenging ingredient in analysis of video imagery for efficient and robust object tracking. This work presents a tracking algorithm based on a set of naive Bayesian classifiers. We consider tracking as a classification problem and train online a set of classifiers which distinguish a target object from the background around it. This paper focuses naïve Bayes classifier approach for tracking a target object in a real-time video dataset. In equivalence to the still images, video sequences render more information on how objects and their scenarios vary overtime. It is always an ambitious task in order to formulate an efficient appearance model. Imprecise extraction of target object and background in model adaptation causes a serious drift problem which leads in degradation of tracking performance. During Pre-processing stages, challenges like illumination, pose variation, occlusion are to be looked upon. This problem can be overcome by continuous detection approach of the target object in each frame. In this approach we formulate a binary classification with the help of a naive Bayes classifier in a compressed knowledge base(domain).**

Keywords- discriminative tracking; naive bayesian classifier; Gaussian function; object tracking; traind sample

I. INTRODUCTION

In recent years visual tracking became an important and fast developing area of computer vision. Object tracking is a critical step in many machine vision applications from video surveillance, traffic monitoring and human-computer interaction to robotics. Trackers are usually designed to track the target object by searching the area in the scene which closely resembles a model of the target. One of the major assumption of tracking algorithms is that the target differs from the background in terms of some (pre-defined) features. In cases where the target resembles the background, the tracking is very difficult, might become unreliable and the tracked target can get lost. The tracking can be improved if not only the properties of the object but also the properties of the background around it are taken into account. Therefore, visual object tracking can be addressed as a binary classification problem [1], [2] where object pixels have to be distinguished from the surrounding background pixels. We follow the general idea of using classifiers for tracking: a classifier is trained to distinguish between the target object and the background. In contrast to existing classifier based tracking methods which are focused on selecting single optimal feature for tracking [1], [3], [4], or which use linear classifiers for feature integration [2], we propose a use of naive Bayesian classifier. This solution provides a straightforward way for integration of various features, without rejecting any information which might be hidden in less informative features. Additionally, it is computationally inexpensive comparing to other machine learning methods, yet capable to produce good classification results. We train a naive Bayesian classifier on a reference image where the object of interest is selected either manually or as a result of previous tracking. Given a new video frame, our naive Bayesian classifier tests pixels and forms a log-likelihood map which we use for tracking.

The usage of only one classifier, trained in the initial stage, when the exact position of the target is known, would be sufficient if the foreground object and background does not change over time. However, the appearance of the foreground object can easily change, because of the change of viewpoint or illumination change. At the same time the appearance of the background around the target can vary as a result of dynamic changes of the background structure or simply because the movement of the target object. Therefore, the classifier has to be dynamically updated in order to remain discriminative and to adapt to variations in the appearance of the target and its background. We solve this problem by maintaining a temporarily updated set of naive Bayesian classifiers. New classifiers, continually trained and added to the set, replace the oldest classifiers in the set. The tracking is done on a joint likelihood map made by combining maps of separate classifiers. Each classifier is optimized to distinguish the foreground object from the background in a particular frame whereas the set of classifiers ensures temporal adaptivity of the tracker. The proposed method for temporal adaptivity of the tracker is similar to the method used in [2], whereas instead of boosting weak linear classifiers, we use voting of naive Bayesian classifiers.

Formulating the tracking as a classification task yields some important problems. Firstly, tracking in dynamic environment cannot relay on a single feature. It has to use a number of different features which have to be appropriately combined. Secondly, as the tracked object moves from place to place, its appearance as well as the appearance of the background around it could change. Therefore, the features used for tracking have to be updated regularly, so that they remain sufficiently discriminative for object/background classification. The proposed approach deals with these two important problems as follows. It offers an easy way for integrating of different features into tracking by using naive Bayesian classifiers. Next, it solves a problem of temporal adaptivity by maintaining a set of classifiers which is updated throughout the time. Rather than directly updating the model of tracked object, as it is usually done in the existing methods [4], [6], we retain some degree of diversity in our target model, represented by a set of classifiers. Results demonstrate that the proposed method successfully track objects of interest in different environment conditions and outperforms existing tracking methods.

II. RELATED WORK

During visual object tracking, when falling across some things, such as strong illumination change, partial occlusion, which may cause object appearance change obviously, tracking will account drift problem, and tracking may fail [11]. In order to solve this problem, researchers propose adaptive appearance model, which may change adaptively as tracking going. Black et al. [12] have proposed to learn a subspace model to represent object offline. IVT [13] tracking uses an incremental subspace model to automatically adapt the change of object appearance. L1 [14] tracker uses a sparse combination of object template and patch template to represent object appearance. These methods have made some success on solving the problem of change of object appearance, but have three shortcuts: Firstly, these methods suppose object appearance would not change obviously during tracking period; Secondly, when sampling some samples around current object location, the appearance model need to adapt misaligned samples, this may cause drift problem; Third, these methods have not used background information.

The temporal update of the target model is often necessary in visual tracking because the appearance of a target tends to change throughout the time. In [11] it is assumed that the initial model remains representative during the tracking. They make a model update so that it represents combination of the initial model and the model associated to the target appearance in the current frame. The extension of this method is used in [14], where the model is updated as a linear combination of target appearance in starting, previous and current frame. Additional anchoring of the target model to the initial one is added by weighting coefficients in linear combination [16]. However, the target object might undergo severe appearance changes as a result of severe illumination change or a change of a the viewpoint. Then, the solution is to update the model regularly as it is done in [12].

In order to use background information in visual object tracking, there has been proposed a new class of visual object tracking algorithm called discriminative tracking algorithm, which treats tracking problem as a binary classification problem. Avidan [15] used support vector machine to classify samples in optical flow based discriminative tracking algorithm and have good gains. Grabner et al. [16] use online boosting algorithm to classify samples in discriminative tracking. These algorithms use current object location as positive samples, and use one positive sample and some negative samples to update the classifier, when appearance model updated adaptively, noise and misaligned instance may occur, this type of method using one positive sample may account drift problem. Babenko et al. [17] use multiple instances learning (MIL) algorithm to track the object in discriminative tracking. Zhang et al. [18] use weighted multiple instances learning (WMIL) algorithm to track the object, and later Zhang et al. [19] propose compressive tracking (CT) algorithm. The classifier in MIL algorithm, WMIL algorithm, and CT algorithm is naive Bayesian classifier, WMIL algorithm makes some improvement on MIL algorithm, and CT algorithm uses just the same classifier.

III. Methodology

A.Naive-Bayes Classifier

A naive-bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms . Our tracking model is generative[13] as the object can be well represented based on the characteristics extracted on the compressive domain. It is also discriminative[16] because we use these features to separate the target from the surrounding background via a naive Bayes classifier. Discriminative algorithms pose the tracking trouble as a binary classification task to find the decision boundary for distinguishing the target object from the background. Avidan[15] broadens the optical flow access with a support vector machine classifier for target object tracking. Collins et al[20] demonstrate that the most discriminative features can be learned online to separate the target object from the background. Grabner et al[16] propose an online boosting algorithm to choose features for tracking. However, these trackers[15–16] only use one positive sample when updating the classifier.

Block Diagram

Video sequence is given as input. It is converted into frames. Sampling is applied to first frame and to the corresponding frames. In processing stage, a series of following processes take place. Classifier, tracks the object in first frame and updater helps to track all other frames.

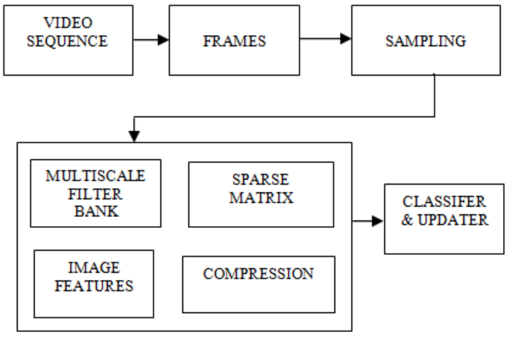

Fig.1. Block Diagram of Compressive Naive-Bayes Classifier

B.Sampling

From the first frame in the video sequence, target object is identified by sampling and it is known as Positive sample. The features away from the centre of target object are identified, few negative samples are considered to separate the object from the background.

C. Multiscale filter bank

In a series of samples extracted from each frame are filtered so that the unwanted samples are eliminated from the total positive and negative samples extracted with the help of multiscale filter bank.

D. Image feature extraction

In this model, features selected by an information preserving and non-adaptive spatiality reduction from the multiscale image feature space based on the compressive sensing theories[21,22]. It has been explained that a small number of randomly generated linear measurements can uphold most of the prominent information and allow almost perfect reconstruction of the signal if the signal is compressible such as natural images [21-23].

E. Sparse matrix

Restricted isometric property (RIP) is satisfied by using a sparse measurement matrix, which facilitates efficient projection from the image feature space.

F. Classifier

The exact location of the target object is pointed with the help of sparse matrix is done by classifier. For the corresponding samples, the classifier is updated till the last frame.

Tracking by Naive Bayesian Classifier

Tracking can be viewed as a problem of probabilistic inference from ambiguous sensor measurements. The local image measurements are combined with a priori information to derive an a posteriori density estimate over the tracking hypotheses. We want to learn a classifier to separate frame pixels into two classes:foreground (target object) and background.

Let C ∈{fg, bg} denote class adherence and let **f** = [f_1, ..., f_n] be a vector of features (image measurements) used for tracking. The classifier estimates the probability of class *C* given the measurement **f** . From Bayes theorem this posterior probability is given as:

$$p(C|f) \alpha\, p(C)p(f|C) \qquad (1)$$

A naive Bayesian classifier assumes that features are independent so the equation (1) can be rewritten as:

$$p(C|f) \alpha\, p(C) \prod_{i=1}^{N} p(f_i|C) \qquad (2)$$

where *N* is the number of features. This assumption is usually over-simplified.However, naive Bayesian classifiers often work much better in many complex real-world situations than one might expect. Recently, careful analysis of the Bayesian classification problem has shown that there are theoretical reasons for the apparently unreasonable efficacy of naive Bayesian classifiers. The details about this topic can be found in [9],[10]. Besides, there are some practical reasons for using this classifier. It is computationally inexpensive, comparing to other machine learning methods, which makes it suitable for tracking. For example, even the simple linear classifier used in [2] requires significantly more computation than the naive Bayesian classifier. If we assume that each pixel is foreground or background pixel, with the same probability, i.e., *p*(*fg*) = *p*(*bg*) than our problem reduces to estimating likelihood *p*(**f** |*C*). Dividing equation (2) written for *fg* class by the same equation written for *bg* class and finding the logarithm of both sides gives the following expression:

$$\log \frac{p(f_g|f)}{p(b_g|f)} = \sum_{i=1}^{N} \log \frac{p(f_i|f_g)}{p(f_i|b_g)} \qquad (3)$$

where the ratio of probability that a pixel belongs to foreground/background is represented in terms of sum of log-likelihood ratios. This expression defines our naïve Bayesian classifier that transforms each video frame into the log-likelihood ratio map which indicate chances that pixels belong to the foreground or background.

The likelihoods *p*(*fi* |*fg*) and *p*(*fi* |*bg*) are estimated directly from the video frames. Given a feature *fi*, let *Hfg i* (*k*) be a histogram of feature values for pixels on the target object and *Hbg i* (*k*) be a histogram for pixels in the background sample, where *k* is in range from 1 to the number of histogram buckets. We use a center-surround approach (shown in Fig. 1) to make histograms of the target object *Hfg i* (*k*) and background *Hbg i* (*k*). The values are sampled from the feature images, where pixel values are values of feature *fi* associated to the same pixel location in original video frame. A rectangular set of pixels covering the object is chosen to represent the target object, while a larger surrounding area is chosen to represent the background. This leads to construction of a discriminative classifier which separates object from background around it.

Fig. 2. Center-surrounding approach for modeling the target object and surrounding background. From the inner box foreground (target) object is modeled, while the background is modeled from the outer ring.

We form an empirical discrete probability distribution *pfg i* (*k*) for the target object, and *pbg i* (*k*) for the background, by normalizing each histogram, i.e. dividing each histogram bin by the number of elements in entire histogram. The empirical estimate of the log likelihood ratio *Li*(*k*) for the particular feature *fi* is given as:

$$L_i(k) = \log \frac{\max(p_i^{fg}(k),\ \delta)}{\max(p_i^{bg}(k),\delta)} \quad (4)$$

where δ is very small number that that prevents dividing by zero or taking the logarithm of zero. The histogram *Li* is an empirical estimate of actual loglikelihood ratio for given feature *fi*:

$$L_i \approx \log \frac{p(f_i, f_g)}{p(f_i, b_g)} \quad (5)$$

As a result we can rewrite equation (3) for the naive Bayesian classifier to be represented in terms of empirical log-likelihood ratios:

$$\log \frac{p(f_g|f)}{p(b_g|f)} = \sum_{i=1}^{N} L_i \quad (6)$$

If the sum of log-likelihood ratios from (6) is bigger than zero the pixel possibly belongs to the target object, while otherwise it is more likely that it belongs to the background. It is possible to make a hard decision about class belonging for each pixel by simple comparison of log-likelihood ratio to zero. However, we are not interested in this hard decision, because we only want to estimate the position of a target, which appears as a peak in log-likelihood map built by applying naive Bayesian classifier from (6) to video frames. We estimate the actual position of the target object by using standard mean shift algorithm on these log-likelihood ratio maps.

Temporal Update of the Target Model

During the visual tracking appearance of the target object as well as the background may easily change. The changes of the object can result from changing of viewpoint, such as when the part of the object facing the camera turns sideways. Similarly, object gradually changes as the object moves from a darker to a brighter part of a scene and vice verse. Also, the target movements result in the changing the background around it. Therefore, it is necessary to update the target model in order to adapt to these changes. It is possible to use the estimate of a target position in each new video frame to extract the new model of the target. However, the simple replacing of the existing model by the new one, could result in misclassification because the tracking might be imperfect and could further lead to eventual tracking failure. Hence, we propose gradual update of a target model. As the tracker follows the target we continuously calculate new log-likelihood ratios corresponding to the current estimate of target position.
As a result new naive Bayesian classifiers are trained to distinguish between new appearance of the foreground and neighboring background. Then, we add the classifiers to the set and make a decision for pixels in a next frame based on voting the classifiers from the set. The final, temporally updated classifier which make a confidence map used for the mean shift tracking in the next frame is a combination of previous *M* classifiers:

$$\log \frac{p^t(f_g|f)}{p^t(b_g|f)} = \sum_{j=0}^{M} \sum_{i=1}^{N} L_i^{t-jT} \quad (7)$$

where superscript denotes the time instance at which the separate classifier is trained. The constant *T* indicates a time distance between training of two consecutive classifiers. The set contains *M* classifiers and when the new one is added, the oldest one is removed from the set. The proposed method gradually updates the target model, whereas a voting mechanism makes the update scheme more robust, because the decision is not based on single classifier. Our solution for the temporal update is most similar to the method used in [2]. However, in contrast to [2] we do not use a set weak linear classifiers (which in fact represents one strong classifier), but a set of mutually independent strong classifiers.

IV. CONCLUSION

We consider tracking as a binary classification problem, where set of a naïve Bayesian classifiers is trained to discriminate the object from the background. The classifiers vote to calculate a confidence map of the next frame. The tracker adjusts to appearance changes of both target object and the background around it, by gradual update of the target model, which is represented by the set of classifiers. The naïve classifier approach continuously trains new classifiers which replace the oldest ones from the set, updating the target model and giving the robustness to the tracker. In some excellent discriminative tracking algorithms, such as MIL, WMIL, CT naive Bayesian Classifier has been used as it is simple but effective.

References

[1]. Collins, R., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1631– 1643 (2005)
[2]. Avidan, S.: Ensemble tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 261–271 (2007)
[3]. Han, B., Davis, L.: Object tracking by adaptive feature extraction. In: Proc. IEEE Conf. Image Processing, vol. 3, pp. 1501–1504 (2004)
[4]. Kwolek, B.: Object tracking using discriminative feature selection. In: Blanc-Talon, J., Philips, W., Popescu,D., Scheunders, P. (eds.) ACIVS 2006. LNCS, vol. 4179, pp. 287–298. Springer, Heidelberg (2006)
[5]. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 564–577 (2003)
[6]. Wang, J., Yagi, Y.: Integrating color and shape-texture features for adaptive real time object tracking. IEEE Transactions on Image Processing 17, 235–240 (2008)
[7]. Stern, H., Efros, B.: Adaptive color space switching for face tracking in multi-color lighting environment.In: Proc. IEEE Int. Conf. on Automatice Face and Gesture Recognition, pp. 249–254 (2002)
[8]. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. IEEE Conf.Computer Vision and Pattern Recognition (2005)
[9]. Zhang, H.: The optimality of naive bayes. In: FLAIRS Conference (2004)
[10]. Kuncheva, L.I.: On the optimality of naive bayes with dependent binary features. Pattern Recognition Letters 27, 830–837 (2006) 11. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Analysis And Machine Intelligence 24, 971–987 (2002)
[11]. Tsagkatakis G., Savakis A. Online Distance Metric Learning for Object Tracking. IEEE Transaction on Circuits and Systems for Video Technology, 2011, 21(12), 1810~1821.
[12]. Black, M., Jepson, A.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. International Journal of Computer Vision 38 (1998) 63-84
[13]. Ross, D., Lim, J., Lin, R., Yang, M H.: Incremental learning for robust visual tracking. International Journalof Computer Vision 77 (2008) 125-141
[14]. MEI, X., LING, H. 2009. Robust visual tracking using L1 minimization. IEEE International Conference on Computer Vision, 1436–1443.
[15]. 15. Avidan, S.: Support vector tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 1064-1072
[16]. 16. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via online boosting. In British Machine Vision Conference (2006) 47-56
[17]. Babenko, B., Yang, M H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 1619-1632
[18]. ZHANG, K., SONG, H. 2012. Real-time visual tracking via online weighted multiple instance learning.pattern Recognition.
[19]. ZHANG, K., ZHANG, L., YANG, M. 2012. Real-time compressive tracking. European Conference on Computer Vision.
[20]. Collins, R., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. PAMI 27, 1631–1643 (2005)
[21]. Donoho, D.: Compressed sensing. IEEE Trans. Inform. Theory 52,1289–1306(2006)
[22]. Candes, E., Tao, T.: Near optimal signal recovery from random projections and universal encoding strategies.IEEE Trans. Inform.Theory 52, 5406–5425 (2006)
[23]. Wright, J., Yang, A., Ganesh, A., Sastry, S., Ma, Y.: Robust face recognition via sparse representation. PAMI 31, 210–227 (2009)