Master Thesis

Visual Object Tracking applying Online Ensemble of multiple trackers

Author:
Jorge Martinez Gomez

Supervisor:
Juan Carlos Niebles

A thesis submitted in fulfilment of the requirements
for the degree of Master of Science

in the

Computer Vision Research Group
Electrical and Electronics Engineering Department

April 2015

"Thanks to my solid academic training, today I can write hundreds of words on virtually any topic without possessing a shred of information, which is how I got a good job in journalism."

Dave Barry

UNIVERSIDAD DEL NORTE

Abstract

ELECTRICAL AND ELETRONICS ENGINEERING DEPARTMENT
Computer Vision Research Group

Master of Science

Visual Object Tracking applying Online Ensemble of multiple trackers

by Jorge Martinez Gomez

The object tracking literature offers a large variety of tracking methods, which exhibit complementary properties in terms of their performance, best usage scenarios and failure modes. In this paper, we introduce a new tracking algorithm based on an online ensemble of tracking algorithms. Our method runs multiple online trackers in parallel and fuses their outputs in an online fashion. The resulting tracker can leverage the strenghts and overcome failures of each individual tracker, producing more robust target tracking. We perform experiments on current object tracking benchmark and show how our ensemble consistently outperforms all trackers in the ensemble, and achieves state-of-the-art object tracking performance.

Acknowledgements

> Not enough people do things that leave others to wonder. RT @BrianMendicino: Wondering why @neiltyson is watching Glee.
>
> Neil deGrasse Tyson

The acknowledgements and the people to thank go here, don't forget to include your project advisor...

Contents

List of Figures

List of Tables

For/Dedicated to/To my...

Chapter 1

Introduction

The goal of visual object tracking is to estimate the state of a target in an image sequence. This is a difficult task, as the target object can be articulated or deformable, the scene illumination can change suddenly, background clutter may introduce distractions that result in tracker drifting, among others. In spite of the multiple challenges, there are many potential applications that make this capability attractive such as activity recognition, motion analysis, human surveillance and robotics.

Many approaches for object tracking have been proposed to cope with some of these challenges. While the state-of-the-art methods achieve relative success, there is still no single approach that is able to handle all challenging situations. For instance, tracking-by-detection methods may not be able to handle scale variations rigorously. On the other hand, generative methods tend to suffer from model drifting and struggle to handle appearance variations.

In this paper, we focus on "model free tracking" of arbitrary objects in videos, in which no prior knowledge other than the object location in the first frame is available. Recently, the online tracking benchmark proposed in [**?**] shows that each tracking algorithm performs best under particular circumstances. There is no single tracking algorithm that can perform well on all sequences in the benchmark. This indicates that each tracking challenge can be addressed better by a different algorithm. In other words, tracking strenghts may be distributed among the available trackers. This is the key observation that inspires our proposed method; we consider a tracking approach that combines the outputs of multiple trackers running in parallel via an online ensemble. This ensemble has the interesting property of leveraging the strenghts of individual trackers, while overcoming the failure modes of each tracker. Since for a new and unseen sequence we do not know which tracker would perform best, our method computes a data-driven

online ensemble that results in improved tracking performance when compared to the results of individual trackers.

In our method, we leverage the observation that only some of the trackers drift into non-target areas of the image in most cases while some of the trackers succeed by focusing on the correct target. Furthermore, our ensemble uses an appearance model that serves as an additional verification mechanism of the tracked region. Using these model components, we identify and exploit the successful trackers to steer failed trackers towards the correct target region. Effectively, our ensemble can correct failed trackers, which ultimately increases tracking performance.

The main contribution of this paper is an ensemble tracking framework that builds on top of the output of available online tracking algorithms running in parallel to produce an online fused tracking result that leverages each tracker best features. Our method does not use prior knowledge about the nature of the trackers in the pool. The fused tracking output is obtained by considering appearance and spatial relations among tracker outputs. In order to cope with trackers weaknesses, our ensemble identifies successful trackers in a data-driven fashion and uses them to steer failed trackers by restarting them asynchronously. This helps to avoid sequence dependent parameters and overtuning.

The rest of the paper is organized as follows. We first briefly review the state-of-the-art of tracking algorithms in Section **??** and then present our online ensemble tracking algorithm in Section **??**. Section **??** illustrates quantitative and qualitative results of our tracker on a standard benchmarking dataset. Finally, we conclude the paper in Section **??**.

Chapter 2

Moving Object Detection Approaches, Challenges, Datasets and Object Tracking

An object can be considered simply as nothing but an entity of interest used for further analysis. These elements can be represented by their shape **Cite here** or appearance **cite color histograms, etc.**. In order to track objects, selecting the right features plays a critical role. In general, the most important property of a visual feature is its uniqueness so that could be easily distinguished from other objects. Mostly features are chosen manually by the user depending on the application domain. this problem of automatic feature selection has receibed significant attention in the pattern recognition community. The most common visual features selections are color, edges, displacement vectors and textures.

Among all features, color is the one of the most widely used feature for tracking. However, color features are sensity to illumination variation. To tackle this problem, in scenarios where this effect is inevitable, other features are incorporated to model object appearance.

2.1 Moving Object Detection

In a video, there are two sources of information that can be used for object detection and tracking: Visual features (color, texture and shape) and motion information. Robust approaches suggest that combining the statistical analysis of visual features and temporal

analysis of motion information. Moving object detection targets the extraction of moving objects that are of interest in sequences (e.g. people and vehicles).

A large number of methodologies have been proposed for object tracking, focusing on the task of object detection first. Most of them apply combinations and intersections among different methodologies, making it very difficult to create a uniform classification of existing approaches. This section classifies different approaches available for object detection from videos.

2.1.1 Background Substraction

Background subtraction is a commonly used technique for object segmentation in static scenarios [3]. This task consist in detecting moving regions by subtracting the current image pixel-by-pixel from a reference background image. The pixels above some threshold are classified as foreground (belongs to an object). The background image is created averaging images over time in an intiialization period, and is updated with new images to adapt to dynamic scene changes. Also, the foreground map is followed by morphological operations such as closing and erosion (elimination of small-sized blobs).

Although background subtraction techniques extracts well most of the relevant pixels, this method is sensitive to changes when some background and foreground pixels have similar value.

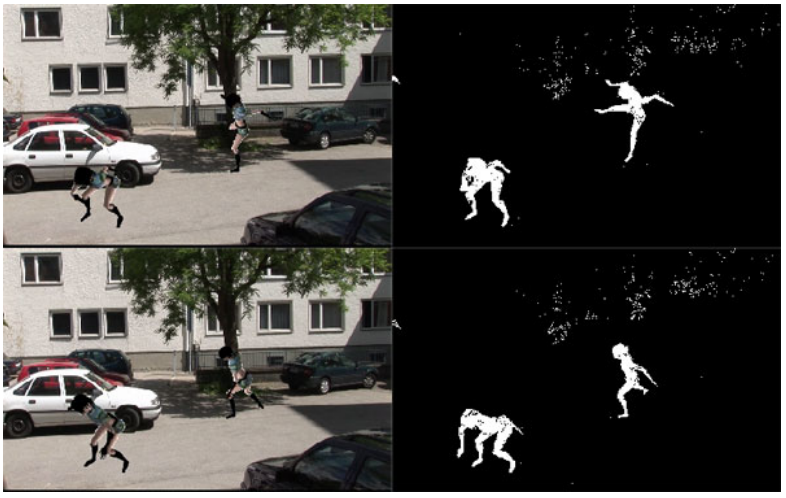


Figure 2.1: Object detection using Gaussian Mixture Models for background subtraction [1]. foreground pixels are drawn in white.

2.1.2 Temporal differencing

In temporal differencing, objects are detected by taking pixel-by-pixel difference of consecutive frames (generally two or three) in a video sequence. This method is most common for moving object detection in scenarios where camera is moving. Unlike static camera scenarios, the background is changing in time for moving camera (not appropiate to create a background model). Alternatively, the moving object is detected by taking the difference between frames $t-1$ and t.

This method is highly adaptive to dynamic changes in the scene as most recent frames are involved in the process. However, it fails detecting small regions as moving objects (ghost regions). Detection will not be correct also, for objects that preserve uniform regions (static objects).

A two-frame differencing method is presented in [4], where the pixels that satisfy the following equation are marked as foreground.

$$|I_t(x,y), I_{t-1}(x,y)| > Th$$

Other methods were developed in order to overcome drastic changes of two frame differencing in some cases. For instance, a three-frame differencing method [5] and a hybrid method that combines three-frame differencing with an adaptive background subtraction model [6].

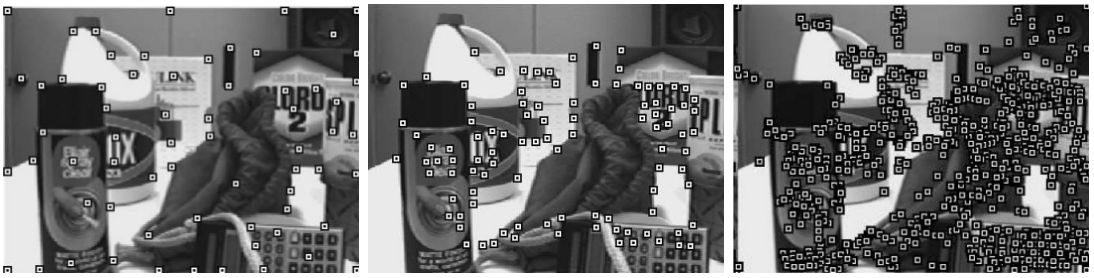
2.1.3 Statistical Approaches

Statistical characteristics of pixels have been used, in order to overcome shortcomings between frames of basic background subtraction methods. The approaches consist in keeping and updating pixels statistics that belong to the background model. Foreground pixels are identified by comparing each pixel's statistics with that of the background model. These methods are becoming more popular due to its reliability in scenes that contain noise, illumination changes and shadows. For instance, some approaches apply Hidden Markov Models (HMM). These methods [2, 7] represent the intensity variation of a pixel in an image sequence as discrete states

The statistical method proposed in [1] describes and adaptive background model for real-time tracking. Every pixel is modeled by a mixture of Gaussians which are udpated online using incoming image data. Then, the Gaussians distributions of the mixture model for each pixel is evaluated in order to detect whether a pixel belongs to foreground and background.

2.1.4 Point detectors

Point detectors are used to find interesting points in objects which have an expressive texture in their respective localities. An interest point should have invariance to changes in illumination and camera viewpoint. One important detector uses optical flow approach [8]. These methods make use of the flow vectors of moving objects over time to detect moving blobs in an image. In this approach the apparent velocity and direction of every pixel in the frame must be computed. Some other methods are SIFT [9] and Harris [10] corners detectors.



2.2 Challenges

Object detection and tracking is still an open research problem in computer vision. A robust, accurate and high performance approach is still a great challenge. The level of difficulty depends on how the object of interest is defined in terms of features. For instance, Using color as object representation method, it is not difficult to identify all pixels with same color as the object. However, there is always a probability of existence a background region with same color information (background clutter). In addition, illumination changes in the scene does not guarantee that the pixel values of an object will be the same in all frames. These variabilities or challenges which are random in object tracking causes wrong object tracking, and are listed below.

- **Illumination Variation (IV):** It is desirable that background model adapts to gradual changes of the appearance of the environment.
- **Scale Variation (SV):** Ratio between initial object size and current object size differs.
- **Occlusion (OC):** Partially or full, occlusion affects the process of computing the background frame. In real life situations, occlusion can occur anytime the object of interest passes behind another object with respect to a camera.

- **Dynamic background:** Some scenery regions contain movement, but should be still remain as background, acoording to their relevance. Such movement can be periodical or irregular, causing blurring (motion blur - MB), e.g. traffic lights, waving trees).
- **Out of view (OV):** Some portion o the target leaves the view.
- **Background clutter (BC):** As stated before, this challenge makes the segmentation task difficult. It is hard to create ans separate background model from moving foreground objects.
- **Fast Motion (FM):** The speed of a moving object plays an important role in its detection and track. If an object is moving too slow, the temporal differencing methods fails to detect object, because it preserves uniform region between frames. In the other case, fast moving object leaves ghost regions in a detected foreground model.
- **Object rotation and deformation (DEF):** Since natural objects move freely, they can appear slightly or completely transformed. Such rotations, in (IPR) or out (OPR) of plane on the images affect object tracking considerably.
- **Low Resolution (LR):** Number of pixels inside the object bounding box is less than 400.

2.3 Tracking Datasets

In computer vision, a *dataset* could be defined as a collection of images or video sequences used for testing algorithms. The amount of data and characteristics presented in the list, depend on the field that is studied. For instance, in scene recognition, a dataset contains images of landscapes or outdoor environments. Generally, this collection is shared between researchers and plays an important role in comparison and evaluation of state-of-the-art approaches.

The Surveillance Performance Evaluation Initiative (SPEVI) can be used for evaluating algorithms for surveillance-related applications. The first dataset contains 5 sequences applied to single person/face detection and tracking. The second dataset applies for multiple person/face detection and tracking. The sequences contain four targets occluding each other repeatedly. ETISEO dataset contains indoor and outdoor scenes, such as corridors, buildings entries, etc. This dataset can be used for surveillance applications.

PETS dataset became a surveillance project whose challenging scenarios are focused only on high level applications of this field. Some issues, like illumination or scale changes

are not considered in these videos. Most of the sequences are used for person/vehicle tracking in outdoor environments(subway stations, building entrances). CAVIAR is a dataset used generally for situation recognition systems. However, sequences can be applied for tracking evaluation methods. Includes videos of people walking alone, meeting other people, entering and exiting shops.

The VIdeo Surveillance Online Repository (VISOR) database covers a wide range of scenarios and situations, including videos for human action recognition, outdoor videos for face detection, indoor videos for people tracking with occlusions, vehicles detection and surveillance. The VIdeo Surveillance Online Repository, includes several sequences for two separate tasks: First, an abandoned baggage scenario and second, a parked vehicle scenario.

In generic visual tracking, a dataset is a collection of videos that contains and object moving in some scenario. The sequences vary in length from hundreds of frames to thousands. Diverse object types are used. Different scene settings (indoor or outdoor, static or moving camera). Also different challenges, such as object occlusions or illumination conditions are presented. Most commonly used tracking benchmarks are summarized in table 2.1. Recently, the authors in [11] released a benchmark containing 50 most commonly used sequences from some datasets mentioned 2.1, to facilitate fair performance evaluation. Also for better evaluation and analysis of strengths and weakness of tracking approaches. They classified sequences, considering a object tracking challenge, as a category, constructing several subsets to report specific challenging conditions. Some attributes occur more frequently, and some sequences are annotated with several attributes.

Name/Author/Paper	Sequences
Babenko	3
Bobot	12
Cehovin	5
Ellis IJCV2011	3
Godec	7
Kalal	10
Kwon	4
Kwon VTD	11
PROST	4
Ross	4
Thang	4
Wang	4

Table 2.1: Popular object tracking datasets

2.4 Object Tracking

The goal of an object tracker is to generate an object path over time. This trajectory consists of the object position over time in every frame of the video. The tracker may provide complete region in the image that is occupied by the object at every time instant. Certainly, this list is not meticulous and covers popular approaches on each category.

2.4.1 Point Tracking

Tracking can be formulated as the correspondence of objects represented by points across frames. This category can be divided into two subcategories:

Deterministic Methods: These approaches for point correspondence define a cost of associating each object in frame $t-1$ to a single object in frame t using motion constraints, such as proximity, velocity, rigidity and motion. Minimization of the correspondence cost is formulated as a combinatorial optimization problem. A solution, which consists in one-to-one correspondence among all possible associations, can be obtained by optimal assignment methods. For instance Hungarian Algorithm [12] or greedy search methods.

Statistical methods for Point Tracking: Statistical correspondence methods solve tracking problems whose measurements obtained from video sensors contain nose, or object motion can undergo random perturbations. These approaches take measurements and model uncertainties into account during object state estimation. Applying state space approach to model the object properties such as position, velocity and acceleration. In single object state estimation, tYhe optimal state of an object is giben by the Kalman Filter [13, 14], assuming measurement noise have a Gaussian distribution. In the general case, that is, object state is not assumed as Gaussian, estimation can be performed using particle filters [2, 15].

In the case of multiobject data association, state estimation using Kalman or particle filters, it is necessary to solve first correspondence problem before these filters can be applied. However, in cases when two objects are close each other, the correspondence could be incorrect. Then, an incorrectly associated measurement can cause the filter to fail to converge. In order to tackle this problem, Joint Probability Data Association Filtering (JPDAF) [16] and Multiple Hypothesis Tracking (MHT) [17] are two used techniques for data association.

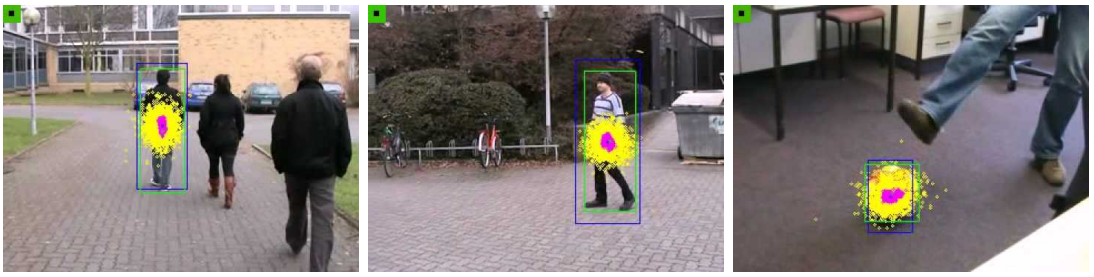


Figure 2.2: [2]

2.4.2 Kernel Tracking

In this type of tracking, object motion is computed using representations of a primitive object region, from one frame to the next. These algorithms differ in terms of appearance representation (features extraction) used, the number of objects tracked, and the method used for object motion estimation.

Density-based tracking: According to [18], the object is modelled with one or more probability density functions, such as Gaussian, mixture of Gaussian, Parzen windows or histogras, that describe the probability of object appearance. Mean-shift is an approach to feature space analysis. This method shifts a data point to the average of data points in its neighborhood. Mean shift uses fixed color distribution. A similar approach is called CAMSHIFT [19] that handles dynamically changing color distribution by adapting the search window size and computing color distribution in the search window.

Template-based tracking: These approaches apply templates of the object to calculate appearance probability on every frame of the video sequence. The most common is *Template matching* [20] that searchs accross the image, a region similar to the object template, defined in previous frames. The similarity measure is calculated using normalized cross correlation. A limitation of this method is its high computational cost due to brute force search. To reduce this cost, some methods limit the object search to a neighborhood near previous position.

Instead of templates, other object representations can be used for tracking. For example, color histograms or mixture models can be computed using the appearance of pixels inside the rectangular or ellipsoidal regions. To reduce computational complexitu, the similarity between object model and the hypothesized position, is computed evaluating the ratio betweem color means between model and position. The position with highest ratio is selected as current object location.

"Tracking by detection" or "Tracking by repeated recognition" [21] systems generally perform target object appearance learning. These methods are closely related to object detection (an area with great progress in computer vision) and has encouraged some

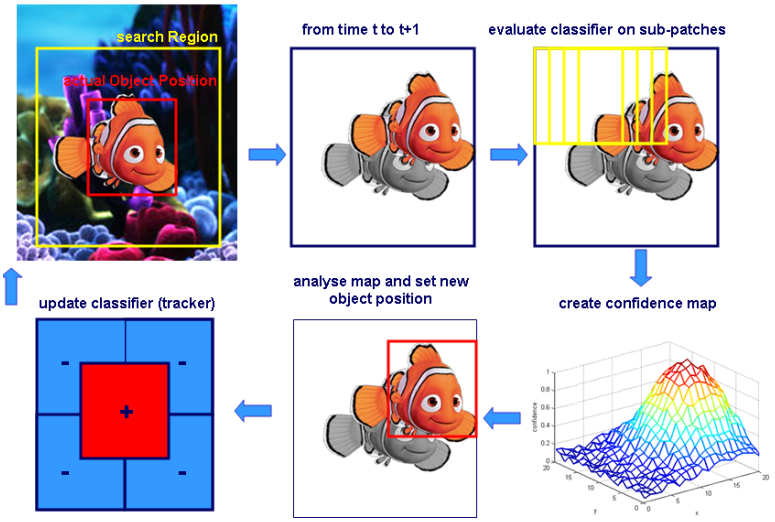


Figure 2.3: Overview for online boosting object tracker

successful real-time tracking algorithms [22, 23]. However, many tracking algorithms employ static appearance models that are defined manually or trained at the first frame only [24–28], these methods are often unable to deal with significant appearance changes. This situations are difficult when there is limited knowledge of the object of interest. In order to cope this problem, an adaptive appearance model that changes during the tracking process as the appearance of the object changes, gets better results [29–31].

Boosting has been used in a wide field of machine learning tasks and applied to computer vision problems. Many tracking algorithms are based on the boosting framework [32] and is related to the work on Online Adaboost [33–35], multi-class boost [36] and MILBoost [37]. The goal of boosting is to combine many weak classifiers (usually decision stumps) into a linear strong classifier.

2.4.3 Silhouette Tracking

The object is tracked via estimation of the object region in each frame. Silhouette-based methods provide an accurate shape description for the objects that are tracked. These approaches can be divided into two main categories, shape matching and contour tracking. Shape matching [38] approaches search object silhouette in the current frame. Contour based, evolve initial contour to its new position in the current frame using state space models or direct minimization of some energy function [39].

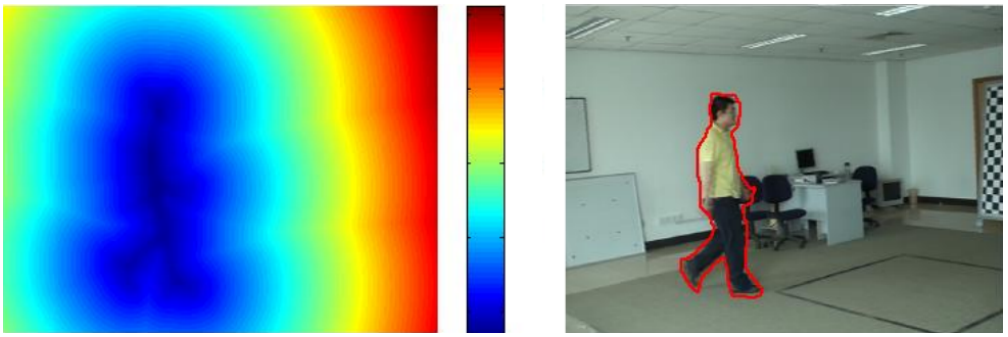


Figure 2.4: Illustration of an active contour representation. Left subfigure shows the signed distance map of a human contour; right image displays contour result.

2.4.4 Tracking applying fusion of trackers or features

Generic object tracking is a defying problem in computer vision. Each tracking method perform well on different sequence than others. However there is not absolute theory about which tracker is the best of all. So, many approaches have been proposed in order to combine different trackers to outperform each individual tracker.

The first algorithm that explicitly applies ensemble methods to tracking-by-detection is shown in [33]. Considering tracking as a binary classification problem, the author extended the work of [40] using the Adaboost algorithm to combine a set of weak features and update object model with online update strategy. The classifier is then used to classify pixels in the next frames as either related to the object or background, obtaining a confidence map.

The authors in [41], combined a template-based tracker, optical flow tracker, and online-random forest tracking-by-detection method into a cascade of trackers. The best selection is summarized into a simple set of rules. The authors explain that augmenting or updating in a smart way a simple online learner in terms of adaptivity can lead to much better results.

Another active fusion approaches are VTD [42] and VTS [43]. In these articles, the approaches obtain several samples of target and trackers states during sampling process using a Markov Chain Monte Carlo method. The trackers are sampled by proposing appearance models, motion models, state representation types, and observation types. Then, the sampled trackers run in parallel and interact with each other, covering target variations.

The authors in [44] proposed a classifier ensemble framwork that uses Bayesian estimation theory to estimate the non-stationary distribution of sampled classifers. In contrast

with general tracking-by-detection classifiers, the weight vector that combines the classifiers is treated as a random variable and the posterior distribution of this vector is treated using Bayes' theory.

In [45] and [46], the authors present an approach that merges the result of different tracking algorithms to produce a better tracking result. Based on the idea of attraction fields, which means the closer a fusion candidate is to a tracking result box, the stronger it is attracted by it. The result that maximizes the attraction of all trackers is chosen as a global result. In [46] present different variants of this method, including a weighted combination of trackers and an approach that favors continuos trajectories throughout the sequence.

Chapter 3

Proposed Approach

This section gives a detailed description of the implemented tracking system. Initially, giving an overview of the whole system and the basic functional blocks. Then, explaining implementation details of the different parts.

3.1 The Tracking Loop

Initially, we explain the necessary building blocks of an object tracking system.

3.2 Tracking system overview

In order to tackle all the problems stated in the previous section, this tracking approach is separated into different modules.

3.2.1 Basic overview

Generic single-object tracking could be defined as the localization of an object through a video sequence. It is generic because the system is able to track any kind of object (faces, cars, etc.), and is single because the system will track just one object and not many at the same time. The proposed approach in this thesis is shown in 3.1. Initially, our method starts with a pool of n trackers $T = \{t_1, t_2, ..., t_n\}$ and input data x. This input corresponds to the initial rectangular box for an object in a sequence. All trackers are initalized and an updateable object model is created. Then, on each frame, our method runs and groups all trackers results by position into a set of m clusters $C = \{c_1, c_2, ..., c_m\}$. Also, we obtain a similarity measure which compares an actually

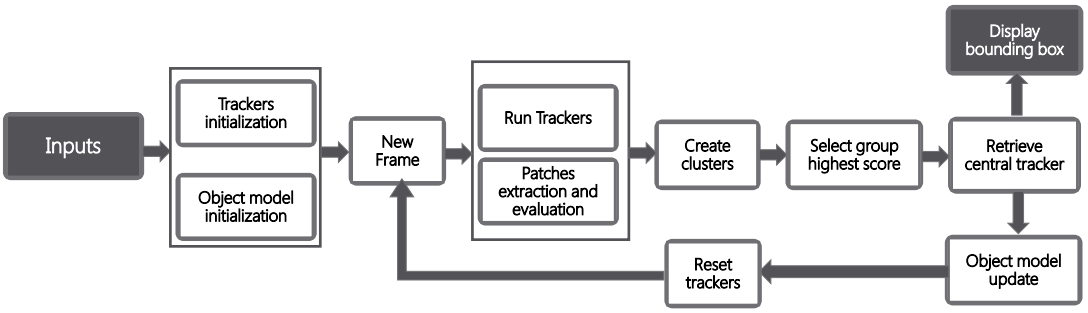


Figure 3.1: General schema for generic single-object tracking proposal.

tracking result patch to the current object model. The output should be the probabilities of similarity between the object model and each tracker result in that frame $S = \{s_1, s_2, ..., s_n\}$. Using clustering information, the system can select between the winner cluster $c*$ which has the highest number of members, or the cluster with highest similarity measure. the others clusters are considered outliers and are reinitialized each 15 frames.

3.2.2 Trackers bounding boxes clustering

Cluster analysis is the formal study of algorithms and methods for grouping, or classifying objects. These objects are described as a set of measurements or by relationships between the object and other objects. A *cluster* is comprised of a number of similar objects collected or group together. Other authors define a cluster as a set of entities which are alike, and entities from different clusters are not alike, or "A cluster is an aggregation of points in the test space such that the *distance* between any two points in the cluster is less than the distance between any point in the cluster and any point not in it". This theory is taken from **Jane and Dubes.**

Cluster analysis is the process of classifying objects into subsets that have meaning in the context of a particular problem. The objects are thereby organized into an efficient representation that characterizes the data. Clustering methods require that an index of proximity, or alikeness, or affinity, or association be established between pairs or patterns. A *proximity matrix* $|d(i, j)|$ accumulates the pairwise indices of proximity in a matrix in which each row and column represents a pattern. Diagonal entries of a proximity matrix are ignored since all paterns are assumed to have the same degree of proximity with themselves. Also it is assmed that all proximity matrices are symmetric, so all pairs of objects have the same proximity index, independent of the order in which they are written.

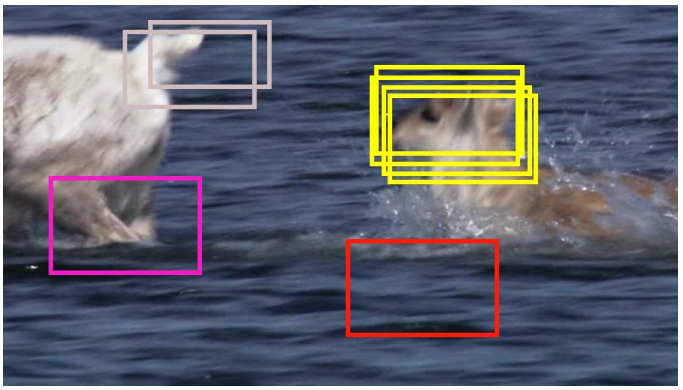


Figure 3.2: Tracking results clustering.

A proximity index is either a *similarity* or a *dissimilarity*. The more the ith and jth objects are similar one another, the larger a similarity index and the dissimilarity index are.

Previous works show the common approach of data fusion using majority voting. The authors in [45] applied this method using a threshold parameter that defines if two result boxes vote for the same position. However, in [46], the authors proved that this approach is sequence dependent. Instead, they settle the idea of attraction fields. We base our approach using the idea of clustering position. On a new frame, each tracker will give a rectangular box of where the object might be. Using this information, we are able to form groups of trackers that have similar positions. For each tracker result, we calculate the distance between its position and the rest of trackers running. The distance d between two boxes b and c is computed as:

$$d(b,c) = 1 - \frac{b \bigcap c}{b \bigcup c} \tag{3.1}$$

Using all distances, we construct a symmetric $l \times l$ proximity matrix D. We take the proximities to be dissimilarities. This means that $d(i,i) = 0$ for all i. We use complete-link hierarchical agglomerative clustering to form groups of trackers. Trackers t_1 and t_2 are "related" if their dissimilarity is below some threshold v. CL merges clusters in order of proximity; the closest clusters will be merged first, and the furthest clusters will be merged last. At each merge, CL creates a *reduced proximity matrix*, with one less row and column. At the end, the algorithm delivers a set of clusters with size m $C = \{c_1, c_2, ..., c_m\}$ satisfying the following:

- $c_i \cap c_j = \emptyset$ for i and j from 1 to m, $i \neq j$
- $c_1 \cup c_2 \cup ... \cup c_m = T$

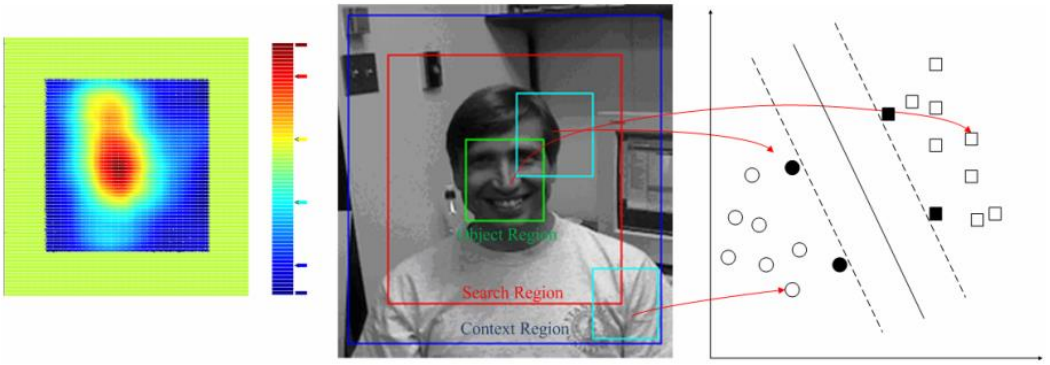


Figure 3.3: Example of tracking-by-detection approach based on SVM classification. The left subfigure shows the score map of face and non-face classification; middle subfigure shows the search region for object localization and the context region for face and non-face samples selection; right subfigure plots the classification hyperplane that separates face and non-face classes.

3.2.3 Object modeling

Visual object tracking has been formulated as a tracking-by-detection problem recently. In this case, object modeling is dynamically performed to support object detection in all frames. Mostly all approaches can be classified into two categories: *Generative appearance models*, that mainly focus on how fit data into their correspondent object class; and *discriminative appearance models*, that assume object tracking as a binary classification issue. The main goal is to maximize the separability between object and non-object regions discriminately.

Generally, Discriminative methods train a classifier using data acquired from previous frames, and subsequently use the trained classifier to evaluate possible object regions at the current frame (Figure 3.3). After localization, a set of *positive* and *negative* samples are heuristically selected to update the classifier. Some approaches apply online boosting **Cites**, that make a discriminative evaluation of features taken from a candidate feature pool, and then select the top ranked features to conduct the tracking process. Other methods apply Support Vector Nachines (SVM) method, which learns a margin-based discriminative appearance model, in order to maximize inter-class separability. It is important to note that these classifiers are trained using visual representations of the object.

Tracking methods applying object appearence online learning have been recently applied. These systems discriminate object from its surrounding background through all the sequence using a classifier updated based on tracking results. However, these approaches except for MILBoost [37], suffer from label jitter. The authors in [**?**] explain that problem of label jitter arises if the bounding boxes of an object are not perfectly aligned

with the target, although it is detected correctly. If label jitter occurs repeatedly over a tracking sequence, the tracker will most likely start to lose the target object. To cope this problem, we consider selecting a bag of trackers which share common similarity with appearance model.

The classifier corresponds to a standard linear SVM, which is trained with a buffer of 10 positives and 100 negative examples. The positive buffer is initialized using x. We extract and image patch and calculate hog features. For the negative buffer we sample random bounding boxes with the same size on the image. During tracking, whenever a new example is added to the buffer, the classifier is retrained.

3.2.4 Best cluster selection

After perfoming clustering stage and obtaining classification scores with each tracker results, we can select the cluster that best follows the object. We present two selection criterias for best cluster.

Best cluster selection based on members criteria: Once groups are formed, we search the cluster c^* in the list of clusters C of size m, with highest number of trackers.

$$c^* = \arg\max_{c \in C} \sum_{t \in c} t_i \tag{3.2}$$

Best cluster selection based on appearance scores: In this case we are considering classification scores for each cluster. Each cluster gives a score per cluster. Then, we select the winner cluster whose score is the highest one.

$$c^* = \arg\max_{c \in C} m_i \tag{3.3}$$

where $m_i \in M = \{m_1, m_2, ..., m_m\}$ corresponds to maximum score value of c_i cluster:

$$M = \left\{ \max_{s \in S} s_i \in c_i \right\} \tag{3.4}$$

3.2.5 Best tracker selection

The centered position x* is selected by choosing the minimum sum of distances of the tracker, to the rest of trackers that belong to c^*. We did not consider selecting the

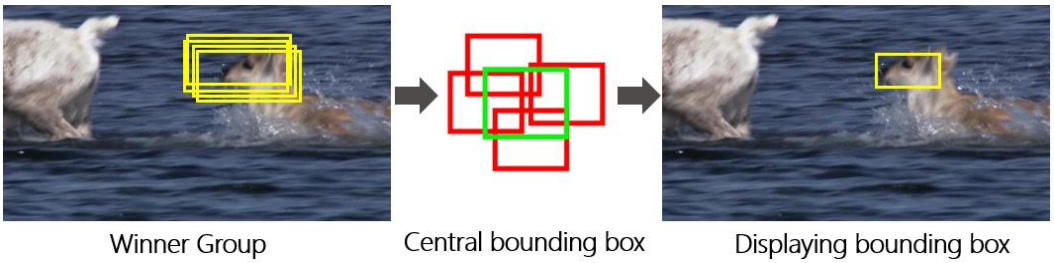


Figure 3.4: Best tracker bounding box selection.

tracker with best appearance score in order to avoid label jitter and drifting problems. Also selecting this value might make the tracker shaky.

$$\mathrm{x}^* = \arg\min_{\mathrm{x}} \sum_{\mathrm{x}_i \in w^*} d(\mathrm{x_i}, \mathrm{x_j}), \quad i \neq j, \mathrm{x_j} \in c^* \tag{3.5}$$

3.2.6 Trackers reset

The outliers correspond to those trackers that does not belong to c^*. These trackers are reinitialized using x^*. Also, from x^*, we crop and image patch and extract hog features as new positive sample for the classifier.

$$R = t_i \not\subset c^* \tag{3.6}$$

3.2.7 Object model update

Chapter 4

Experiments and results

4.1 Experimental setup

Our approach is implemented in native Matlab. The experiments are performed on an Intel Core i7 CPU with 16 GB RAM. We put our tracker to the test by using the recent benchmark [**?**] that includes 50 sequences. The sequences used in our experiments pose challenging scenarios that include situations such as motion blur, illumination changes, scale variation, occlusions, in-plane and out-plane rotations, object deformation, background clutter and low resolution. We are encouraged to build a generic algorithm that can perform well in different scenarios.

4.2 Evaluation Methodology

To validate the performance of our proposed approach, we follow the one-pass evaluation methodology (OPE) proposed in [**?**]. For performance criteria, we chose for our evaluation, the precision and success plots. In *precision*, a frame may be considered correctly tracked if the predicted target center is within a distance threshold of ground truth. In [**?**], the authors explain that plotting the precision for all thresholds, no parameters are required. This makes the curves unambiguous and easy to interpret. A higher precision at low thresholds means the tracker is more accurate, while a lost target will not achieve perfect precision on a large threshold range. The chosen theshold is 20 pixels, that is done in previous works [37**?** **?**]. *Success* measures the overlap between a tracking and a ground truth box and checks where the overlap exceeds a threshold $t \in 0, 1$:

$$O(a, b) = \frac{|a \bigcap b|}{|a \bigcup b|} \tag{4.1}$$

Overlap penalizes if the size of a tracking box is different to the ground truth and the error will not increase if the object is lost. Different to precision, in success the trackers are ranked using the area under the curve (AUC), which means the average overlap over all frames.

4.3 Experiments with sequence attributes

The videos in the benchmark dataset are organized and selected with attributes, which describe the conditions where a tracking algorithm might fail - e.g., occlusion, object deformations. These properties are useful for diagnosing tracking behavior, without the need of analyzing each video separately. Tables **??** and **??** presents a more specific quantitative attribute-based evaluation. Our approach performs favorably on 8 of 11 attributes, and outperforms state-of-the-art algorithms in *deformation*, *out-of-plane rotation* and *scale variation*.

Bibliography

[1] Vu Pham, Phong Vo, Vu Thanh Hung, and Le Hoai Bac. GPU Implementation of Extended Gaussian Mixture Model for Background Subtraction. *2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, pages 1–4, 2010.

[2] J Rittscher, J Kato, S Joga, and A Blake. A probabilistic background model for tracking. *Computer Vision—ECCV 2000*, pages 336–350, 2000.

[3] Am McIvor. Background subtraction techniques. *Proc. of Image and Vision Computing, ...*, 2:13, 2000.

[4] A J Lipton, H Fujiyoshi, and R S Patil. Moving target classification and tracking from real-time video. *Proceedings Fourth IEEE Workshop on Applications of Computer Vision WACV98 Cat No98EX201*, 98:8–14, 1998. ISSN 09031936.

[5] Liang Wang, Weiming Hu, and Tieniu Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36:585–601, 2003. ISSN 00313203.

[6] Robert T Collins, Alan J Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, and Lambert Wixson. A System for Video Surveillance and Monitoring, 2000. ISSN 19406029.

[7] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J.M. Buhmann. Topology free hidden Markov models: application to background modeling. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, 1, 2001.

[8] Jianbo Shi Jianbo Shi and C. Tomasi. Good features to track. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, 1994. ISSN 1063-6919.

[9] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. pages 1–28, 2004.

[10] C. Harris and M. Stephens. A Combined Corner and Edge Detector. *Procedings of the Alvey Vision Conference 1988*, pages 147–151, 1988. ISSN 09639292.

[11] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online Object Tracking: A Benchmark. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, (Iccv): 2411–2418, June 2013.

[12] Zhen Qin and Christian R. Shelton. Improving multi-target tracking via social grouping. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1972–1978, 2012.

[13] Xiaofeng Ren. Finding people in archive films through tracking. In *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008.

[14] Janne Heikkilä and Olli Silvén. A real-time system for monitoring of cyclists and pedestrians. *Image and Vision Computing*, 22:563–570, 2004. ISSN 02628856.

[15] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A Boosted Particle Filter : Multitarget Detection and Tracking. *Proceedings of the 8th European Conference on Computer Vision - ECCV 2004*, pages 28–39, 2004. ISSN 03029743.

[16] Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B. Cremers. People Tracking with Mobile Robots Using Sample-Based Joint Probabilistic Data Association Filters, 2003. ISSN 02783649.

[17] Mohd Asyraf Zulkifley, Bill Moran, and David Rawlinson. Robust hierarchical multiple hypothesis tracker for multiple object tracking. In *Proceedings - International Conference on Image Processing, ICIP*, pages 405–408, 2012.

[18] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:790–799, 1995. ISSN 01628828.

[19] David Exner, Erich Bruns, Daniel Kurz, Anselm Grundhöfer, and Oliver Bimber. Fast and robust CAMShift tracking. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, pages 9–16, 2010.

[20] Simon Korman, Daniel Reichman, Gilad Tsur, and Shai Avidan. FasT-match: Fast affine template matching. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2331–2338, 2013.

[21] Greg Mori and Jitendra Malik. Recovering 3D human body configurations using shape contexts. *IEEE transactions on pattern analysis and machine intelligence*, 28:1052–1062, 2006. ISSN 01628828.

[22] Xiaoming Liu and Ting Yu. Gradient feature selection for online boosting. In *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

[23] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-Time Tracking via On-line Boosting. *Technology*, 1:1–10, 2006. ISSN 0162-8828.

[24] M. Isard and J. MacCormick. BraMBLe: a Bayesian multiple-blob tracker. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, 2, 2001.

[25] Vincent Lepetit and Pascal Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1465–1479, 2006. ISSN 01628828.

[26] Michael J Black and Allan D Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *International Journal of Computer Vision*, 26:63–84, 1996. ISSN 0920-5691.

[27] D Comaniciu, V Ramesh, and P Meer. Real-time tracking of non-rigid objects using mean shift. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:142–149, 2000. ISSN 01628828.

[28] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Robust fragments-based tracking using the integral histogram. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 798–805, 2006.

[29] David a. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision*, 77:125–141, 2007. ISSN 0920-5691.

[30] Iain Matthews, Takahiro Ishikawa, and Simon Baker. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:810–815, 2004. ISSN 01628828.

[31] A D Jepson, D J Fleet, and T F El-Maraghi. Robust online appearance models for visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1296–1311, 2003. ISSN 0162-8828.

[32] Y Freund and R E Schapire. A Decision-theoretic Generalization of On-line Learning and an Application to Boosting. *Journal of Computing Systems and Science*, 55:119–139, 1997. ISSN 00220000.

[33] Shai Avidan. Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:261–271, 2007. ISSN 01628828.

[34] Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised on-line boosting for robust tracking. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5302 LNCS, pages 234–247, 2008.

[35] NC Oza and S Russell. Online ensemble learning. *AAAI/IAAI*, 6837:1109–1109, 2000.

[36] Amir Saffari, Martin Godec, Thomas Pock, Christian Leistner, and Horst Bischof. Online multi-class LPBoost. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3570–3577, 2010.

[37] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual Tracking with Online Multiple Instance Learning. *IEEE transactions on pattern analysis and machine intelligence*, pages 983–990, 2010. ISSN 1939-3539.

[38] Baoxin Li, Rama Chellappa, Qinfen Zheng, and Sandor Z. Der. Model-based temporal object verification using video. *IEEE Transactions on Image Processing*, 10: 897–908, 2001. ISSN 10577149.

[39] Daniel Cremers and Christoph Schnörr. Statistical shape knowledge in variational motion segmentation. *Image and Vision Computing*, 21:77–86, 2003. ISSN 02628856.

[40] Robert T. Collins, Yanxi Liu, and Marius Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1631–1643, 2005. ISSN 01628828.

[41] Jakob Santner, Christian Leistner, Amir Saffari, Thomas Pock, and Horst Bischof. PROST: Parallel robust online simple tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 723–730, 2010.

[42] Junseok Kwon and Kyoung M. Lee. Tracking of a non-rigid object via patch-based dynamic appearance modeling and adaptive basin hopping monte carlo sampling. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, pages 1208–1215, 2009.

[43] Junseok Kwon and Kyoung Mu Lee. Tracking by sampling trackers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1195–1202, 2011.

[44] Qinxun Bai, Zheng Wu, S Sclaroff, M Betke, and C Monnier. Randomized Ensemble Tracking. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2040–2047, 2013.

[45] Christian Bailer, Alain Pagani, and Didier Stricker. A user supported tracking framework for interactive video production. *Proceedings of the 10th European Conference on Visual Media Production - CVMP '13*, pages 1–8, 2013.

[46] Christian Bailer, Alain Pagani, and Didier Stricker. A Superior Tracking Approach: Building a strong Tracker through Fusion. In *European Conference on Computer Vision*, pages 170–185, 2014.