Universidad del Norte

Master Thesis

Visual Object Tracking applying 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

March 2015

Declaration of Authorship

I, John Smith, declare that this thesis titled, 'Thesis Title' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- lelele
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"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

UNIVERSITY NAME (IN BLOCK CAPITALS)

Abstract

Faculty Name
Department or School Name

Doctor of Philosophy

Thesis Title

by John Smith

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

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

Abbreviations

LAH **L**ist **A**bbreviations **H**ere

Physical Constants

Speed of Light $c = 2.997\ 924\ 58 \times 10^8\ \mathrm{ms}^{-\mathrm{S}}$ (exact)

Symbols

a	distance	m
P	power	W (Js^{-1})
ω	angular frequency	rads^{-1}

For/Dedicated to/To my...

Chapter 1

Introduction

Visual object tracking is an important problem in computer vision. This field has a wide range of applications such as surveillance,en success building trackers for specific object classes, a generic object tracker represents a challenging task. In general case, tracking an arbitrary object in an unknown scenario is still considered unsolved. Common challenges are for example, object deformations, illumination human-computer interaction and motion analysis. Althought there has be changes, partial and complete occlusions, drifting, background clutter and similar objects in the scene. This particular situations make some trackers better than others. However, there is no algorithm that has mastered all possible problems that a scenario might generate.

Recently, the evaluation performed in [**?**] shows, each tracking algorithm performs well on different sequences. This explains that different tracking algorithms avoid challenges that can occur in general object tracking. We consider an approach that combines the virtues of different algorithms while evading their weaknesses could outperform each single algorithm. Just as "two heads are better than one", making trackers perform this task together in an unknown scenario, may result in a higher level of performance and achievement, than could be obtained individually. This is what in psychology states as "positive interdependence", the ability of group members to encourage and facilitate each other's efforts [**?**].

1.1 Goals

In this paper, we focus on the problem of tracking an arbitrary object in videos, with no prior knowledge other than its location in the first frame, also known as "model free tracking". We are motivated to link trackers together so one cannot succeed, unless all

group members succeed. A common tracking system consists of an appearance model, which can evaluate the likelihood of the object of interest at a given location. A motion model, that stores and contains the locations of the object over time. Finally, a search strategy to find the best location in the current frame [**?**].All methods share the same goal, meaning that each tracker's individual "effort" is required and is indispensable for group success. Using these models, we make trackers correct each other, increasing performance and ensuring the group is united to a common goal, a concrete reason of being, a purpose for existence.

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 [1]. 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.

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 [2], 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 [3] and a hybrid method that combines three-frame differencing with an adaptive background subtraction model [4].

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 **Cite here!**.

The statistical method proposed in **Cite here** describes and adaptive background model for real-time tracking. Every pixel is modeles 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. 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.

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.

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.

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

Recently, the authors in **benchmark** 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.

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 **cite Here!** 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, 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.

  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) and Multiple Hypothesis Tracking (MHT) are two used techniques for data association.

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 **Cite master thesis**, 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 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* 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" [**?**] 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 successful real-time tracking algorithms [**?** **?**]. However, many tracking algorithms employ static appearance models that are defined manually or trained at the first frame only [**?** **?** **?** **?** **?**], 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 [**?** **?** **?**].

  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 [**?**] and is related to the work on Online Adaboost [**?** **?** **?**], multi-class boost [**?**] and MILBoost [**?**]. 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 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 functional.

2.4.4 Tracking applying fusion of trackers or features

Tracking algorithms fusion can be performed actively, that means that each tracker receives feedback; or passively without feedback. The first algorithm that explicitly applies ensemble methods to tracking-by=detection is shown in [**?**]. The author extended the work of [**?**] using the Adaboost algorithm to combine a set of weak features and update object model with online update strategy. The authors in [**?**], 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. Another active fusion approaches are VTD [**?**] and VTS [**?**]. In this articles, 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. The authors in [**?**] proposed a classifier ensemble framwork that uses Bayesian estimation theory to estimate the non-stationary distribution of sampled classifers. In [**?**] and [**?**], the authors present a passive approach based on the idea of attraction fields. The closer a fusion candidate is to a tracking result box, the stronger it is attracted by it.

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

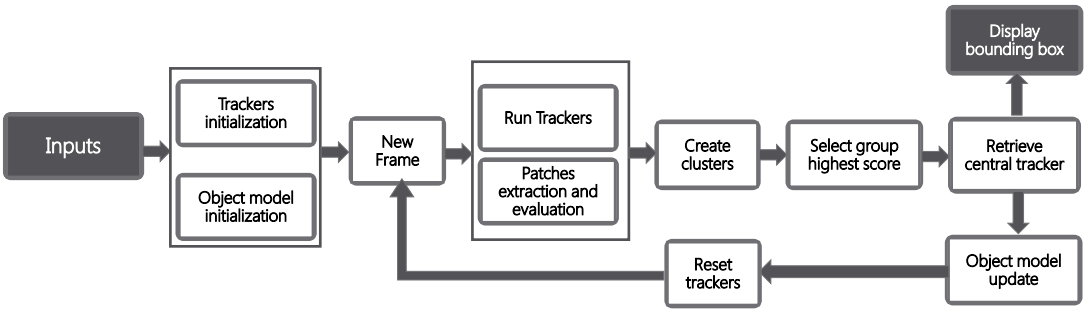

FIGURE 3.1: General schema for generic single-object tracking proposal.

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 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.

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 [**?**] applied this method using a threshold parameter that defines if two result boxes vote for the same position. However, in [**?**], 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 $C = \{c_1, c_2, ..., c_m\}$

3.2.3 Object classification

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

[2] 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.

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

[4] 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.