Siamese Learning Visual Tracking: A Survey

Roman Pflugfelder, *Member, IEEE*

(Draft Article)

Abstract—The aim of this survey is the attempt to review the kind of machine learning and stochastic techniques and the ways existing work currently uses machine learning and stochastic methods for the challenging problem of visual tracking. It is not intended to study the whole tracking literature of the last decades as this seems rather impossible by the incredible vast number of published papers. This first draft version of the article focuses very targeted on recent literature that suggests Siamese networks for the learning of tracking. This approach promise a step forward in terms of robustness, accuracy and computational efficiency. For example, the representative tracker SINT performs currently best on the popular OTB-2013 benchmark with AuC/IoU/prec. 65.5/62.5/84.8 % for the one-pass experiment (OPE). The CVPR'17 work CVNet by the Oxford group shows the approach's large potential of HW/SW co-design with network memory needs around 600 kB and frame-rates of 75 fps and beyond. Before a detailed description of this approach is given, the article recaps the definition of tracking, the current state-of-the-art view on designing algorithms and the state-of-the-art of trackers by summarising insights from existing survey literature. In future, the article will be extended by the review of two alternative approaches, the one using very general recurrent networks such as the Long Shortterm Memory (LSTM) networks and the other most obvious approach of applying sole convolutional networks (CNN), the earliest approach since the idea of deep learning tracking appeared at NIPS'13.

Index Terms—Computer Vision, Object Tracking, Learning Tracking, Siamese Learning.

1 Introduction

Machine learning as data-driven programming approach [1] has steadily improved the robustness and accuracy of vision techniques for the last decades, even more, machine learning has now finally shifted our conceptual view on the design of vision algorithms since the advent of Nvidia's CUDA platform in 2007 and since the availability of large manually labelled image datasets in 2010. State-of-the-art tells us to see vision algorithms as flexible reusable and for now neural learning processes, trained end-to-end on the application-specific data.

The breakthrough in solving categorial object recognition (ILSVRC'10 [2]) has motivated the field to re-apply the connectionism paradigm [3], [4], [5] to hard vision problems with in many cases outstanding success. For example, categorial object recognition has been pushed down on ILSVRC'10-14 from 28.2 % classification error (SVM classifier [6]) to 4.82 % (CNN ensemble [7]) which proved experimentally deep learning as ample solution to the problem. Since connectionism [8] grasped the visual tracking community after the AI winter again in 2013 [9], results have been accordingly shown for the important problem of tracking. For example, Area under the Curve (AuC) performance on OTB-2013 and OPE has been improved by 31 % from 49.9 % (structured SVM [10]) to 56.0 % in 2014 (discriminative correlation filter[1] [12]) to 65.5 % (Siamese network [13]; see Sec. 4.2.2) in 2016 with lots of room to improve.

The remainder of this section introduces the state-of-the-art view on visual tracking and the significant step change of learning the tracking. Sec. 2 overviews other significant surveys covering the tracking literature of the last 15 years and works out as well as summarises in Sec. 3 the mentioned grand challenges of tracking. Sec. 4 focuses on very recent works using Siamese networks and deep offline learning which promise some impact on tracking performance in the coming years, a discussion

• *R. Pflugfelder is with the Centre of Digital Safety and Security, AIT Austrian Institute of Technology and Computer Vision Lab at the Institute of Computer Aided Automation, Vienna University of Technology.*
E-mail: roman.pflugfelder@{ait.ac.at | tuwien.ac.at}

I would be very glad to everyone who supports me by comments and suggestions to substantially improve this working paper.

1. DSST is not a connectionist approach and was VOT'14 winner [11].

which concludes the paper in Sec. 5.

1.1 Tracking Definition

Tracking [14] and in particular visual tracking [15], [16] is basically a sequential inference problem. As a result, tracking makes predictions of latent variables[2] by the way of collecting sufficient evidence with visual sensory inferred from spatiotemporal stimuli in the scene. In most applications, the observations are made by RGB, infrared, depth i.e. LIDAR, ToF cameras, radar or some combination, but other cameras such as event cameras [17], hyperspectral cameras and fluorescence microscopy in microbiology [18] are becoming increasingly popular. Physical objects or phenomena in the scene cause usually these stimuli, for example natural objects such as humans, animals, cells or artificial objects like cars but also complex phenomena as the formation and movement of clouds [19].

The latent variables originate from the mathematical description of scene, objects, camera and their relationships and the assumptions made by the designer. A description is a set of data structures, related by computational feature transforms that give the latent variables the necessary meaning to make inference useful. The image template obtained from the original image is a good example for a well-known simplest description with the four-parameter axis-aligned bounding box as latent variable. In many cases, simple descriptions of appearance, shape, location and motion are sufficient, although, some situations need more complex descriptions such as manoeuvring objects, tracking in clutter [14] and tracking multiple objects [16]. The latter, for example, needs mutual descriptions in order to infer identity.

1.2 Tracker Design

The paradigm shift from pure hand-crafted designs to connectionism in computer vision suggests to design trackers as multiple complementary and loosely coupled learning processes which are finally embedded and intertwined into some kind of management and control process [20], [21]. There is evidence in neuroscience and cognitive science that the visual brain and mind works in similar way [22], [23]. The motivation comes from the outstanding success of convolutional networks as solution to categorial object recognition and by the higher complexity of tracking compared to categorial object recognition.

2. Other names are (hidden) state variables or variables.

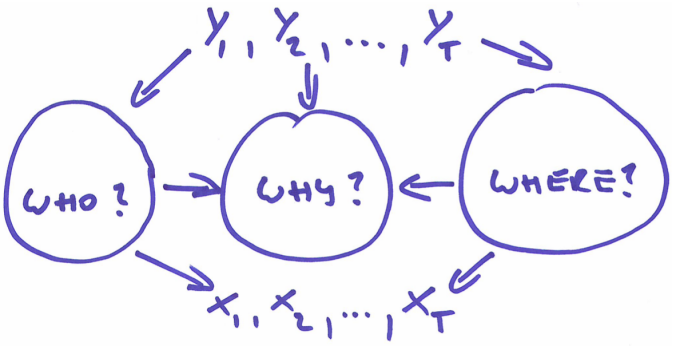

Fig. 1. Data flow diagram of a tracker in three pathways: who, where and why. These interconnected pathways, constituted by the descriptions of scene, objects, camera and their relationships, process incoming observation data $Y_1, \ldots Y_T$ and predict finally the latent variables $X_1, \ldots, X_T$.

Each learning process is a hierarchical meaningful description of the visual input, with the aim to yield evidence of the tracker's state and to finally make predictions of the particular latent variables given by the design. The tracker under this paradigm works basically in two operating states either it is learning from incoming visual input to adapt the parameters of the learning processes by end-to-end training or the tracker is inferring the latent variables of the tracker.

The data flow of such a tracker's design shows three fundamental pathways [24] which Fig. 1 illustrates:

- **Who?** Collects evidence of objects' appearance and shape as well as their identities, similar to the ventral pathway in the human brain. Modules of this pathway are in literature known as object representation, appearance modelling, feature selection and data association.
- **Where?** Collects evidence of objects' location and motion (derivatives of location with respect to time) similar to the dorsal pathway in the human brain. Modules of this pathway are in literature known as object initialisation, object localisation and motion estimation.
- **Why?** Formalises the objectives of the calculus.

Trackers are inference machines, gathering evidence of who and where by collecting sequentially observations. Parametric descriptions are precisely formulated and initially given. Each prediction step completes the inference process by computing the latent variables given the evidence of who and where. The third why pathway formalises the ob-

jective function so that the tracker can come to a conclusion. For example, a popular objective is least-squares to given ground truth, e.g. a given image template of the object.

Different interconnected modules capturing the various descriptions of the tracking problem constitute these pathways. The complexity of interconnection and the choice of the mathematical formulation has direct consequence to the robustness and accuracy of the tracking results as well as to the computational efficiency of the tracking algorithm. Popular designs are mode seeking, HMM's, Kalman filter, variants and generalised SMC methods.

Recently [25] proposed five key building blocks of a tracker: motion model, feature extractor, observation model, model updater, ensemble post-processor.

1.3 Learning Tracking

This network of pathways has a longtime been seen through the glass of dynamical systems theory [14], [26]. Motivated by radar technology, a huge body of statistical estimators has been studied since the 1950s. Although learning has been considered in the form of system identification, i.e. the algorithms are able to identify and adapt the system parameters and may switch the underlying statistical model of the data, machine learning has not widely been recognised as tool to estimate the functional parts of the system itself whilst or before receiving the data. Radar tracking mostly aims for efficient statistical algorithms covering the identification of objects and the where and why pathways.

This is opposite to visual tracking that mainly focuses on visual appearance - the who pathway. As images are complex data encoding a plethora of information, handcrafted algorithms are very constrained in their reliability, hence researchers started early in the 90s to apply machine learning to functions of feature selection with a huge body of literature since then.

Learning rigorously the functional parts such as the camera model, the feature transform, the motion behaviour, the object identities individually as well as holistically is still in its infancy. There are results in human vision research, that learning tracking is an intentional, preconscious process [27], so there is a good chance for learning tracking from large weakly labeled data. There are also hints that humans simultaneous tracking is possible for a very low number of three to five objects, an incentive for artificial vision to beat human capabilities.

2 Related Work

Five significant surveys of tracking have been published recently [28], [29], [30], [31], [32]. Yilmaz et al. [28] proposed a first taxonomy of tracking and proposed important challenges for trackers such as unconstrained video, context, efficient inference and application specific fine-tuning online either discriminative or un-/semisupervised by using machine learning.

Cannons [29] gives a comprehensive overview of building blocks such as representation, initialisation, prediction, association and adaptation. The paper discusses pros and cons of features such as points, edges, contours, regions and their combinations. The work identifies as important challenges for future research feature selection i.e. the representation, the evolution of the representation over longer time periods, the combination of trackers and tracker evaluation.

Yang et al. [30] review global and local features, the integration of object context and online generative versus discriminative learning. The work identifies ensemble learning to combine heterogeneous features into a coherent inference framework where particle filters are compared in detail. The work states the drift problem as important problem and proposes as future research consideration of adaptive priors, object context, pixel-wise segmentation, both generative and discriminative learning and SMC methods for inference.

Li et al. [31] survey solely representation with a summary of global and local features and a review on generative, discriminative and hybrid learning approaches. The survey concludes with the inevitable trade-off between robustness and accuracy by carefully balancing representation and prior information. The work proposes the need for a principled approach to reconstruct 3-D pose, the need for viewpoint and frame-rate invariance and the need for attentive mechanisms for initialisation similar to human vision.

Finally, Smeulders et al. [32] compare 19 online trackers by their predicted bounding boxes on the ALOV benchmark. The paper concludes with the importance of the experimental setup which is crucial to assess performance and that tracking unknown objects in unknown scenarios is an open problem. The experiments show little conceptual consensus among the trackers and find none of the trackers superior. There is evidence that complex representations are inferior to simpler ones. Sparse

and local features seem appropriate to handle rapid changes such as occlusion, clutter and illumination.

3 Grand Challenges

We summarise from this related work the following remaining challenges to solve tracking:

3.1 Complexity

A description of tracking limits the number of predictable latent variables. The higher the complexity, i.e. the information capacity of the description, the more information can be inferred by a set of latent variables. For example, the image template predicts either bounding boxes or pixelwise segmentations of the object but not 3-D pose which needs descriptions of depth. Higher complexity, however, comes with difficulty of collecting unambiguous evidence about the predictions of latent variables. Descriptions should be made simple to be informative but as complex to meet application specific demands.

3.2 Uncertainty

Trackers conform to a specification of quality parameters during a period of time such as the robustness needed for tolerating changes and for preventing failure. The description of tracking needs to be invariant to temporal changes of the scene, objects, camera and their relationships. For example, changes in object appearance need to be tolerated by the tracker. At this point, we have to distinguish vagueness and uncertainty. While vagueness refers to controllable risks of specified changes, uncertainty refers to risks of possible changes, i.e. a potential subset of changes is unspecified hence unknown to the tracker. An example is a tracker for arbitrary objects. Such trackers become important. Descriptions of arbitrary objects are therefore needed. Such descriptions should be sufficiently made time-invariant to meet application specific demands.

3.3 Initialisation

Application users or object detectors initialise trackers. Initialisation takes place in the first video frame and after full occlusions during tracking. For initialisation in the first video frame, the standard approach assumes either the human in the loop or the limitation to certain object categories. Online learning a detector of the individual object handles usually the problem of full occlusions during tracking. In case of uncertainty, a generalised initialisation is needed which is able to handle arbitrary objects. Such an attentive tracker detects and tracks salient objects occurring in the video. For example, a salient object might be an image region of atypical spatiotemporal characteristics compared to its neighbourhood. Attentive mechanisms based on saliency are also found in human vision. Trackers for arbitrary objects should be made attentive to fully exploit their potential.

3.4 Computability

Inference has to be as efficient and accurate as possible, being able to compute predictions of latent variables and satisfying quality parameters at the same time. Complexity of the description and the prediction of the latent variables go hand in hand, turning inference very quickly to computational intractability. A principled approach balances both, leading to an optimal graceful degradation of inference.

3.5 Comparison

Trackers need to be objectively compared by accepted experimental methodologies. This would foster scientific progress in the field.

3.6 Disucssion

The literature shows the enormous success of using machine learning to improve the robustness of tracking. While recent work and initiatives try to establish community platforms, evaluation protocols and allow new insights into tracking, only a few works consider the problem of initialisation. Vagueness, complexity and computability of tracking are strongly intertwined and suggest a common machine learning approach as principled solution. However, it is important to point out that although machine learning is very promising to control vagueness by fully exploiting video information, learning will fail in the most general case of uncertainty, as learning assume priors of the underlying random processes, a constraining assumption in case of real disjuncture between known random processes and new unknown processes with unknown statistics. [33] emphasised recently this an other problems as robustness to distributional shift.

4 Siamese Tracking

Learning with a Siamese network [34], [35] is a promising approach to tackle some of these difficult tracking challenges. A Siamese network is a Y-shaped neural network that joins two network branches in final layers to produce a single output. The idea originated 1993 in fingerprint recognition [34] and signature verification [35], where the task is to compare two imaged fingerprints or two hand-written signatures and infer identity. A Siamese network captures the comparison of the preprocessed input as a function[3] of similarity with the advantageous ability to learn similarity and the features jointly and directly from the data [4]. Despite their generality and usefulness in various applications, relatively less is known about statistical foundation and properties [37], [38]. Siamese networks have also been applied to face verification and recognition [39], [40], [41], [42], areal-to-ground image matching [43], stereo matching [44], patch matching [45], [46], [47], optical flow [48], large-scale video classification [49] and one-shot character recognition [50].

4.1 Overview

Motivated by these successful applications, some research groups studied very recently the Siamese networks for tracking [13], [51], [52], [53], [54]. These proposed methods consider similarity as a priori given except for [51], [52] who learn similarity and features jointly. Joint learning utilises the Y-shaped network architecture to its full extent, while assumptions, such as a given similarity, restrict parts of the network. Joint learning is currently little understood, while feature learning for the aim of compression has been extensively studied over decades by the signal processing community [38]. Learning similarity with given features as last case is rigorously studied in statistical decision theory and machine learning.

All methods assume an initial given bounding box in the first frame and the presence of a search region in the next frame where the object template is matched. Object template and search region are input to the network except for [13].

The attempt of all proposed methods is to learn a hierarchy of convolutional features of arbitrary training objects by ignoring categorisation and to train entirely offline, end-to-end the network by using back-propagation and infer at runtime the object's bounding box either by regressing directly [52] or by estimating its centre position and scale in two subsequent steps [51], [53], [54] or by ranking proposed bounding box candidates given certain criteria to retrieve the best match [13]. [52] showed that this approach learns very generic features which generalise to new objects and even new object categories not present in the training data. Siamese networks on one hand combine the expressive power of convolutional networks in the branches with real-time inference which is indispensable from an application point of view. On the other hand, the approach allows due to its simplicity a better understanding of the implications of learning jointly features and similarity.

4.2 Proposed Methods

This section summarises the proposed methods concerning the technical details of network, training and inference and then compares and discusses the important differences to gain some insight into the approach of Siamese tracking.

4.2.1 YCNN

[51] propose as possibly first but unpublished attempt two identical branches similar to VGGNet [55] with three conv and max-pooling layers, both linked to three fc layers. The conv layers share the same parameters. Each layer finishes with a ReLU except for the output which finishes with a sigmoid function. The network output is a 0-1 bounded prediction map with high values at pixels indicating object presence. The branches work as feature hierarchies aggregating fine-to-coarse spatial details, while the fc layers design spatiotemporal features as well as a general similarity function. Thus, YCNN learns discriminating features of growing complexity while simultaneously learning similarity between template and search region with corresponding prediction maps. Training is done in two stages on augmented images of objects from ImageNet and for fine-tuning with videos from VOT-15 [56] and TB-100 [57]. Training minimises a weighted L^2 loss by using Adam [58], mini-batches and dropout. Weighting is important as nearly 95 % of pixels in the prediction map have very low to zero values. During tracking, the feed forward pass infers then position as maximum in the prediction map. By averaging the prediction map over the five

3. A function of the class of Lipschitz functions $f : [-1, 1]^d \to [-1, 1]$ [1].

4. Similarity is understood as decision function and features are known as experimental design in statistical decision theory [36].

most confident maintained templates avoids drift. Repeating inference with scaled templates estimates additionally overall scale.

4.2.2 SINT

[13] propose two identical query and search networks inherited from AlexNet [59] and VGGNet with five conv and two max-pooling layers, three region pooling layers, an fc layer and a final L^2 normalisation layer. ReLUs follow each conv layer. Max-pooling is done after the first two conv layers. Both networks are unconnected but share the same weights. Instead of object and search region templates, the whole two subsequent frames are input, hence bounding boxes locating the object in the query frame and bounding boxes locating candidates in the search frame are additionally fed to the networks. The networks' outputs are normalised features lying on the same manifold. Again, the networks work as feature hierarchies aggregating fine-to-coarse spatial details, however in this work similarity is a priori defined by the training loss function. So SINT learns discriminating solely features of growing complexity with bounding boxes in query and search frame and an additional binary variable indicating correct and incorrect pairs measured by the Jaccard index. Training is done on images of objects from ALOV [32]. Training minimises a margin contrastive loss and uses pre-training on ImageNet. During tracking, the query is fed with the initial bounding box in the first frame resulting in a query feature vector. Inference samples candidates at radial positions and different scales and feeds the search at once resulting in feature vectors for each candidate. An offline learned ridge regressor refines finally position and scale of the winning candidate with maximal inner product to the query.

4.2.3 SiamFC

[53] propose two identical branches inherited from AlexNet with five conv layers, max-pooling following the first two conv layers and ReLUs after every conv layer except for conv5. A novel cross-correlation layer links the two conv5 layers. By waiving padding the whole network is fully-convolutional. The output is an unbounded correlation map with high values at pixels indicating object presence. As for YCNN and SINT, the branches can be seen as spatial description of increasing complexity which is embedded in a metric space where cross-correlation is used as similarity function. Like SINT, SiamFC learns discriminating solely the features with triplets of template, search region and corresponding prediction map. Values isotropically within a radius of the centre count correctly to the object's position, hence are labeled positively whereas all other values are labeled negatively. Training is done on videos of objects from ImageNet [2]. Augmentation considers scale but not translation, because of the fully-convolutional network property. Training minimises a discriminative mean logistic loss by using SGD, mini-batches, Xavier initialisation and annealing of the learning rate. Tracking computes the position via the up-sampled prediction map for a given template. The tracker handles scale by searching over five different scale variations and updates scale by linear interpolation.

4.2.4 CFNet

[54] adds a correlation filter and crop layers to the branch that concerns the template. These layers follow directly the convolutional network. The input is a larger region of the frame including the template, hence resolution of feature maps in the branches and prediction map is larger. Feature maps are further multiplied by a cosine window and cropped after correlation to remove the effect of circular boundaries. CFNet inherits the basic ability from SiamFC to discriminate spatial features with triplets of template region, search region and corresponding prediction map. Instead of unconstrained features, CFNet learns features that especially discriminate and solve the underlying ridge regression of the correlation layer by exploiting background samples in the surrounding region of the template. The learnt parameters of the correlation layer remain fixed during tracking, no online learning happens as shown by [60]. Training is done as with SiamFC by using the same algorithms on videos of objects from ImageNet. To make training end-to-end, emphasis has been on a differential correlation layer and on back-propagation of the parameters. Correlation is formulated in the Fourier space to preserve efficiency of computation. Tracking is similar simple as in SiamFC and computes position and scale by a single feed forward pass. The prediction map is multiplied by a spatial cosine window to penalise larger displacements. Instead of handling five different scale variations, scale is handled as by [61]. To fully exploit the correlation filter, the initial template is updated in each frame by a moving average.

4.2.5 GOTURN

[52] proposes two convolutional branches inherited from AlexNet up to pool5. Both branches share the same parameters. These pool5 features of both branches are connected to a single vector and fed to three fc layers. ReLUs are used after each fc layer. The final fc layer links to an output layer with four nodes describing the bounding box. The output is scaled by a validated constant factor. GOTURN learns simultaneously the hierarchy of spatial features in the branches as well as spatiotemporal features and the similarity function in the fc layers to discriminate between template and search region with corresponding bounding boxes. Training is done in two stages on augmented images of objects from ImageNet and on videos from ALOV by using standard back-propagation of CaffeNet. Augmentation assumes linear translation and constant scale with parameters sampled from a Laplace distribution, hence small motion is assumed to occurs more frequently than larger displacements. Training minimises a L^1 loss between predicted and ground truth bounding box by using mini-batches, dropout and pre-training of the branches on ImageNet without fine-tuning these parameters to prevent overfitting. Tracking initialises the template in the first frame and updates the template with the predicted bounding box for each frame. Crops of the current and next frame yield template and search region. These crops are not exact but padded to add context.

4.3 Discussion

After understanding the methods' details, this section compares the details concerning differences in branches, outputs and connections of branches. Differences in the networks' training and inference are finally discussed.

4.3.1 Network Branches

All proposed methods suggest convolutional branches inherited either from AlexNet or VGGNet with five conv layers except for YCNN that complements three layers by two fc layers and CFNet that studies one to five layers. The inheritance from AlexNet and VGGNet allows transfer learning from ImageNet and ALOV. The methods consider equal branches by effectively sharing the parameters which avoids during the fine-tuning overfitting to the small datasets currently available [13]. The first two conv layers capture very local visual detail, for example edges, contributing to the accuracy of the tracker, while conv layers three to five aggregate this detail to an object specific description, for example category specific details, which is important for the robustness of the tracker [60]. Max-pooling as it is part of AlexNet and VGGNet introduces invariance to deformations of the object but it also reduces significantly image resolution and by that hinders improvements to the tracker's accuracy. All authors except [51] recognise this limitation and use two max-pooling layers to trade-off accuracy and deformation invariance. [54] also show important insights into the number of layers. They report saturation of tracking performance with increasing network depth and that more than five conv layers yield minor performance gains. CFNet implements the object specific description on higher layers with a correlation filter which allows an effective object specific description and fast computation in the Fourier domain. This significantly improves computational performance and shows that the branches are representable in various ways by combining layers of heterogeneous features.

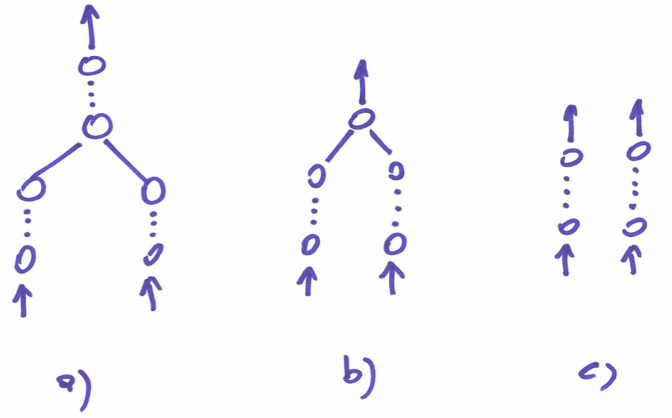

Fig. 2. The proposed methods use network architectures of varying complexity: a) convolutional branches and fc connection layers (YCNN, GOTURN), b) convolutional branches and a single connection layer (SiamFC, CFNet), c) convolutional branches with equal final normalisation layers (SINT).

4.3.2 Connection of Branches

All proposed methods except SINT connect the branches, SiamFC and CFNet with a single cross-correlation layer, YCNN and GOTURN with three fc layers. SINT omits the concatenating layer by using normalisation layers at the end of both branches. Fig. 2 illustrates these three variations of network architecture. This Siamese network architecture of SiamFC, CFNet and SINT in combination with parameter sharing limit the feature selection to the spatial image domain. Instead, YCNN and GOTURN enable additional learning of spatiotemporal

features in the concatenating layers, as argued by [52] such as "relationships between an object's appearance change and its motion" which seems very general for different categories of objects. Parameter sharing has the consequence that all methods require appearance constancy between template and search region, hence [51]'s argument that YCNN's deep features show "superiority of recognising an object with varying appearance" is questionable. SiamFC,

Theoretically, the network of GOTURN generalises over the network of SiamFC which allows capturing features beyond sole visual features of the exemplar image and which allows regression of the bounding box instead of convoluting a final score map capturing potential positions of the exemplar image within the search image. The author's argue that GOTURN learns a generic relationship between arbitrary motion and visual features, however this is not clear yet. Due to the more general Y-shaped architecture it might learn features beyond pure visual such as motion and their relationship in the fully-connected layers, however the network might also be able to learn context features as well.

CFNet and SINT assume a specific function of similarity and the idea is to solely learn visual features to best match the given similarity. SINT even expresses similarity by the training loss which might have advantages in generalisation as particular different functions of similarity and training loss might derive adversary optimisation problems. This is not a problem for YCNN and GOTURN, as similarity and features are jointly learned, however, the interference with the particular training loss is unclear.

4.3.3 Network Training

Training is a crucial for sufficient performance. All methods describe basically two training phases, (i) a pre-training phase to transfer-learn generic features of objects from labeled datasets and (ii) a fine-tuning phase to adapt features to given video sets. The cross-correlation layer has here advantages as cross-correlation preserves the convolutional property of the whole network which introduces invariance to object translation. Therefore training samples must not contain translated versions which reduces significantly the effort for training. Less augmentation of training data is needed. The training loss and its choice has significant influence on the training result. [52] argue that L^1 is superior to L^2 as it penalises more harshly small errors near zero which increases substantially accuracy of the predicted bounding box. This argument is an exception, as none of the other studies show some insights into this important problem. [52] chose also different inputs for training and studied their influence on the mean error derived from VOT accuracy and robustness measures. They show that feeding the network with whole frames instead of template and search region pairs, the frames' contexts are exploited which reduces significantly the mean error, especially in cases of occlusion. SINT is the only method following this insight but without any hints of their awareness. The reason is that their motivation comes from image retrieval where processing of frames is common.

GOTURN allows the use of still images.

4.3.4 Tracker Inference

All proposed networks return in a single feed forward pass information about the bounding box in the search region. YCNN, SiamFC and CFNet return position and in a post-processing step then scale. SINT needs prior sampling of candidates in the search region and returns similarities to the template all at once thanks to the region proposal layer. The best candidate with maximum similarity defines the bounding box in the new frame. GOTURN is different as it regresses directly the bounding box. The idea of inferring direct position seems elegant and superior to the idea of candidate sampling as it allows e.g. in case of SiamFC and CFNet dense cross-correlation at pixel level. GOTURN regresses directly the bounding box and is the simplest recurrent network as it can be seen as unrolled recurrent network with Markov property. Although [53] recognises this relationship, none of the works [13], [51], [53], [54] follow this fully sequential approach, as the intention is to learn a matching function. A significant advantage of YCNN, SINT, SiamFC and CFNet over GOTURN is that the predictions allows solutions for tracking in clutter, SINT however limited by the candidate samples. SiamFC and SINT consider a single initial template, while YCNN maintains the k-best templates, GOTURN keeps the last and CFNet a moving average template. [52] show that updating the template improves accuracy and robustness of the tracker.

4.3.5 Tracker Results

CFNet seems a promising method as it has state-of-the-art performance with 75 fps[5] with less than

5. GOTURN runs best at 100 fps [52].

4 % of the parameters of other five layer methods (in total 600 kB) such as SiamFC. This makes CFNet applicable to embedded applications. CFNet, SiamFC and SINT show comparable performance by reaching IoU/prec. 60/80 % on OTB-2013 and one pass evaluation (OPE)[6]. [51] reports significant lower IoU/prec. of 60/70 %. [52] did not report results on OTB-2013.

5 Conclusion

The various combinations of possible inputs, outputs and features and their implementation as layers in the network need definitely future research work. There are strong pros for a fixed similarity function, nevertheless learning similarity with fixed features or learning similarity and features jointly might conceal success as shown in the fields of re-identification [62] and sensor networks [63]. All methods keep for a good reason the tracking framework simple, namely to be able to better study the network's properties. There is much room for improvement concerning the tracking, for example by combining the network with filtering methods. Seeing the Siamese network as matching function or seeing the network as simplest recurrent network poses important questions about the integration of the network into the tracking framework which have not been answered yet. More training data is needed as well as new ideas for combining supervised and unsupervised training approaches as labelled data will always be limited. There is currently little knowledge about the influence of training loss on the overall performance. Insights into these topics by in-depth ablative analysis such as done by [52], [54] are further needed.

There are currently three lines of research: There is tracking research that assumes an initial label of the unknown object e.g. a bounding box and investigates tracking in the subsequent frames. These methods are combined with detection which on the one hand allows integrated perception but on the other hand the use of detectors unnecessarily restricts the tracker to certain object categories. The third line of research studies tracking jointly with attentional mechanism that does not assume any knowledge of the object.

Acknowledgments

I thank all reviewers for reading the article and for their valuable comments which improved substantially the work. This research has received funding from the EU ARTEMIS Joint Undertaking under grant agreements no. 621429 (EMš) and from the FFG (Austrian Research Promotion Agency) on behalf of BMVIT, The Federal Ministry of Transport, Innovation and Technology. This work was supported by the AIT strategic research programme 2017 *Visual Surveillance and Insight*.

6. SINT performs best with IoU/prec. 62.5/84.8 % on OTB-2013 and OPE [13].

References

[1] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[3] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[4] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, pp. 65–386, 1958.

[5] G. Hinton and J. Anderson, Eds., *Parallel Models of Associative Memory*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Publishers, 1981.

[6] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: Fast feature extraction and svm training," in *CVPR 2011*, June 2011, pp. 1689–1696.

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, D. Blei and F. Bach, Eds. JMLR Workshop and Conference Proceedings, 2015, pp. 448–456.

[8] S. J. Nowlan and J. C. Platt, "A convolutional neural network hand tracker," in *Proceedings of the 7th International Conference on Neural Information Processing Systems*, ser. NIPS'94. Cambridge, MA, USA: MIT Press, 1994, pp. 901–903.

[9] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 809–817.

[10] S. Hare, A. Saffari, and P. H. S. Torr, "Struck: Structured output tracking with kernels," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 263–270.

[11] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. Nebehay, T. Vojíř, G. Fernández, A. Lukežič, A. Dimitriev, A. Petrosino, A. Saffari, B. Li, B. Han, C. Heng, C. Garcia, D. Pangeršič, G. Häger, F. S. Khan, F. Oven, H. Possegger, H. Bischof, H. Nam, J. Zhu, J. Li, J. Y. Choi, J.-W. Choi, J. F. Henriques, J. van de Weijer, J. Batista, K. Lebeda, K. Öfjäll, K. M. Yi, L. Qin, L. Wen, M. E. Maresca, M. Danelljan, M. Felsberg, M.-M. Cheng, P. Torr, Q. Huang, R. Bowden, S. Hare, S. Y. Lim, S. Hong, S. Liao, S. Hadfield, S. Z. Li, S. Duffner, S. Golodetz, T. Mauthner, V. Vineet, W. Lin, Y. Li, Y. Qi, Z. Lei, and Z. H. Niu, *The Visual Object Tracking VOT2014 Challenge*

Results. Cham: Springer International Publishing, 2015, pp. 191–217.

[12] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference*, M. Valstar, A. French, and T. Pridmore, Eds. BMVA Press, 2014.

[13] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 1420–1429.

[14] S. Challa, M. R. Morelande, D. Musicki, and R. J. Evans, *Fundamentals of Object Tracking*. Cambridge University Press, 2011.

[15] E. Maggio and A. Cavallaro, *Video Tracking: Theory and Practice*. Wiley, 2011.

[16] M. Betke and Z. Wu, "Data association for multi-object visual tracking," *Synthesis Lectures on Computer Vision*, vol. 6, no. 2, pp. 1–120, 2017/06/30 2016.

[17] D. Scaramuzza, G. Gallego, and A. Censi, Eds., *IEEE International Conference on Robotics and Automation (ICRA): First International Workshop on Event-based Vision*. Singapore: IEEE, June 2017.

[18] S. Skylaki, O. Hilsenbeck, and T. Schroeder, "Challenges in long-term imaging and quantification of single-cell dynamics," *Nat Biotech*, vol. 34, no. 11, pp. 1137–1144, 11 2016.

[19] C. W. Chow, S. Belongie, and J. Kleissl, "Cloud motion and stability estimation for intra-hour solar forecasting," *Solar Energy*, vol. 115, pp. 645 – 655, 2015.

[20] P. Winston, "Learning structural descriptions from examples," Ph.D. dissertation, MIT, January 1970.

[21] G. Sussman, "A computational model of skill acquistion," Ph.D. dissertation, MIT, August 1973.

[22] S. Grossberg, "The complementary brain: unifying brain dynamics and modularity," *Trends in Cognitive Sciences*, vol. 4, no. 6, pp. 233–246, 2017/06/30 2000.

[23] J. H. S. Kandel, Eric R and T. M. Jessell, *Principles of Neural Science*, ser. Health Professions Division. New York: McGraw-Hill, 2000.

[24] L. Bazzani, N. Freitas, H. Larochelle, V. Murino, and J.-A. Ting, "Learning attentional policies for tracking and recognition in video with deep networks," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ser. ICML '11, L. Getoor and T. Scheffer, Eds. New York, NY, USA: ACM, June 2011, pp. 937–944.

[25] N. Wang, J. Shi, D. Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3101–3109.

[26] T. K. Yaakov Bar-Shalom, X. Rong Li, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Wiley, March 2004.

[27] P. Jarvis, *Towards a Comprehensive Theory of Human Learning*. Routledge, 2012.

[28] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006.

[29] K. Cannons, "A review of visual tracking," York University, Department of Computer Science and Engineering and the Centre for Vision Research, 4700 Keele Street Toronto, Ontario M3J 1P3 Canada, Tech. Rep. CSE-2008-07, September 2008.

[30] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomput.*, vol. 74, no. 18, pp. 3823–3831, Nov. 2011.

[31] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 4, pp. 58:1–58:48, Oct. 2013.

[32] A. W. M. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual tracking: An experimental survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1442–1468, July 2014.

[33] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," *CoRR*, vol. abs/1606.06565, 2016.

[34] P. Baldi and Y. Chauvin, "Neural networks for fingerprint recognition," *Neural Computation*, vol. 5, no. 3, pp. 402–418, May 1993.

[35] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a siamese time delay neural network," in *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, 1993, pp. 737–744.

[36] J. O. Berger and J. O. Berger, *Statistical decision theory and Bayesian analysis*. New York: Springer-Verlag, 1985.

[37] D. Blackwell, "Comparison of experiments," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, Calif.: University of California Press, 1951, pp. 93–102.

[38] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "On surrogate loss functions and f-divergences," *The Annals of Statistics*, vol. 37, no. 2, pp. 876–904, 2009.

[39] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005, pp. 539–546 vol. 1.

[40] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1701–1708.

[41] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[42] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 815–823.

[43] T. Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocalization," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 5007–5015.

[44] J. Žbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1592–1599.

[45] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3279–3286.

[46] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 118–126.

[47] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 4353–4361.

[48] A. Dosovitskiy, P. Fischery, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 2758–2766.

[49] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1725–1732.

[50] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proceedings of the 32nd International Conference on Machine Learning Deep Learning Workshop*, vol. 37, Lille, France, 2015.

[51] K. Chen and W. Tao, "Once for all: a two-flow convolutional neural network for visual tracking," *CoRR*, vol. abs/1604.07507, 2016.

[52] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 749–765.

[53] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *Computer Vision – ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II*, G. Hua and H. Jégou, Eds. Springer International Publishing, 2016, pp. 850–865.

[54] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for Correlation Filter based tracking," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2017.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, vol. abs/1409.1556, 2014.

[56] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, R. Pflugfelder, A. Gupta, A. Bibi, A. Lukezic, A. Garcia-Martin, A. Saffari, A. Petrosino, and A. S. Montero, "The visual object tracking vot2015 challenge results," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 564–586.

[57] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, Sept 2015.

[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[59] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[60] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 621–629.

[61] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[62] S. Liao, Y. Hu, X. Zhu, and S. Z. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 2197–2206.

[63] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Decentralized detection and classification using kernel methods," in *Proceedings of the Twenty-first International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 80–.

Roman Pflugfelder is Scientist at the AIT Austrian Institute of Technology and lecturer at TU Wien. He received in 2002 a MSc degree in informatics at TU Wien and in 2008 a PhD in telematics at the TU Graz, Austria. In 2001, he was academic visitor at the Queensland University of Technology, Australia. His research focuses on visual motion analysis, tracking and recognition applied to automated video surveillance. He aims to combine sciences and theories in novel ways to gain theoretical insights into learning and inference in complex dynamical systems and to develop practical algorithms and computing systems. Roman contributed with more than 55 papers and patents to research fields such as camera calibration, object detection, object tracking, event recognition where he received awareness of media as well as several awards and grants for his scientific achievements. Roman is senior project leader at AIT where he has been managing cooperations among universities, companies and governmental institutions. Roman co-organised the Visual Object Tracking Challenges VOT'13-14 and VOT'16-17 and was program co-chair of AVSS'15. Currently he is steering committee member of AVSS. He is regular reviewer for major computer vision conferences and journals. For more details see https://www.caa.tuwien.ac.at/cvl/staff/roman-pflugfelder.