# ALExTRAC: Affinity Learning by Exploring Temporal Reinforcement within Association Chains

Alex Bewley[1], Lionel Ott[2], Fabio Ramos[2] and Ben Upcroft[1]

***Abstract*— This paper presents a self-supervised approach for learning to associate object detections in a video sequence as often required in tracking-by-detection systems. In this paper we focus on learning an affinity model to estimate the data association cost, which can adapt to different situations by exploiting the sequential nature of video data. We also propose a framework for gathering additional training samples at test time with high variation in visual appearance, naturally inherent in large temporal windows. Reinforcing the model with these difficult samples greatly improves the affinity model compared to standard similarity measures such as cosine similarity. We experimentally demonstrate the efficacy of the resulting affinity model on several multiple object tracking (MOT) benchmark sequences. Using the affinity model alone places this approach in the top 25 state-of-the-art trackers with an average rank of 21.3 across 11 test sequences and an overall multiple object tracking accuracy (MOTA) of 17%. This is considerable as our simple approach only uses the appearance of the detected regions in contrast to other techniques with global optimisation or complex motion models.**

## I. Introduction

This paper presents the design and implementation of a self-supervised framework to solve the data association component for tracking-by-detection. In tracking-by-detection, a low level object detector typically operates independently of the high level data association. This independence offers several benefits including: robustness to drift as it does not rely on state information, accommodates a changing number of objects in the scene, and implicit recovery from detection failure. To perform the data association most approaches rely on position information and incorporate motion models [1], [2] where hand crafted appearance features, such as colour histograms are only used to resolve ambiguous situations [3]–[5].

While the tracking-by-detection problem has been investigated with various formulations, little attention has been invested into improving the appearance similarity measure, commonly used for quantifying the data association cost. Our approach addresses this gap by actively modelling the pairwise similarity between detections with a probabilistic classifier, referred to as the *affinity model*. The output of this model can be inserted into any tracking-by-detection framework to improve data association in any arbitrary environment.

We demonstrate that the temporal structure of a video sequence can be explored to gather training samples to reinforce the proposed affinity model. For example, Fig. 1 illustrates an *association chain* where the frame-to-frame affinities are appropriately high and can be used to identify a difficult matching pair with a large temporal separation. To gather reliable matching and non-matching samples, we rely on two basic constraints for exploiting the structure of video data. Firstly, an object should have higher visual affinity to itself than other objects over short temporal durations, and secondly, co-existing and non overlapping detections cannot represent the same object, known as mutual exclusion. These constraints provide a valuable utility in governing the self-supervision, particularly in preventing drift, to facilitate life long learning.

Fig. 1. An illustration of an association chain showing that while the frame-to-frame affinities are strong, the direct affinity over longer temporal windows are affected by viewpoint change and background clutter. Warmer colours denote stronger affinity.

Our design is simple and practical as it can learn at test time without labels, automatically adapting to new visual appearances. Additionally, fewer parameters are required, in contrast to conventional motion model approaches. Finally, this purely appearance based affinity model is complimentary to other approaches relying on object dynamics such as [6] or approaches that incorporate an appearance based cost in a global optimisation [7]. The key contributions of this paper are as follows:

- modelling visual affinity with a probabilistic classifier,
- gathering matching examples with high variance by exploring association chains,
- use of co-existing detections as a source of non-matching examples,
- use of a purely appearance based data association cost,
- affinity model reinforcement without explicit labelling.

[1]A. Bewley and B. Upcroft are with the Queensland University of Technology (QUT) alex.bewley@hdr.qut.edu.au

[2]L. Ott and F. Ramos are with the School of Information Technologies, The University of Sydney, Australia

This paper is organised as follows: In the next section, we position the proposed approach among existing works. An overview of the proposed method is given in section III. In section IV, we demonstrate the performance of the proposed method before a conclusion and an outlook to future work is given in section V.

## II. Related Work

Here, we review techniques incorporating visual appearance for multiple object tracking (MOT) with a strong focus on techniques that include a learning component or rely on affinity measures. In a general sense, many techniques associate detections between frames by utilising features extracted from each frame to measure and compare the strength of all potential matches [1], [2], [8], [9]. Selecting among the candidate matches can either be performed in a greedy (winner-take-all) approach [8], [9] or by formulating a bipartite graph [1], [2] which can be solved optimally [10]. These approaches all require a measure of affinity between two candidate detections but have mostly only considered fixed affinity measures such as the intersection kernel [7], [11] or Kullback-Leibler distance [5] between two colour histograms or normalised cross correlation between patches [2].

The focus of this work is to learn an affinity measure which is optimised for the task of visual tracking, and which can complement the above matching and graph optimisation approaches. This approach is also related to the structural SVM approach, originally used for supervised clustering [12]. Kim et al. [13] use implicit spatial-temporal structure of video based tracking by employing a structural SVM to predict the optimal matching as formulated in a bipartite graph structure.

Another approach is to cast data association into a ranking problem where similar objects should be ranked higher over dissimilar objects [14]. Soleraet et al. [5] propose a method based on the Latent Structual SVM to partition a scene and learn which input features are most important in a divide and conquer fashion. While they use colour histograms in their experiments, their learning approach can adapt to any set of provided features. However, the divide step of their approach is based on spatial distance limiting their state-of-the-art performance to only static cameras.

More recently, Choi et al. [15] learnt an affinity model to measure if two detections correspond where the input is a set of low level feature trajectories, requiring that optical flow must first be computed. Probably the most similar work to ours is that of Bae and Yoon [16] which sequentially grows tracklets computing confidence from multiple sources. They learn to associate detections to tracklets using incremental discriminative learning analysis on features composed of both appearance and motion cues. Our method differs from theirs as we do not use motion information and our affinity model estimates a universal pairwise affinity measure while theirs treats each tracklet as a separate class.

In collecting training samples from test data we make use of the mutual exclusion constraint as a hard assumption in self-supervision. The *exclusion principle* was first introduced by [17] for the purpose of disambiguating two intersecting tracks. Similarly, [18] enforced the exclusion property at both the detection and trajectory levels using penalty terms within a conditional random field framework, however their work only considered the position of the detections. Position based exclusion methods are generally combined with other cues including motion and appearance [7] to become competitive. This work extends the notion of exclusion from these earlier works to reinforce the appearance affinity model to coherently distinguish multiple detections when collecting samples for training.

## III. Proposed Method

### A. Overview

The input to our system is a set of detections for each frame in a sequence. Although our approach is agnostic to the type of detector, we require that a description of the visual appearance is supplied for each detection. Convolutional network features are used as descriptors since they have been found to successfully capture visual similarity in related fields such as image retrieval [19] and clustering [20]. Additionally, when a convolutional network is used for the prior detection step (e.g. [21]), computing such descriptors comes for free as the computation is performed in the detection step. While any sufficient descriptor could be used to describe appearance, a detailed comparison with different descriptor types is beyond the scope of this paper.

Following the lines of [1], [2], [13], we formulate the frame-to-frame matching process as a bipartite graph. This strictly enforces a one-to-one matching among the detections between adjacent frames. The Hungarian algorithm [10] (also known as the Kuhn-Munkres method) is then used to find the optimal assignment which maximises the total affinity across all frame-to-frame matches. The overall performance of this assignment boils down to the quality of the affinity measure employed.

We concentrate on modelling the visual affinity and propose to formulate it as a binary classification problem. This affinity model takes a pair of visual descriptors to produce an output indicating whether the two samples match. The affinity model's output is then used as the matching score to drive the Hungarian based matching to associate detections to tracklets. In the remainder of this section we go into more detail of our approach and describe how we train this model in a self-supervised manner.

### B. Pairwise Features

To discriminate a pair of candidate patches as either matching or non-matching, a bidirectional feature is required. For pragmatic reasons, we simply compute the absolute difference between each element of the two patch descriptors as follows:

$$\mathbf{d}_{i,j} = |\mathbf{d}_i - \mathbf{d}_j| \qquad (1),$$

where $\mathbf{d}_{i,j}$ is a vector representing the pairwise descriptor computed from the descriptors $\mathbf{d}_i$ and $\mathbf{d}_j$ extracted from

detection patches $i$ and $j$ respectively. A comparison of various descriptor extraction techniques and methods for constructing pairwise feature vectors is beyond the scope of this paper, so unless stated otherwise this is the pairwise feature descriptor used throughout this work.

### C. Affinity Metric

The affinity is modelled with a linear logistic regression classifier [22] and trained on a set of candidate patch pairs. Given a descriptor describing the appearance of each patch, the goal is to learn which features within the descriptors are most informative for discriminating between a pair of matching and non-matching patches. The affinity model approximates the probability that two patches match denoted by:

$$p_a(m|\mathbf{d}_{i,j}) = \frac{1}{1+\exp^{-(\theta^T \mathbf{d}_{i,j}+b)}} \quad (2),$$

where $\theta$ denotes the weights corresponding to each feature and $b$ is a bias term. Both $\theta$ and $b$ are efficiently optimised through Stochastic Gradient Descent (SGD) on a set of matching and non-matching pairs.

### D. Collecting Initial Pairs

As with any classification problem, a set of training samples is required in order to train the affinity model. It is desirable to have a means of collecting training samples in any new environment for which this system is deployed for life-long learning. In the contexts of visual tracking, we can rely on the mutual exclusion constraint to gather non-matching pairs from each frame. Similarly, an initial set of positive pairs can be collected using optimal bipartite matching across frames and conservatively keeping only high confident matches. To achieve this however, an initial measure of similarity is required.

The cosine similarity metric is used as an initial measure of affinity since it possesses several desirable properties, including:

- When features are constrained to positive spaces, the cosine similarity is bounded to interval $[\mathbf{0}, \mathbf{1}]$. This reflects the probabilistic output range for our desired affinity model.
- Computation of cosine distance does not require expensive modelling of the data distribution, i.e. Mahalanobis distance.
- Considered to be more stable in higher dimensions than other measures based on Minkowski distance.

Fig. 2 shows that the cosine similarity metric is a reasonable measure as its maximal match is generally the same object. However, as the cosine similarity rapidly decays with increasing temporal windows we restrict this initial set of positive matching to adjacent frames. This initial set of samples is used to seed the affinity model with a single round of SGD. Next, we describe how these frame-to-frame associations are used to learn an affinity measure for estimating the matching likelihood over multiple frames.

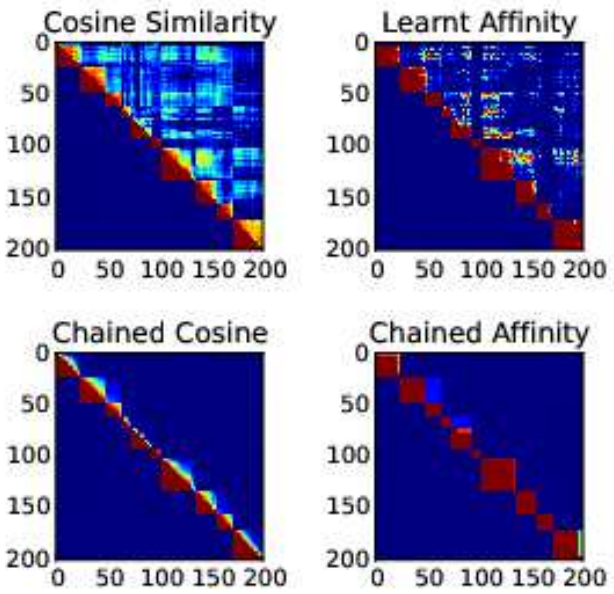


Fig. 2. A comparison between cosine similarity and the learnt affinity model after three iterations on a set of 200 tracklet patches of pedestrians taken from the KITTI benchmark [23]. The lower triangle represents the ground truth matching where the detections have been grouped and reordered such that each red block represents an instance of an object. The upper triangles show the affinity estimate (top row) and chained affinity (bottom row). This comparison demonstrates that having a near binary estimate is important in long association chains. Best viewed in colour.

### E. Association Chains

Using the initial affinity model as a similarity measure for frame-to-frame associations, we seek to extend its capability to match more challenging examples with higher variation as observed over larger temporal windows. To achieve this, we explore the frame-to-frame associations to incrementally grow a chain and select pairs along this chain to reinforce the affinity model. This process is provided in Algorithm 1 with key points described in this section.

Using the probability chain rule, the probability that two non-temporally-adjacent detections represent the same object can be modelled as:

$$p_c(m|\mathbf{d}_{i,j}) = \prod_{t=i}^{j-1} p_a(m|\mathbf{d}_{t,t+1}). \quad (3)$$

This *chained affinity* essentially captures the joint probability that all associated detections in the tracklet segment $[i, j]$ belong to the same object. As the temporal duration between the two detections $i$ and $j$ increases, there is a monotonic decrease in the chained affinity, accurately capturing the growing uncertainty. Put another way, the chained affinity has an upper bound which is less than the minimum link affinity along the chain. This is important in identifying the temporal boundary of an objects existence as shown in Fig. 2.

By exploring the tracklet history, errors in the direct affinity measure can be compared to the chained affinity and used to collect additional positive training samples over

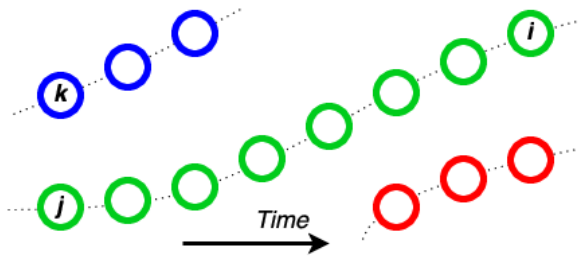


Fig. 3. Three tracklets where colour denotes object identity. By exploring the tracklet, detection $i$ in the current frame can be linked to detection $j$ several frames before by using the chained affinity to create a new positive sample pair $(i, j)$. Similarly, since $j$ and $k$ share the same frame we can also gather a non-matching pair $(i, k)$.

larger temporal windows. To prevent learning from potentially incorrect matches, only confident positive pairs are collected as measured by comparing the chained affinity to a threshold $\delta$ (line 6 of Algorithm 1). These samples are used to reinforce the affinity model through incremental learning – causing the frame-to-frame direct affinities to approach 1 – resulting in a reduction in the decay of the chained affinity – and ultimately increasing the chain for gathering additional samples.

Consistently adding new examples of positive matches from the temporal history can lead to an imbalance of the negative to positive ratio. The common remedy of increasing the weighting ratio between the negative and positive classes can be limiting as it is often difficult to estimate the ratio ahead of time. To alleviate this dissension, we build off the association chain affinity introduced in the last section to also find additional non-matching examples over longer temporal windows. The association chains are also used to collect additional negative samples to supplement the non-matching examples of the current frame collected by using the mutual exclusion constraint.

Given that we have formed a tracklet $[i, j]$ with high confidence, other non-overlapping detections in the same distant frame can also be used as negative match examples. The intuition of this is aligned with the notion of self-similarity, i.e. as the visual appearance of a single object changes over time so does the relative appearance of that object with respect to other objects. For example, Fig. 3 shows a matched tracklet chain $[i, j]$ where the older detection $j$ shares the same frame as detection $k$. Given that $i$ matched $j$ with a confidence of $p_c(m|d_{i,j})$, and that $j \neq k$ due to mutual exclusion, we expect that the direct affinity between $i$ and $k$ should be low (see lines 14 and 15). This places an upper limit on the affinity between $i$ and another non-overlapping detection $k$ from the same frame as $j$. Errors from the affinity model which violate this constraint are used as non-matching samples during the incremental updating of the model.

### F. Classifier Balance

Self-supervised frameworks are prone to drift, particularly where the model learnt is the same model used to gather examples. The intuition behind the mutual exclusion and

**Algorithm 1** Exploring tracklets for difficult samples

**Input:** $\mathbf{d}_i$ ▷ descriptor of current detection $i$
**Input:** $\mathbf{t}_{i-1}$ ▷ tracklet list matched to detection $i$
**Input:** $f_\theta()$ ▷ Eqn. 2
**Output:** $X, \mathbf{m}, \mathbf{w}$ ▷ Training inputs, outputs and weights

1: **function** $(\mathbf{d}_i, \mathbf{t}_{i-1}, f_\alpha)$
2: $p_c = f_\alpha(\mathbf{d}_{i,t_i})$
3: **for** $j = reverse\_iterate(\mathbf{t}_{i-1})$ **do**
4: $p_a = f_\alpha(\mathbf{d}_{i,j})$
5: $p_c = p_c * f_\theta(\mathbf{d}_{j-1,j})$ ▷ Eqn. 3
6: **if** $p_c(i,j) < \delta$ **then**
7: **break**
8: **if** $p_c > p_a$ **then**
9: $X \leftarrow \mathbf{d}_{i,j}$
10: $\mathbf{m} \leftarrow 1$
11: $\mathbf{w} \leftarrow (p_c - p_a)$
12: **for each** $k \in frame(j)$ **do**
13: **if** $overlap(j,k) \equiv False$ **then**
14: $p_n = 1 - f_\theta(\mathbf{d}_{i,k})$
15: **if** $p_c > p_n$ **then**
16: $X \leftarrow \mathbf{d}_{i,k}$
17: $\mathbf{m} \leftarrow 0$
18: $\mathbf{w} \leftarrow (p_c - p_n)$
**return** $X, \mathbf{m}, \mathbf{w}$

bipartite matching constraints offer some robustness to drift by selectively rejecting potential pairs which violate these constraints. However, the bipartite property assumes a one-to-one matching which does not factor in situations where objects enter or leave the scene and/or the detector misses or introduces false positives. These situations are actively handled by further rejecting matches if their affinity is less than 50%. If the affinity model is incorrect with this prediction, i.e. the pair are truly matched then this small failure can lead to severe consequences. This situation occurs if the model becomes negatively biased, which generates a ripple effect where the tracklet is broken, thus limiting the positive feedback effect of the association chain reinforcement causing the model to become further negatively biased.

*Not all errors are created equally!*

The key to combating this bias problem is in the placement of importance for the samples collected. Lines 11 and 18 of Algorithm 1 weight the positive and negative samples respectively. Both of these were vetted by first checking if the affinity model created an error by comparison with the chained affinity. This weighting of samples by the amount of error creates a balanced interplay between the positive and negative collection of samples, keeping the model neutral.

Finally, the samples collected over the test sequence are then used to retrain the logistic regression classifier. This improves the affinity model and ultimately enhances the performance of the Hungarian algorithm in assigning the frame-to-frame associations during the tracklet building process.

## IV. Experiments

The effect of this learnt affinity method is evaluated in the context of tracking-by-detection on various MOT benchmark sequences described in [24]. This MOT benchmark provides a unified framework for evaluating different multiple object trackers over a variety of sequences filmed from different viewpoints, with different lighting conditions, and different levels of target density.

In this experiment, we used the supplied detections provided with the dataset which where generated using the aggregated channel features (ACF) detector [25]. From each detection bounding box, we use the deep 16-layer convolutional network of [26] as a descriptor extractor. Each detection patch is resized to $224 \times 224$ and propagated forward through 13 convolutional layers and one fully-connected layer to produce a 4096 dimensional feature to represent the visual descriptor.

Both the Hungarian matching and logistic regression model implementation were from the scikit-learn toolkit [27]. Our unoptimised, single threaded python implementation of this tracker runs at 3.7 frames per second on a 2.5 GHz intel$^{TM}$ i7.

Table I shows the performance of the proposed method using the widely accepted CLEAR MOT metrics [28] evaluated with a 0.5 intersection over union threshold. The Multiple Object Tracking Accuracy (MOTA) combines all false positives, false negatives, and identity switches into a single number, and Multiple Object Tracking Precision (MOTP) measures the average distance between the ground truth and the tracker output. For both the MOTA and MOTP a higher value represents better performance. The ID switches indicate the total number of times a true object has switched identities according to the tracker output following the strict definition in [14]. Other metrics listed are the False Alarms per Frame (FAF), Mostly Tracked (MT) and Mostly Lost (ML) metrics along with the number of False Positives (FP) and False Negatives (FN). The number of Ground Truth object (GT) is also shown for each sequence.

The results are split into training and test sequences. It is important to note that the ground truth labels were not used within the self-supervised procedure. Decisions on selecting hyper-parameters such as the minimum confidence threshold $\delta$ and logistic regression regularisation were selected to increase the MOTA on these training sequences. The minimum confidence threshold $\delta = 0.9$ was found to give best results on the training sequences.

We also include the results of other tracking methods discussed in Section II. Namely a general tracking-by-detection method that uses the Hungarian algorithm TBD [2] and two affinity learning based methods: TC_ODAL [16], LDCT [5]. When considering the key MOTA score our approach performs quite favourably to the related method affinity methods indicating that our self-supervised approach is able to correctly adapt to the data observed. However, it is important to note that several of these sequences involve a dynamic camera. This heavily impacts the overall MOTA score for LDCT [5] which focuses primarily on applications with a static camera. Our method also achieves comparable results to a purely motion based technique [6], indicating that appearance is adequate in object tracking and could be treated as an independent low level tracker if combined with a motion based tracker.

Qualitative results are shown in Fig. 5. The vertical lines indicate an ID switch. With closer inspection (see Fig. 4), many of these are caused by a duplicate detection or false positive in a frame adjacent to a object entering of leaving the scene. Since this technique only matches detections in adjacent frames using visual appearance, ID switches and fragmentation errors are to be expected. Nevertheless, given the large number of tracklets and frames, the affinity model produces accuracy data association in the majority of frames. Furthermore, the white circles in Fig. 5, show detections which were not matched to either the previous or next adjacent frames. In the PETS sequence, it is obvious that these detections correspond to a stationary object which are actually false positives generated by the low level detector. This indicates that temporal inconsistency could be used as a cue for false positive detections which are implicitly handled within this technique as they are not matched to any object tracklet. A clear downside however is the amount of track fragmentation. This is caused by the frames where an object is miss-detected or when two detections cover an object creating a new tracklet. These limitations indicate that this method could be further improved if combined with motion based models.

Fig. 4. Frames 74, 77, and 80 of the KITTI-17 sequence showing a false positive in the top frame getting reassigned to an actual pedestrian emerging from behind a foreground pedestrian. Best viewed in colour.

TABLE I

PERFORMANCE OF THE PROPOSED APPROACH ON MOT BENCHMARK SEQUENCES.

| | | MOTA↑ | MOTP↑ | FAF↓ | GT | MT↑ | ML↓ | FP↓ | FN↓ | ID sw↓ | Frag↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Train Sequences** | TUD-Stadtmitte | 53.8 | 65.6 | 0.90% | 10 | 50.0% | 0.0% | 146 | 348 | 40 | 33 |
| | TUD-Campus | 35.7 | 70.6 | 0.34% | 8 | 12.5% | 12.5% | 51 | 157 | 23 | 25 |
| | PETS09-S2L1 | 67.7 | 71.4 | 0.87% | 19 | 78.9% | 0.0% | 648 | 603 | 193 | 152 |
| | ETH-Bahnhof | 29.3 | 73.4 | 1.52% | 171 | 23.4% | 37.4% | 1544 | 2121 | 165 | 208 |
| | ETH-Sunnyday | 30.9 | 76.2 | 0.29% | 30 | 10.0% | 50.0% | 118 | 1104 | 61 | 73 |
| | ETH-Pedcross2 | 7.9 | 71.3 | 0.17% | 133 | 0.0% | 85.0% | 173 | 5543 | 54 | 78 |
| | ADL-Rundle-6 | 25.4 | 71.8 | 2.09% | 24 | 4.2% | 8.3% | 974 | 2590 | 174 | 160 |
| | ADL-Rundle-8 | 10.9 | 72.4 | 2.51% | 28 | 10.7% | 39.3% | 1653 | 4213 | 178 | 187 |
| | KITTI-13 | 5.8 | 70.9 | 0.47% | 42 | 4.8% | 38.1% | 327 | 355 | 36 | 29 |
| | KITTI-17 | 48.0 | 70.8 | 0.14% | 9 | 0.0% | 11.1% | 44 | 292 | 19 | 21 |
| | Venice-2 | 13.3 | 72.6 | 3.81% | 26 | 11.5% | 19.2% | 2421 | 3591 | 177 | 164 |
| | **Overall** | 24.5 | 72.1 | 1.43% | 500 | 14.6% | 45.6% | 8099 | 20917 | 1120 | 1130 |
| **Test Sequences** | TUD-Crossing | 51.2 | 73.0 | 0.2% | 13 | 15.4% | 15.4% | 39 | 459 | 40 | 50 |
| | PETS09-S2L2 | 27.5 | 70.6 | 1.0% | 42 | 0.0% | 26.2% | 449 | 6153 | 385 | 359 |
| | ETH-Jelmoli | 32.9 | 73.2 | 0.6% | 45 | 13.3% | 28.9% | 283 | 1334 | 86 | 98 |
| | ETH-Linthescher | 15.4 | 74.1 | 0.1% | 197 | 1.5% | 76.1% | 94 | 7369 | 90 | 103 |
| | ETH-Crossing | 19.6 | 74.7 | 0.1% | 26 | 7.7% | 65.4% | 13 | 783 | 10 | 10 |
| | AVG-TownCentre | 13.3 | 70.0 | 1.0% | 226 | 1.3% | 61.1% | 442 | 5637 | 118 | 152 |
| | ADL-Rundle-1 | 5.1 | 71.3 | 7.0% | 32 | 15.6% | 21.9% | 3503 | 5010 | 321 | 306 |
| | ADL-Rundle-3 | 18.1 | 71.8 | 3.9% | 44 | 6.8% | 22.7% | 2420 | 5523 | 385 | 261 |
| | KITTI-16 | 13.9 | 72.1 | 0.5% | 17 | 0.0% | 23.5% | 105 | 1279 | 80 | 73 |
| | KITTI-19 | 13.5 | 66.5 | 1.1% | 62 | 6.5% | 30.6% | 1128 | 3266 | 227 | 326 |
| | Venice-1 | 12.5 | 71.9 | 1.7% | 17 | 0.0% | 41.2% | 757 | 3120 | 117 | 134 |
| *Proposed method | **Overall** | **17.0** | 71.2 | **1.6%** | 721 | 3.9% | 52.4% | 9233 | 39933 | 1859 | 1872 |
| ◇TC_ODAL [16] | **Overall** | 15.1 | 70.5 | 2.2% | 721 | 3.2% | 55.8% | 12970 | 38538 | 637 | 1716 |
| ◇LDCT [5] | **Overall** | 4.7 | **71.7** | 2.4% | 721 | 11.4% | 32.5% | 14066 | 32156 | 12348 | 2918 |
| ◇TBD [2] | **Overall** | 15.9 | 70.9 | 2.6 | 721 | 6.4% | 47.9% | 14943 | 34777 | 1939 | 1963 |
| †SMOT [6] | **Overall** | 18.2 | 71.2 | 1.5% | 721 | 2.8% | 54.8% | 8780 | 40310 | 1148 | 2132 |

* Purely appearance based
◇ Position and appearance based
† Purely motion based

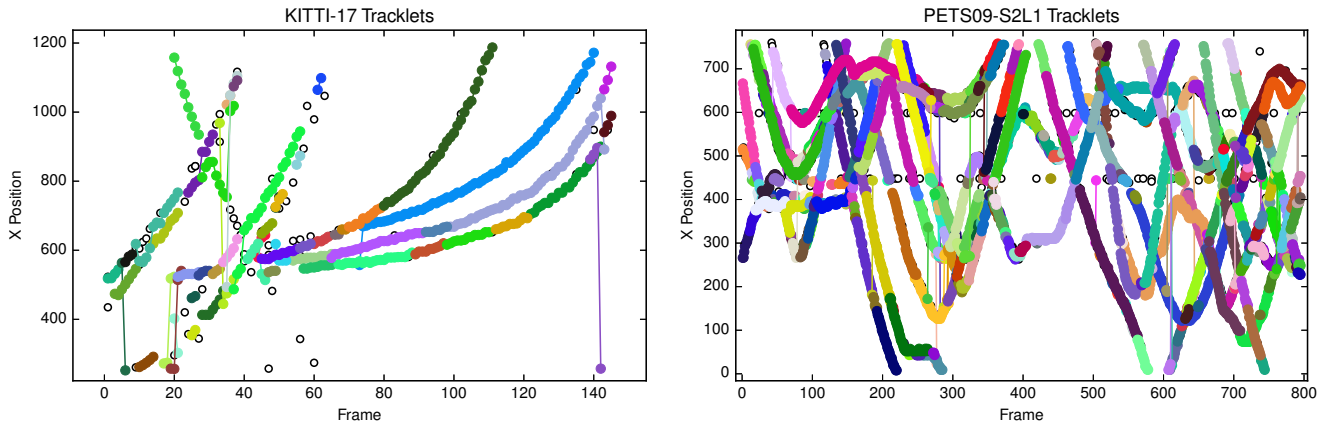


Fig. 5. Spatial-temporal representation of the tracklets discovered in a low target density setting (left) and high density setting (right). It is important to note that frame-to-frame matching was performed using appearance affinity making it robust to complex dynamics as observed in the PETS09-S2L1 sequence. Near vertical lines indicate an ID switch. Best viewed in colour.

## V. Conclusion

In this paper we have proposed a method to learn a visual appearance affinity model for the problem of associating object detections across video frames. The proposed method exploits the sequential nature of video sequences to collect additional training data as new frames arrive and explore existing associations for additional training pairs. We show that this affinity model can efficiently be updated from the test data without explicit labelling enabling life-long learning. Experimental evaluation shows that this appearance based affinity model is capable of operating as the primary association score for multiple object tracking. In future work, we intend on extending this affinity model to perform online tracking and learning. Additionally, incorporating motion information and tracklet merging strategies offer potential avenues for reducing track fragmentation.

## ACKNOWLEDGMENT

This work is supported by the Australian Coal Association Research Program (ACARP).

## References

[1] H. Zhang, A. Geiger, and R. Urtasun, "Understanding High-Level Semantics by Modeling Traffic Patterns," in *International Conference on Computer Vision (ICCV)*, 2013.

[2] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3D Traffic Scene Understanding from Movable Platforms," *Pattern Analysis and Machine Intelligence (PAMI)*, 2014.

[3] A. Torabi and G.-A. Bilodeau, "A Multiple Hypothesis Tracking Method with Fragmentation Handling," in *2009 Canadian Conference on Computer and Robot Vision*. Ieee, May 2009, pp. 8–15.

[4] P. Morton, B. Douillard, and J. Underwood, "Multi-sensor identity tracking with event graphs," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, May 2013, pp. 4742–4748.

[5] F. Solera, S. Calderara, and R. Cucchiara, "Learning to Divide and Conquer for Online Multi-Target Tracking," in *International Conference on Computer Vision (ICCV)*, 2015.

[6] C. Dicle, M. Sznaier, and O. Camps, "The way they move: Tracking multiple targets with similar appearance," in *International Conference on Computer Vision (ICCV)*, 2013.

[7] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking." *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 58–72, Jan. 2014.

[8] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2011, pp. 1201–1208.

[9] A. Bewley, V. Guizilini, F. Ramos, and B. Upcroft, "Online Self-Supervised Multi-Instance Segmentation of Dynamic Objects," in *International Conference on Robotics and Automation*. Hong Kong, China: IEEE, 2014, pp. 1296–1303.

[10] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[11] A. R. Zamir, A. Dehghan, and M. Shah, "Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs," in *European Conference on Computer Vision (ECCV)*, 2012.

[12] T. Finley and T. Joachims, "Supervised clustering with support vector machines," *Proceedings of the 22nd International Conference on Machine learning (ICML)*, pp. 217–224, 2005.

[13] S. Kim, S. Kwak, J. Feyereisl, and B. Han, "Online Multi-target Tracking by Large Margin Structured Learning," in *Computer Vision*, 7726th ed. Springer Berlin Heidelberg, 2013, pp. 98–111.

[14] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybrid-Boosted multi-target tracker for crowded scene," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, Jun. 2009, pp. 2953–2960.

[15] W. Choi, "Near-Online Multi-target Tracking with Aggregated Local Flow Descriptor," *arXiv 1504.02340v1*.

[16] S.-H. Bae and K.-J. Yoon, "Robust Online Multi-Object Tracking based on Tracklet Confidence and Online Discriminative Appearance Learning," *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[17] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, no. 1, pp. 57–71, 1999.

[18] A. Milan, K. Schindler, and S. Roth, "Detection-and Trajectory-Level Exclusion in Multiple Object Tracking," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013.

[19] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the Performance of ConvNet Features for Place Recognition," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, 2015.

[20] A. Bewley and B. Upcroft, "From ImageNet to Mining : Adapting Visual Object Detection with Minimal Supervision," *Proceedings of the 10th International Conference on Field and Service Robotics (FSR)*, 2015.

[21] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[22] S. H. Walker and D. B. Duncan, "Estimation of the probability of an event as a function of several independent variables," *Biometrika*, vol. 54, no. 1-2, pp. 167–179, 1967.

[23] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[24] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking," *arXiv preprint*, 2015.

[25] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast Feature Pyramids for Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[26] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2014.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in {P}ython," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[28] K. Bernardin and R. Stiefelhagen, "Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics," *EURASIP Journal on Image and Video Processing*, 2008.