2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Online Self-Supervised Multi-Instance Segmentation of Dynamic Objects

Alex Bewley[1], Vitor Guizilini[2], Fabio Ramos[2] and Ben Upcroft[1]

***Abstract*— This paper presents a method for the continuous segmentation of dynamic objects using only a vehicle mounted monocular camera without any prior knowledge of the object's appearance. Prior work in online static/dynamic segmentation [1] is extended to identify multiple instances of dynamic objects by introducing an unsupervised motion clustering step. These clusters are then used to update a multi-class classifier within a self-supervised framework.**

In contrast to many *tracking-by-detection* based methods, our system is able to detect dynamic objects without any prior knowledge of their visual appearance shape or location. Furthermore, the classifier is used to propagate labels of the same object in previous frames, which facilitates the continuous tracking of individual objects based on motion.

The proposed system is evaluated using recall and false alarm metrics in addition to a new multi-instance labelled dataset to measure the performance of segmenting multiple instances of objects.

I. Introduction

This paper addresses the problem of detecting and segmenting multiple dynamic objects simultaneously from a monocular video sequence, where the camera itself is moving within the scene. Motion segmentation remains as one of the fundamental computational challenges and is a critical perceptive capability for several robotic tasks, such as collision avoidance and path planning in dynamic environments. The approach taken in this paper uses a combination of unsupervised motion based clustering methods to supervise a multi-class classifier with training examples collected online.

Much research effort is being expended on object recognition based methods which use various supervised classifiers to train a predictive model off-line with a (preferably) large manually labelled training dataset [2], [3], focusing on the detection of a single class of object [4], [5], [6]. The performance of these methods is highly dependent on having a comprehensive training dataset, which contains sufficient number of labelled examples of objects of interest along with negative examples. It is costly and often impractical to obtain such a training set, where each class is known and completely represented for different view-points and lighting conditions. Instead, we collect training examples online in a self-supervised framework, without any prior knowledge of the object's shape, location or visual appearance.

The work presented here falls within the self-supervised classification category, however we restrict ourselves to the detection of only independently moving objects in the scene. Here we are not concerned with assigning semantic labels such as 'car' or 'human' to image regions. Rather we assume that any dynamic object is an obstacle and needs to be tracked in a dynamic motion planning framework. In order to predict where each dynamic object will likely be located in the future, it is desirable to separate the dynamic pixels into independent groups with similar motion. To achieve this, we go beyond the self-supervised binary classification of [1] to identifying multiple independent motion regions within an image as shown in Fig. 1.

Fig. 1: Example detection of multiple dynamic objects discovered using our proposed method that corresponds to the input image sequence shown above. The different colours (hue) in the output image represents the multiple object instances detected, while the intensity denotes the likelihood of the assigned class at each pixel. These objects were detected using only the motion of the scene and not any off-line models describing the visual appearance of the objects.

This paper addresses several challenges absent in the binary case. Firstly the input sequence need to be first segmented into multiple spatially consistent motion segments. Each motion segment must be assigned a temporally coherent class label before used to train a multi-class classifier. Finally multi-class classification is inherently more difficult than binary classification, which is further exacerbated by the non-stationary nature of video data.

The paper is organised as follows: In the next section we review relevant literature, followed by a brief overview of the proposed system in section III. In section IV we describe the static segmentation and motion clustering of sparse optical flow. In section V we detail the online process for using the sparse motion clusters to update a non-parametric model enabling temporally consistent inference over the entire image sequence. Section VI shows some results before conclusions and outlook to future work in section VII.

[1]A. Bewley and B. Upcroft are with the School of Electrical Engineering and Computer Science, Queensland University of Technology, Australia {aj.bewley,ben.upcroft}@qut.edu.au

[2]V. Guizilini and F. Ramos are with the School of Information Technologies, The University of Sydney, Australia {vgui2872,fabio.ramos}@sydney.edu.au

II. Relevant Literature

Over the last decade many algorithms have been developed to detect obstacles from a moving platform using vision [7]. These methods often combine object recognition and visual ego-motion estimation with occupancy maps. A tracking-by-detection framework is commonly employed utilising advancements in visual object recognition and can further be improved by utilising 3D tracking from stereo images [8]. However this method only detects pedestrians as it is trained off-line for detecting humans.

The Random Sample Consensus (RANSAC) [9] paradigm has been extended to model multiple motions from monocular vision simultaneously [10], however it is restricted to only fitting rigid objects. Kitt *et al*. use a similar two stage approach (RANSAC with an ensemble of extremely randomised decision trees) [11] however their method requires off-line training with hand-labelled training examples of all likely situations. Another approach is to estimate the full structure-from-motion of the camera with a robust estimation of the ground plane [12].

An alternative to learning appearance models of the target objects is to build statistical models the background appearance and temporal shifts online [13], [14]. Recent advancements in compressive sensing [15], [16] led to new approaches for extracting dynamic objects from monocular video sequences, however these approaches have only been demonstrated for the case of static or nominal camera motion.

The objective of this work is to segment regions of an image sequence corresponding to individual moving objects, observed from a mobile camera, without using offline training or prior knowledge of the location, shape or appearance of the target objects. To robustly model the camera motion we estimate the epipolar geometry of matched points within a RANSAC framework mirroring the initial binary classification step in [1]. We also take inspiration from [17] for clustering different motion regions before modelling the spatial location and colour of dynamic objects in addition to the static background.

III. System Overview

The work presented here builds on the binary dynamic classification work by Guizilini and Ramos [1] by extending it to segment multiple objects. An overview of the proposed system is illustrated in Fig. 2 and can be described using the following pipeline:

1) As in [1] sparse optical flow is computed by matching key-points from the current and previous frames as new images are acquired.
2) The optical flow vectors corresponding to the static environment are identified by fitting a motion model within a RANSAC framework to account for outliers.
3) The outliers of the previous step undergo density based clustering to identify independent motion in the scene and remove mismatched key-points.
4) The static and dynamic clusters are then used to incrementally update a multi-class non-parametric kNN model by matching each cluster to either an existing class or a new class.
5) This non-parametric model is then used to infer the object instance for any pixel in the image.

IV. Dynamic Object Discovery

The online detection of dynamic objects begins with an initial segmentation of the sparsely matched key-points from two consecutive frames. These motion segments provide a continuous source of training examples to update the dynamic object classifier described in the next section with new objects and new views of existing objects. The motion clustering happens in two stages, firstly separating static and dynamic points followed by clustering the dynamic points into groups representing consistent motion.

The sparse optical flow is computed by first detecting salient image features using both 'SURF' [18] and 'Good Features to Track' [19]. Using a combination of feature detectors provides reasonable coverage of the image space, including corners, edges and textured areas. The optical flow of these features is extracted by computing the 'BRISK' descriptor [20] of each point and compared to the features detected in the previous frame. This essentially extracts a sparse sampling of the optical flow across the image, providing a basis for motion segmentation described in this section.

A. Static and Dynamic Classification

The process of detecting dynamic objects begins with the binary classification of static and non-static key-points. In this step the global image motion is estimated to account for optical flow generated by the camera motion itself. When the camera is moving, the optical flow from static points in the world are constrained by the epipolar geometry of the two viewpoints.

We classify static points using the epipolar constraint that describes the motion of key-points from two viewpoints using the fundamental matrix [21]. A suitable technique for this is the RANSAC algorithm robust to outliers from the dynamic objects and has been demonstrated as a basis for online classification of static and dynamic points [1]. With sufficient key-points for the static environment, the RANSAC algorithm should elect the fundamental matrix that best represents the camera motion. Therefore, any matched key-point that lies on the epipolar line defined by both the estimated fundamental matrix and the corresponding key-point in the previous frame should belong to a static object. Point matches that do not fit the epipolar geometry are further processed as described in the next section.

B. Dynamic Point Clustering

So far the point correspondences have only been separated into static and dynamic binary classes. We extend this to include multiple instances of moving objects within the image, by grouping similar and separating dissimilar dynamic points based on motion. This grouping task can be formulated as a

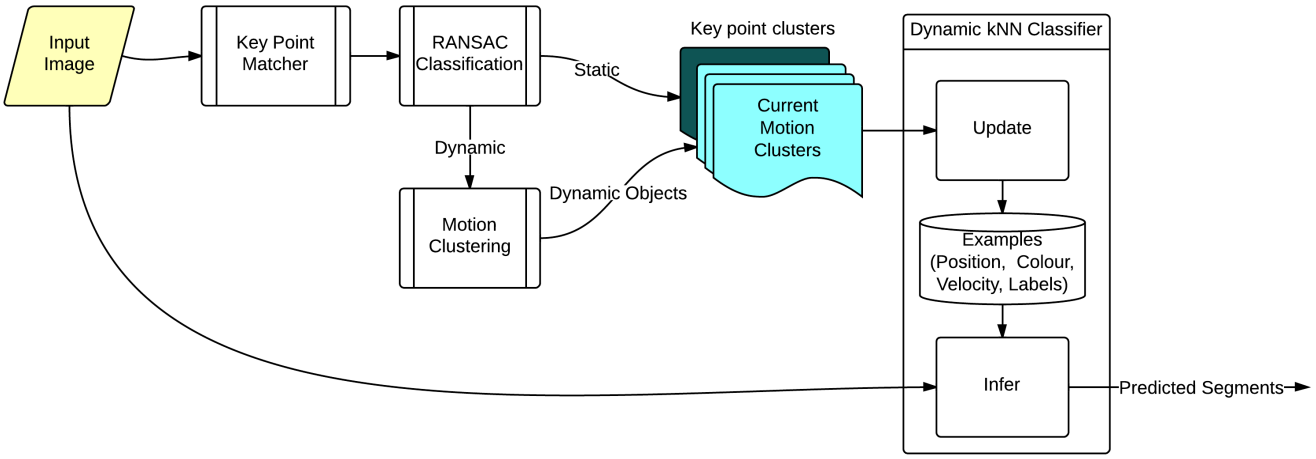

Fig. 2: An overview of the object extraction and learning process used in this paper.

clustering problem for an unknown number of clusters with noise introduced by mismatched key-points.

For characterising the motion of key-points, both the image location and optical flow velocity of the non-static features are used to represent the image motion of the corresponding object. By only considering the non-static features for clustering, we limit the computation required along with providing a larger separation between points. Let the motion feature vector w corresponding to each dynamic key-point describing both the image position(u, v) and inter-frame displacement $(\dot{u}, \dot{v})$ as:

$$w = [u, v, \dot{u}, \dot{v}]^T.$$

To cluster the dynamic points we chose an algorithm which doesn't require the number of clusters to be known *a priori*, suitable for non-rigid objects and can identify outliers which correspond to mismatched features. For this we choose to use the Density Based Spatial Clustering Analysis with Noise (DBSCAN) algorithm [22]. DBSCAN is governed by two parameters: a radius ϵ and the minimum number of points to form a cluster $minPts$ which essentially defines the minimum local density of a cluster.

The motion features for the set of dynamic points $(w_i, i \in DynamicSet)$ are used as the input to the DBSCAN algorithm. Using the average key-point density in the image we can evaluate suitable neighbourhood density parameters (ϵ and $minPts$) for grouping features corresponding to the moving objects in the scene. The density is estimated using the average key-point density in the 2D image and assuming that adjacent key-points on the same object will share similar inter-frame displacements as opposed to if they lie on different dynamic objects. For example if two points w_p and w_q on the same object have similar motion *i.e.* $\|w_p - w_q\| \approx \|(u_p, v_p) - (u_q, v_q)\|$ and the main difference lies in their relative image position. If p and q are from different objects or different parts of the image we expect high separation in position and motion space respectively thus reducing the likelihood these points would share a common neighbourhood.

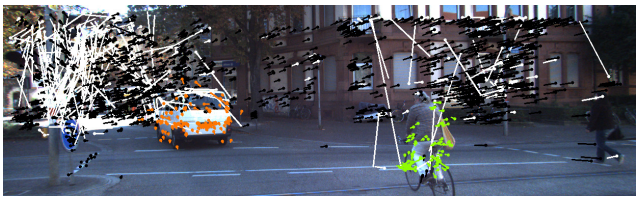

Fig. 3: Sparse optical flow vectors grouped by motion. Static points are shown in black, while the dynamic points are coloured by their cluster assignment. Points with low density in motion space (considered noise) are marked in white.

Using the average matched key-point density $(n/(width \times height))$ of the current frame we can automatically set the ϵ radius to be:

$$\epsilon^2 = \frac{minPts \cdot width \cdot height}{n\pi}, \quad (1)$$

where n is the number of matched key-points including static and dynamic in the frame. The $minPts$ parameter is explicit set to the minimum number of points we expect in a cluster. We found that setting $minPts$ to 10 is a good compromise between the number of false clusters and missing clusters.

Points with fewer than $minPts$ in their ϵ neighbourhood are considered as noise within the DBSCAN framework, unless on the boundary of a dense cluster. This attribute of DBSCAN essentially eliminates non-static points caused by mismatches in optical flow as observed in Fig. 3.

V. Dynamic KNN Classification

While the motion clustering method described in the previous section can identify both individual dynamic objects in the current frame and previously unseen objects, it has limited use for object tracking as it does not maintain any memory of which objects were previously identified. Here we introduce a self-supervised classifier for associating currently detected clusters with previously found objects. Knowledge of previous objects can be maintained for short durations if temporally occluded or when an object is missed due to the

number of matched key-points dropping below the $minPts$ threshold required by DBSCAN.

The k-nearest neighbour (kNN) classifier provides a suitable mechanism for this task, as it's capable of representing complex decision boundaries and naturally supports multi-class classification problems, while being simple to implement [23]. As the name suggests, the k nearest neighbour algorithm assigns a class label to an unlabelled test point by considering the distance to and frequency of labels amongst the k nearest neighbours in the model. This enables the kNN classifier to represent complex and non-Gaussian decision boundaries defined by the representative data. For finding the k nearest neighbours, we use the randomised kd-trees method described in [24] to achieve efficient search time with precision guaranteed in Euclidean space.

The kNN classifier is trained with a set of input feature vectors x_i that describe the image location and colour of the each key-point along with the assigned cluster number k_i. The input feature vector x is defined as:

$$x = [u, v, S \cdot cos(H), S \cdot sin(H), V]^T,$$

where H, S, V are the hue, saturation and intensity value (HSV) of the pixel located at (u, v) in the image. The HSV colour space is chosen over the RGB colour to limit sensitivity of lighting to a single channel.

For clarity, in this section we refer to *clusters* as the output of the unsupervised approach for the current frame and *class labels* as the temporally consistent moving object identifier stored in the kNN classifier. Also note that $k = 0$ represents the static cluster from RANSAC while $k = 1...K$ is a unique identifier for the individual dynamic clusters found using DBSCAN for the current frame.

This classifier is initialised with the initial clusters found in the first pair of frames and then incrementally updated there after. Clusters found in subsequent frames are used to continuously update the kNN non-parametric model allowing it to adapt to the changing dynamics of the scene using a continuous supply of training examples. However, the cluster numbers of the dynamic objects are arbitrary in the unsupervised framework and need to be matched to the class numbers representing previously discovered objects.

In the remainder of this section, we detail the inference method used for predicting new class labels, address the issues of cluster to class association and manage the growth of the model through selective updating and structured forgetting of uninformative points.

A. Inference

A drawback of the standard majority voting classification in this application arises due to the static class being exceptionally more frequent than any dynamic class and essentially dominates the feature space near the decision boundaries. To help alleviate this problem we use a probabilistic soft-max weighting function to evaluate the conditional probability of assigning the label c to test point x as follows:

$$p(c|x, N^k(x)) = argmax_c \left(\frac{\sum_{i \in c \cap N^k(x)} e^{-\|x_i - x\|^2}}{\sum_{i \in N^k(x)} e^{-\|x_i - x\|^2}} \right), \tag{2}$$

where $N^k(x)$ are the kNN points from x in the non-parametric model. This essentially weights the votes from each neighbour by their proximity to the test point.

B. Cluster Association

As subsequent frames provide new key-point examples of dynamic objects, these clusters need to be associated to previously seen objects or assigned to a new object. Evaluating the appropriate and temporally consistent class label for a given cluster is critical for the classifier to continuously build on its knowledge of a specific object. Here the classifier itself is used to predict the class labels of the new training clusters and resolve inconsistencies through information filtering.

Cluster association is achieved through an initial step of inferring the class label of each supplied training point and comparing its overlap ratio of class labels and cluster numbers. This is equivalent to assigning each cluster to the class with the highest Jaccard similarity coefficient [25]. Given all key-points of cluster K and the set of key-points classified with label C (denoting an existing object in the non-parametric model), the Jaccard coefficient is computed as:

$$J(K, C) = \frac{|K \cap C|}{|K \cup C|}. \tag{3}$$

The clustering result of the current frame is compared to the output of the dynamic kNN classifier to evaluate new objects and update existing objects. Given the cluster labels and predicted class assignments for each key-point provided in the current frame's training set, we take a greedy approach by remapping the entire cluster to the class number with the highest Jaccard coefficient; a point p_{ck} jointly assigned to class c and cluster k is reassigned to the c^* that has the highest Jaccard coefficient voted by all point in the same cluster.

The discrepancies between the predicted class labels and the consensus/remapped class labels can be further used to identify anomalies in the temporal consistency of the unsupervised clustering and identify clusters representing new/unseen objects. For example, in the top row of Fig. 4, a non-static cluster found corresponding to the entering car on the left, doesn't match any existing dynamic labels signifying the need for a new class label (represented as red in the output image). If this cluster was a false positive, it is expected that the cluster is not temporally coherent and that current and future static points located in the same region of the input space will rectify this scenario in the forgetting step.

C. Updating

Using the assigned labels from the cluster association, kNN learning is as simple as adding an example point to the

Fig. 4: Examples of temporal coherent label output of the proposed method. Top row shows the immediate detection of a car as it enters the scene denoted by the red cluster introduced in the middle frame. The bottom row demonstrates correct label assignments are maintained over multiple frames, even during temporary occlusion of the pedestrian on the right of the image.

non-parametric model. To minimise the unbounded growth rate of the model, new points are filtered before being added to the model. New points are only inserted if the inferred class at the feature location differs from the reassigned cluster label or if the local density of the input point is low.

Additionally, as the state of dynamic objects naturally change over time, it is desirable to reflect this behaviour in the classification process. This has many advantages over a stationary kNN classifier, particularly in regions of occlusion and dis-occlusion (see bottom row of Fig. 4). As we are already computing the sparse optical flow for our learning examples we can re-use this computation to set the temporal state of each individual training sample.

The kNN is updated by evaluating x_{t-1} at each key-point location in the previous image and x_t in the current image using the optical flow correspondence. The temporal partial derivative of each point is approximated as:

$$\delta x/\delta t = x_t - x_{t-1}.$$

After new examples are added to the kNN database the entire non-parametric model is updated using the partial derivatives:

$$x_{t+1} = x_t + \delta x/\delta t.$$

This keeps the learnt input values relevant as points move though the image and gradually change colour as lighting conditions change. At this point, we rebuild the kd-tree structure for efficient kNN inference on the updated point set.

D. Forgetting

As more points are added to the non-parametric model in the learning phase the speed of inference degrades. As the scene evolves over time many points added to the model become irrelevant and overcrowded by points with mixed labels. This has a detrimental effect on the speed of inference as the size of the model continues to grow. Unlimited growth is restricted in a number of ways:

1) Firstly, we actively search for irrelevant data points by performing inference on each point in the model and removing points classified with a different class to their assigned labels. This is a common outlier detection method used for kNN and is analogous to the Gaussian Process feature filtering component of [1].
2) Secondly, as we propagate points in our model, points with non-zero velocity eventually exit the input volume bounded by the image dimensions. This input boundary is evaluated and expanded by checking the minimum and maximum values of each input dimension. Any point in the model propagated outside the boundary by a user selected margin is considered irrelevant and is discarded. The default value of this margin is set to the maximum average velocity defined by the optical flow evaluated in a similar fashion to the boundary itself.
3) Finally, as misclassified points are removed in the first step there is the potential the area would be over represented by points with uniform labels. These points are removed by limiting the maximum density within the non-parametric model with uniform class labels.

VI. Experimental Results

To evaluate the proposed method we first consider its performance in segmenting dynamic from static objects and compare to other online learning methods. Then we measure the performance of continuously segmenting dynamic objects on a multi-instance basis using a hand annotated sequence taken from the KITTI dataset [3].

A. Online static vs dynamic classification

For comparison to other static/dynamic binary classification methods we use the receiver operator characteristic

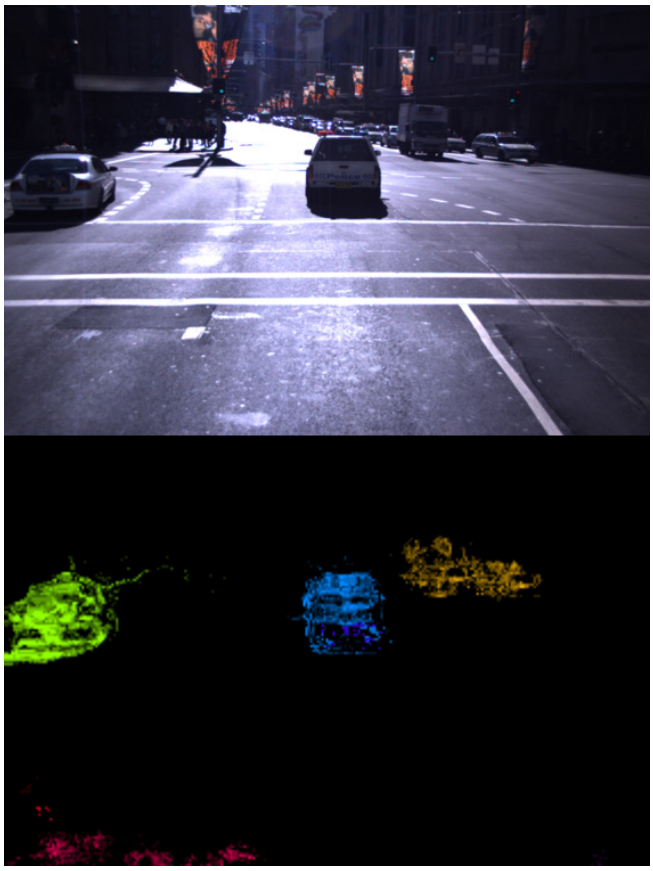

Fig. 5: Detected dynamic objects from the Sydney dataset. In this frame the truck and car on the opposite side of the intersection are considered as a single object as they share similar motion. The red object is a false positive due to the lack of texture on the road. It is expected that this system would be combined with a ground plane estimator to remove these false positives and shadows if deployed specifically for road applications.

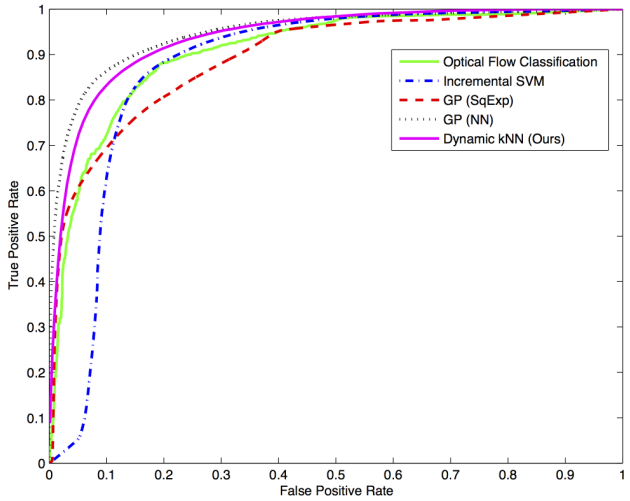

Fig. 6: Receiver Operator Characteristic (ROC) curves over 100 frames from the Sydney dataset and comparison to other online dynamic object segmentation algorithms including: an optical flow based classification of [26], Incremental SVM [27] and the Gaussian Process methods from [1]. Our method outperforms all other techniques except for [1]. However, it must be noted that all these methods are binary classifiers whereas ours includes multiple instances. The effectiveness of multi-instance classification is evaluated in the following experiments.

(ROC) curve, generated by varying the discriminative probability threshold on $1-p(static|x)$ (of equation 2). A larger area under this curve indicates a better overall performance in all threshold levels. This provides a graphical illustration of the true positive rate vs. the false positive rate. To evaluate the ROC curve performance for the proposed system we use the same ground truth dataset of [1], consisting of 100 frames taken from various sections of a 1500 frame dataset taken from a moving vehicle around the city of Sydney. This dataset contains a significant variance in lighting conditions, as the camera moves in and out of building shadows creating high contrast in the visual appearance of dynamic objects (see Fig. 5). With the exception of [1], Fig. 6 shows that our proposed method outperforms several other online techniques including an optical flow based classification of [26]. However, it should be noted that all these other methods are binary classifiers and their *one-vs-all* multi-class equivalent would generally require a single instance classifier for each cluster found as opposed to the kNN classifier which naturally handles multi-class data.

Fig. 7 shows how the accuracy varies in terms of area under the ROC curve for different values of k. The improvement shown by increasing k can be contributed to the smoothing effect of sampling more neighbours, which is beneficial in texture-less areas with a high false positive rate. This gain is ultimately limited as k becomes significantly larger than the number of samples on small and distant object leading to miss detections.

B. Online motion clustering

Since the Sydney dataset was originally produced for static / dynamic classification, it only contains binary labels in the form of an image mask. This paper is mostly concerned with not only detecting a moving object, but also discriminating between each dynamic object. Due to the complexity of labelling for multiple instances of dynamic objects, existing hand labelled dataset with pixel-wise semantic labels are not suited to evaluate the effectiveness of our algorithm. We address this by hand labelling the bike image sequence from [3] with pixel-wise annotations of the individual objects and use the V-measure proposed in [28] as a guide of how well the system can segment these objects. The V-measure is chosen as it combines two desirable aspects of clustering, homogeneity (each cluster contains only members of a single class) and completeness (all members of the same class are contained in a single cluster), without explicitly assigning semantic labels to each cluster. Our online motion clustering system attains a V-measure of 0.24 comprised of a completeness score of 0.31 and homogeneity score of 0.19 as defined in [28].

Since the system doesn't go as far as assigning semantically meaningful labels to each cluster, we evaluate the system's performance in handling multiple instances by visualising the cluster purity and completeness in Fig. 8. This plot shows the true object composition spread across the static cluster (left most column) and the ten largest dynamic

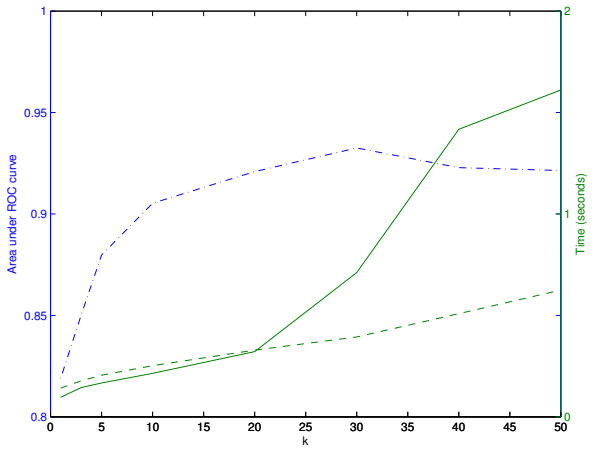

Fig. 7: Comparing performance (dotted dashed blue) along with the update (solid green) and inference (dashed green) computation times for different number of k neighbours on the Sydney dataset. The performance is measured by the area under the ROC curve such that larger values indicate a better overall performance in terms of true positive and false positive rates for all threshold levels. This experiment shows that increasing k improved the overall performance up to $k = 30$, however, this improvement is at the cost of longer computation times.

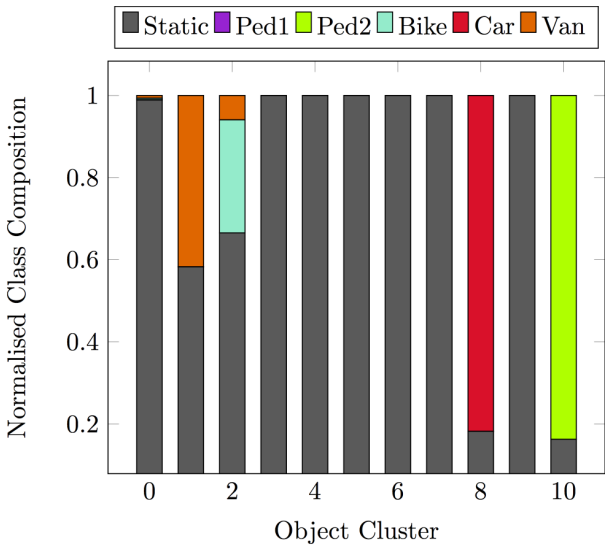

Fig. 8: Visual representation of the class/cluster composition over the KITTI Bike Sequence with individually pixel-wise annotated objects. The columns represent the clusters found by the algorithm with cluster 0 the static cluster and the colours represent the ratio of true class labels for pixels of that cluster.

clusters (in terms of the number of pixels). The colours in each bar represents the true object instance label with the hight of each interval denoting the ratio of overlap between that true class and the object cluster. The bar magnitudes have been normalised to aid visibility of object instances which are only present in a few frames, such as the car and pedestrians.

This plot can be interpreted as: the first pedestrian was not detected, the other true classes are generally contained within a single object cluster. The van and bike classes have a significant portion of the static background within this cluster as the white van shares similar colour to the nearby saturated road surface and for the bike a few key-points around the wheels tend to have the same colour as the road making the colour spread to the texture-less road where there are few true static key-points to correct this. After the system has had time to sufficiently sample the static background the introduction of new dynamic objects such as the car (see top row of Fig. 4) and second pedestrian tend to be more homogeneous. This is largely due to colour change in the localised region input space near the new object tends to be under represented by the static class. We have also run this system on different image sequences to consider it's generality and in environment where there is good contrast between the colour and the environment we observe that the system can accurately segment different dynamic objects (see Fig. 9).

C. Computational cost

A prototype of this algorithm was implemented in C++, making extensive use of the OpenCV 2.4 library, and was deployed on an Intel i7 machine with 8GB RAM. The total time from image acquisition to training and then inferring the

Fig. 9: Moving segments of mining dataset. This shows all moving parts of the scene has been accurately detected. The truck undergoing loading is stationary.

presence of dynamic objects across the entire image takes 3.5 seconds, with the breakdown times shown in Table I. This was evaluated by taking the average frame processing time for a typical image sequence from the Sydney [1] and KITTI [3] datasets, with resolution shown in pixels and time in seconds. The two main components which consume the majority of the time are the key-point matching and the kNN training. There are many alternative point trackers which can supply sufficient number of key-points correspondences that when implemented on parallel hardware is capable of real-time performance (e.g. [29]). Furthermore, other forms of efficient indexing, such as Locality Sensitivity Hashing [30], could potentially be a faster choice for nearest neighbour searching in various components of this application, requiring further investigation.

Dataset	Sydney (640 x 480)	KITTI (1242 x 375)
Point Matching	0.35	0.98
RANSAC Fitting	0.09	0.19
Motion Clustering	0.02	0.02
kNN Training	0.63	2.01
Dense kNN Inference	0.11	0.31
Total	1.20	3.51

TABLE I: Average execution times by section (seconds per frame) with $k = 30$.

VII. Conclusion

In this paper we have proposed a method for combining unsupervised motion clustering in a self-supervised framework, for the purpose of segmenting multiple dynamic objects from a monocular image sequence. Furthermore, the segments from each frame are made temporally coherent through the continuous inference, remap and update learning cycle. We have evaluated this approach quantitatively using the original 100 frame dataset of [1], in addition to qualitatively evaluating the multi-class performance on a range of datasets which differ in context. While the binary dynamic segmentation of [1] has better detection performance, we believe this to be an important step towards multiple object tracking and ultimately instance based object characterisation in an unsupervised framework.

In future work, we plan to optimise the individual system components further with respect to run-time and performance. Additionally, the ability to segment whole objects in an unsupervised framework opens many opportunities in object tracking and visual appearance learning.

ACKNOWLEDGMENT

This research has been supported by the Australian Coal Association Research Program (ACARP). The authors would also like to give thanks to Lionell Ott for providing access to a custom cluster assignment plotting tool.

References

[1] V. Guizilini and F. Ramos, "Online self-supervised segmentation of dynamic objects," in *International Conference on Robotics and Automation*, 2013.

[2] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2012, pp. 3354–3361.

[4] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Jun. 2008.

[5] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *International Conference on Computer Vision*, no. Iccv. IEEE, Sep. 2009, pp. 32–39.

[6] P. E. Rybski, D. Huber, D. D. Morris, and R. Hoffman, "Visual classification of coarse vehicle orientation using Histogram of Oriented Gradients features," in *2010 IEEE Intelligent Vehicles Symposium*. Ieee, Jun. 2010, pp. 921–928.

[7] S. Wangsiripitak and D. Murray, "Avoiding moving outliers in visual SLAM by tracking moving objects," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, May 2009, pp. 375–380.

[8] A. Ess, B. Leibe, K. Schindler, and L. V. Gool, "Moving obstacle detection in highly dynamic scenes," in *International Conference on Robotics and Automation*. IEEE, 2009.

[9] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[10] K. Ni, H. Jin, and F. Dellaert, "GroupSAC: Efficient consensus in the presence of groupings," in *International Conference on Computer Vision*. IEEE, Sep. 2009, pp. 2193–2200.

[11] B. Kitt, F. Moosmann, and C. Stiller, "Moving on to dynamic environments: Visual odometry using feature classification," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2010, pp. 5551–5556.

[12] K. Yamaguchi, T. Kato, and Y. Ninomiya, "Vehicle Ego-Motion Estimation and Moving Object Detection using a Monocular Camera," *International Conference on Pattern Recognition (ICPR)*, pp. 610–613, 2006.

[13] K. Patwardhan, G. Sapiro, and V. Morellas, "Robust foreground detection in video using pixel layers," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 30, no. 4, pp. 746–751, Apr. 2008.

[14] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection: A systematic survey," *Recent Patents on Computer Science*, vol. 4, no. 3, pp. 147–176, 2011.

[15] R. Sivalingam, A. D'Souza, M. Bazakos, and R. Miezianko, "Dictionary learning for robust background modeling," in *International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 4234–4239.

[16] C. Lu, J. Shi, and J. Jia, "Online robust dictionary learning," in *Computer Vision and Pattern Recognition*. IEEE, Jun. 2013, pp. 415–422.

[17] D. Almanza-Ojeda, M. Devy, and A. Herbulot, "Active method for mobile object detection from an embedded camera, based on a contrario clustering," *Informatics in Control, Automation and Robotics*, vol. LNEE 89, pp. 267–280, 2011.

[18] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.

[19] J. Shi and C. Tomasi, "Good features to track," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE Comput. Soc. Press, 1994, pp. 593–600.

[20] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *International Conference on Computer Vision (ICCV)*. IEEE, Nov. 2011, pp. 2548–2555.

[21] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[22] M. Ester, H.-p. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *International Conference on Knowledge Discovery and Data Mining*, 1996.

[23] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2001.

[24] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Application*. INSTICC Press, 2009, pp. 331–340.

[25] R. Real and J. M. Vargas, "The probabilistic basis of Jaccard's index of similarity," *Systematic Biology*, vol. 45, no. 3, pp. 380–385, 1996.

[26] C. Liu, "Beyond Pixels: Exploring New Representations and Applications for Motion Analysis," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

[27] C. Diehl and G. Cauwenberghs, "Svm incremental learning, adaptation and optimization," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 4. IEEE, 2003, pp. 2685–2690.

[28] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, no. June, 2007, pp. 410–420.

[29] M. Garrigues and A. Manzanera, "Real time semi-dense point tracking," *Image Analysis and Recognition*, vol. 7324, pp. 245–252, 2012.

[30] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Symposium on Computational Geometry*, 2004.