2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Multi-Sensor Clustering using Layered Affinity Propagation

Lionel Ott[1] and Fabio Ramos[2]

Abstract— **Current robotic systems carry many diverse sensors such as laser scanners, cameras and inertial measurement units just to name a few. Typically such data is fused by engineering a feature that weights the different sensors against each other in perception tasks. However, in a long-term autonomy setting the sensor readings may change drastically over time which makes a manual feature design impractical. A method that can automatically combine features of different data sources would be highly desirable for adaptation to different environments. In this paper, we propose a novel clustering method, coined Layered Affinity Propagation, for automatic clustering of observations that only requires the definition of features on individual data sources. How to combine these features to obtain a good clustering solution is left to the algorithm, removing the need to create and tune a complicated feature encompassing all sources. We evaluate the proposed method on data containing two very common sensor modalities, images and range information. In a first experiment we show the capability of the method to perform scene segmentation on Kinect data. A second experiment shows how this novel method handles the task of clustering segmented colour and depth data obtained from a Velodyne and camera in an urban environment.**

I. Introduction

With robots becoming more widely deployed in settings where they are left unattended for extended periods of time, long-term autonomy capabilities become increasingly important. For robotic systems to be able to function autonomously for extended periods of time they need to be able to detect and adapt to changes in the environment without supervision. This obviously means that methods depending on supervised training are less desirable as they either cannot adapt or require frequent retraining. Clustering methods are a very prominent method of unsupervised learning which can be used to group observations according to their similarity. However, when multiple data sources (such as laser scans and images) need to be clustered jointly, selecting a feature set that balances the contribution of each data source becomes challenging. One possibility is to manually engineer such a feature set by combining multiple simpler ones through carefully tuning their relative weights. The resulting feature is likely to not generalise well and makes an unsupervised method dependant on human supervision. A better solution would be an algorithm which can automatically combine simpler features without requiring any manual tuning by the user.

In this paper we propose a novel clustering method for multi sensor data, called layered affinity propagation (LAP). As the name suggests the method builds on affinity propagation, a recent clustering method introduced by Frey and Dueck [3] that clusters the data by propagating messages over an initially fully connected graph where each node represents a point. The term layered refers to the fact that we represent the similarity values derived from each data source and the associated features by a separate affinity propagation instance or layer. Broadly speaking our method consists of the following two parts:

1) data layers, each representing a single data source,
2) a merging layer ensuring an overall consistent clustering solution.

During clustering the data layers are all updated independently followed by an update of the merging layer. This cycle is repeated until convergence is achieved. Because we do not explicitly define how the different features are merged, we can use simple, well known methods to obtain similarities between data points for a single data source. The difficult part, which is to decide how to best combine the different data sources, is left to the algorithm.

The main contribution of this work is a novel, principled way to perform clustering of multiple data sources by message propagation. In the experiments we show that the method is capable of handling data from different sensors and properties with significant better clustering quality than competing alternatives. A first set of experiments demonstrates the capability to perform scene segmentation on RGB-D data collected indoors with a Kinect. In the second experiment we cluster segments extracted from a Velodyne and camera combination in urban settings into groups of similar appearance.

II. Related Work

In machine learning, there have been several extensions to affinity propagation addressing hierarchical clustering. Xiao et al. [16] propose a greedy hierarchical model in which each subsequent layer is based on the exemplars of the previous layer. The method proposed by Givoni et al. [6] uses a graphical model which connects subsequent layers and shows it to outperform the simple greedy approach. A two-layer hierarchical model is proposed by Wang et al. [15]. Their method jointly finds exemplars as well as the clustering of those. These last two methods derive a set of update messages from a graphical model. Our work uses a lateral rather then a hierarchical model but is also based on a graphical model from which we derive update messages.

There are also other methods designed to cluster data from multiple sources. Zhang et al. [17] propose a markov random field model with mutual information as potential functions to

[1]Lionel Ott is with the Australian Centre for Field Robotics, School of IT, University of Sydney l.ott at acfr.usyd.edu.au

[2]Fabio Ramos is with the Australian Centre for Field Robotics, School of IT, University of Sydney f.ramos at acfr.usyd.edu.au

cluster data with multiple modalities. Another method based on spectral clustering is proposed by Bekkerman and Jeon [2] who build a k-partite graph based on the input data. This graph is then used to derive the matrices needed by spectral clustering.

In another line of research, the combination of different sensor modalities has been shown to improve perception performance. Triebel et al. [14] process point cloud data into a mesh and compute features to segment scenes and identify objects in the scene. Jebari and Filliat [8] performed object segmentation by combining depth and colour features of superpixels with a Markov random field. Schoenberg et al. [12] presented another method employing a Markov random field to cluster coloured 3D point clouds in order to segment urban scenes. In Howard et al. [7] texture and geometric features are used to learn terrain type and traversability from stereo camera data. The clustering of these features is performed using a distance function that weights the different features. A different approach to learning terrain traversability is taken by Sun et al. [13] who use shape and colour as features. Those are then clustered using an ad-hoc clustering method based on the feature similarity. Katz et al. [9] use a linear weighted combination of visual and laser stamps to detect dynamic obstacles in the environment. In all of these methods there is a requirement to manually define how the different features are to be combined. Our method, in contrast, only requires the definition of features for each data source. How these are combined is a task automatically solved by the algorithm.

III. Affinity Propagation

In this section we give a short introduction to the original affinity propagation algorithm [3] using the alternative derivation proposed by Givoni and Frey [5]. We derive our novel clustering method, layered affinity propagation, using this alternative binary variable model in the next section.

Affinity propagation is a clustering method capable of determining the number of clusters directly from data. Given pairwise similarity values S_{ij} between data points i and j, a graphical model is built on which message propagation is used to optimise the energy function. By optimising the energy function, a clustering solution is found consisting of exemplars (the most representative point of a cluster) and the assignment of data points to these exemplars.

The graphical model consists of nodes associated to binary variables $\{h_{ij}\}_{j=1}^{N}$ corresponding to each data point $i \in \{1,\dots,N\}$, with $h_{ij} = 1$ iff j is the exemplar of point i. Thus the clustering solution is described by the N^2 binary variables $\{h_{ij}\}$ with $i,j \in \{1,\dots,N\}$. In order to find good solutions two types of constraints are added:

1) 1-of-N Constraint (I_i). Each data point has to choose exactly one exemplar.
2) Exemplar Consistency Constraint (E_j). For point i to select point j as its exemplar, point j must declare itself an exemplar.

These constraints can be formulated mathematically as follows:

$$I_i(h_{i:}) = \begin{cases} 0 & \text{if } \sum_j h_{ij} = 1 \\ -\inf & \text{otherwise} \end{cases} \tag{1}$$

$$E_j(h_{:j}) = \begin{cases} 0 & \text{if } h_{jj} = \max_i h_{ij} \\ -\inf & \text{otherwise} \end{cases} \tag{2}$$

where $h_{i:} = h_{i1},\dots,h_{iN}$ and $h_{:j} = h_{1j},\dots,h_{Nj}$.

Combining these constraints with the user provided pairwise data similarities S_{ij}, the following energy function is maximised:

$$T(\{h_{ij}\}) = \sum_{i,j} S_{ij}(h_{ij}) + \sum_i I_i(h_{i:}) + \sum_j E_j(h_{:j})\,. \tag{3}$$

In order to optimise this energy function the max-sum algorithm is used [10] to recover the maximum a posterior (MAP) assignments of the h_{ij} variables. Denoting f as a factor, or a function of a subset of variables, the following messages can be defined:

$$\mu_{v\to f}(x_v) = \sum_{f^*\in ne(v)\setminus f} \mu_{f^*\to v}(x_v)\,, \tag{4}$$

$$\mu_{f\to v}(x_v) = \max_{x_1,\dots,x_M} \Big[f(x_v, x_1,\dots,x_M) + \sum_{v^*\in ne(f)\setminus v} \mu_{v^*\to f}(x_{v^*}) \Big]\,, \tag{5}$$

where $\mu_{v\to f}(x)$ is the message sent from node v to factor f, $\mu_{f\to v}(x_v)$ is the message from factor f sent to node v, $ne()$ is the set of neighbours of the given factor or node and x_v is the value of node v.

In Figure 1a it can be seen that each node h_{ij} is connected to three factors S_{ij}, I_i and E_j. This shows that the messages ρ_{ij} and β_{ij} are sent from nodes to factors and thus are derived using Eq. (4). The other three messages s_{ij}, α_{ij} and η_{ij} go from factor to node and need to be derived using Eq. (5). Since we are using binary variables they can only take on two values, 1 and 0. Therefore, it is sufficient to compute the difference between the two variable settings. This simplification together with constraints imposed by the energy function allows us to derive the following messages shown in Figure 1a:

$$s_{ij} = S_{ij} \tag{6}$$

$$\beta_{ij} = s_{ij} + \alpha_{ij} \tag{7}$$

$$\eta_{ij} = -\max_{k\neq j} \beta_{ik} \tag{8}$$

$$\rho_{ij} = s_{ij} + \eta_{ij} \tag{9}$$

$$\alpha_{ij} = \begin{cases} \sum_{k\neq j} \max(0,\rho_{kj}) & i = j \\ \min\Big[0, \rho_{jj} + \sum_{k\notin\{i,j\}} \max(0,\rho_{kj})\Big] & i \neq j. \end{cases} \tag{10}$$

We can further simplify this by expressing ρ_{ij} as follows:

$$\rho_{ij} = s_{ij} + \eta_{ij} = s_{ij} - \max_{k\neq j} \beta_{ik} = s_{ij} - \max_{k\neq j}(s_{ik} + \alpha_{ik}) \tag{11}$$

recovering the availability (α) and responsibility (ρ) messages of the original affinity propagation formulation [3]. In

order to find the MAP assignments we initialize all messages α_{ij} and ρ_{ij} to 0 and then iteratively compute ρ_{ij} and α_{ij} until convergence. Upon convergence we find the exemplars as the entries for which $(\alpha_{ii} + \rho_{ii}) > 0$ holds.

IV. Layered Affinity Propagation

Our proposed method is based on affinity propagation but optimises a different energy function resulting in the graphical model shown in Figure 1c. The basic idea is to use multiple data layers, Figure 1c (left), that represent the data from the different sensors. These layers are then combined via a merging layer, Figure 1c(right). This structure allows us to find solutions that are optimal when considering the layers jointly. It is important to note that this is not a hierarchical clustering approach. The different data layers influence each other indirectly through the merging layer, which interconnects them. The messages involved in this model are shown in Figure 1b. For comparison purpose the messages used in standard affinity propagation are depicted in Figure 1a.

Comparing the messages exchanged by affinity propagation and layered affinity propagation we can see that the data layers L are very similar to standard affinity propagation. Both methods have a factor node I which ensures that every data point is assigned to exactly one cluster. The difference comes from the Q factor node, which replaces the E factor node. This factor enables communication between the data layers and the merging layer. The role though stays the same with the addition that the exemplar consistency constraint is enforced over the entire network. The merging layer has a few more differences as it merges the information from all data layers through the Q nodes into its own cluster assignments. It also accesses the similarities $S^{:}_{ij}$ of all data layers. While the model allows the values of $S^{:}_{ij}$ to differ between the data layer and the merging layer we have kept the values identical. The merging layer uses the information from the data layers to come up with its own decision which is fed back into the data layers and thus information is shared between all layers. All this is encoded in the following energy function which we optimise again by finding its MAP assignments:

$$\begin{aligned} T(\{h^{:}_{ij}, \widetilde{h}_{ij}\}) = &\sum_{i,j,l} S^l_{ij}(h^l_{ij}) + \sum_{i,l} I^l_i(h^l_{i:}) \\ &+ \sum_i \widetilde{I}_i(\widetilde{h}_{i:}) + \sum_{j,l} Q^l_j(h^l_{:j}, \widetilde{h}_{:j}) \,, \end{aligned} \tag{12}$$

where h^l_{ij} is the binary variable for point i and j in layer l, $\widetilde{h}_{ij}$ is the binary variable for point i and j in the merging layer. The different terms of the energy function are defined as follows:

$$I^l_i(h^l_{i:}) = \begin{cases} 0 & \text{if } \sum_j h^l_{ij} = 1 \\ -\inf & \text{otherwise} \end{cases} \tag{13}$$

$$\widetilde{I}_i(\widetilde{h}_{i:}) = \begin{cases} 0 & \text{if } \sum_j \widetilde{h}_{ij} = 1 \\ -\inf & \text{otherwise} \end{cases} \tag{14}$$

$$Q^l_j(h^l_{:j}, \widetilde{h}_{:j}) = \begin{cases} 0 & \text{if } h^l_{jj} = \max_i h^l_{ij} \\ & \wedge h^l_{jj} = h^k_{jj} \forall l, k \\ -\inf & \text{otherwise} \end{cases} \tag{15}$$

We can now derive the messages shown in Figure 1b using the same process as for affinity propagation messages. The messages of the data layers are:

$$s^l_{ij} = S^l_{ij} \tag{16}$$

$$\beta^l_{ij} = s^l_{ij} + \alpha^l_{ij} \tag{17}$$

$$\eta^l_{ij} = -\max_{k \neq j} \beta^l_{ik} \tag{18}$$

$$\rho^l_{ij} = s^l_{ij} + \eta^l_{ij} \tag{19}$$

$$\begin{aligned} \alpha^l_{ij} = \min \Big[&0, \rho^l_{jj} + \tau^l_{jj} + \\ &\sum_{k \notin \{i,j\}} \max(0, \rho^l_{kj}) + \sum_{k \neq j} \max(0, \tau^l_{kj}) \Big] \end{aligned} \tag{20}$$

$$\alpha^l_{jj} = \tau^l_{jj} + \sum_{k \neq j} \max(0, \rho^l_{kj}) + \sum_{k \neq j} \max(0, \tau^l_{kj}) \,. \tag{21}$$

While the messages of the merging layer have the following form:

$$\widetilde{\beta}_{ij} = \sum_t s^t_{ij} + \sum_t \phi^t_{ij} \tag{22}$$

$$\widetilde{\eta}_{ij} = -\max_{k \neq j} \widetilde{\beta}_{ik} \tag{23}$$

$$\tau^l_{ij} = \widetilde{\eta}_{ij} + \sum_t s^t_{ij} + \sum_{t \neq l} \phi^t_{ij} \tag{24}$$

$$\begin{aligned} \phi^l_{ij} = \min \Big[&0, \tau^l_{jj} + \rho^l_{jj} + \\ &\sum_{k \notin \{i,j\}} \max(0, \tau^l_{kj}) + \sum_{k \neq j} \max(0, \rho^l_{kj}) \Big] \end{aligned} \tag{25}$$

$$\phi^l_{jj} = \rho^l_{jj} + \sum_{k \neq j} \max(0, \tau^l_{kj}) + \sum_{k \neq j} \max(0, \rho^l_{kj}) \,. \tag{26}$$

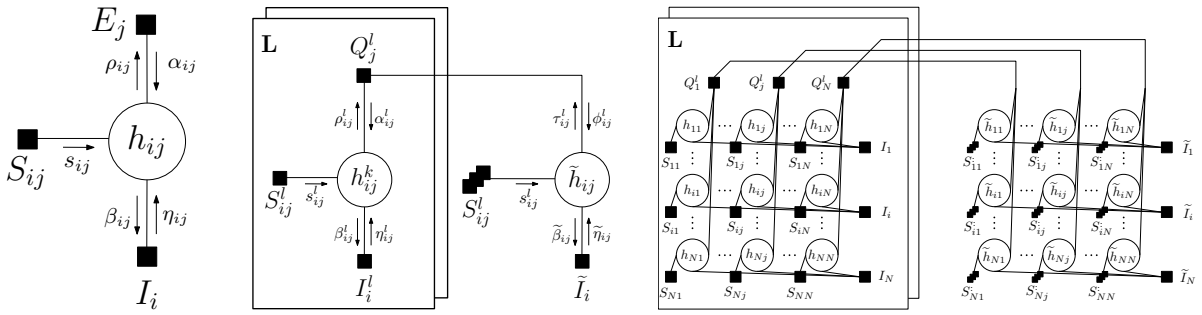
We can then simplify these messages through substitution, yielding:

$$\begin{aligned} \rho^l_{ij} &= s^l_{ij} + \eta^l_{ij} \\ &= s^l_{ij} - \max_{k \neq j} \beta^l_{ik} \\ &= s_{ij} - \max_{k \neq j} \left(s^l_{ik} + \alpha^l_{ik} \right) \end{aligned} \tag{27}$$

$$\begin{aligned} \tau^l_{ij} &= \sum_t s^t_{ij} + \sum_{t \neq l} \phi^t_{ij} + \widetilde{\eta}_{ij} \\ &= \sum_t s^t_{ij} - \sum_{t \neq l} \phi^t_{ij} + \max_{k \neq j} \left(\widetilde{\beta}_{ik} \right) \\ &= \sum_t s^t_{ij} - \sum_{t \neq l} \phi^t_{ij} + \max_{k \neq j} \left(\sum_t s^t_{ij} + \sum_t \phi^t_{ij} \right) . \end{aligned} \tag{28}$$

To compute the actual clustering solution the algorithm proceeds as follows:

1) initialise all messages to 0,
2) for each data layer l compute ρ^l_{ij} and α^l_{ij},
3) compute $\tau^{:}_{ij}$ and $\phi^{:}_{ij}$,
4) iterate step 2) and 3) until convergence.



(a) AP Messages (b) LAP Messages (c) LAP Model

Fig. 1: The messaging structure for a) affinity propagation and b) layered affinity propagation. A graphical model representation of layered affinity propagation is shown in c). As one can see the Q factor node is in a sense an augmented version of the E factor node of the original affinity propagation.

In general affinity propagation is not guaranteed to converge as it is based on loopy belief propagation. However, by damping the message updates convergence problems are not a problem in practice.

The exemplars and assignments are obtained in a similar manner as in standard affinity propagation by selecting the nodes for which the value of $\sum_l \left(\tau_{jj}^l + \phi_{jj}^l\right)$ is positive as exemplars. All other points are assigned to the exemplar that is the most similar.

The runtime of this algorithm is $O((L+1)N^2)$ per iteration where L is the number of data layers and N is the number of data points, as we need to run affinity propagation for every layer and the merging layer. Note that the L data layers can all be run in parallel as they do not influence each other directly. With multi-core CPUs being the norm, it is trivial to distribute the different layers onto the available cores. Thus in practice the actual cost of running a single iteration of the algorithm is not more than two to three times that of affinity propagation.

V. Results

In this section we evaluate the proposed method in two different applications. First we demonstrate that layered affinity propagation (LAP) can be used to perform scene segmentation on Kinect data. Next, we cluster data obtained from a Velodyne and camera pair. We compare our method against affinity propagation using only colour or depth information as well as k-means using a combined colour and depth feature vector. In all experiments the self-similarity values S_{ii} were set to the median of S_{ij} multiplied by a scaling factor between 2 and 10. Convergence of affinity propagation is achieved once the similarity score of the assignments is stable over a number of iterations, 20 in our case. In the experiments convergence was never an issue and typically achieved after 100 to 200 iterations.

A. Kinect

In this experiment we evaluate how well combining different features using LAP performs at segmenting scenes captured with a Kinect. The Kinect provides us with dense depth and colour information with a 1:1 mapping between depth and colour pixels. We approach the segmentation task by first over segmenting the data by extracting super pixels from the image using SLIC [1]. From these super pixels we extract colour and depth features which we subsequently cluster to obtain the final segmentation. In this experiment we use LAB colour histograms and average surface normals as our features. The similarity values required by affinity propagation are computed using the Bhattacharyya distance for colour histograms and angular difference between vectors for the mean surface normals. K-means operates directly

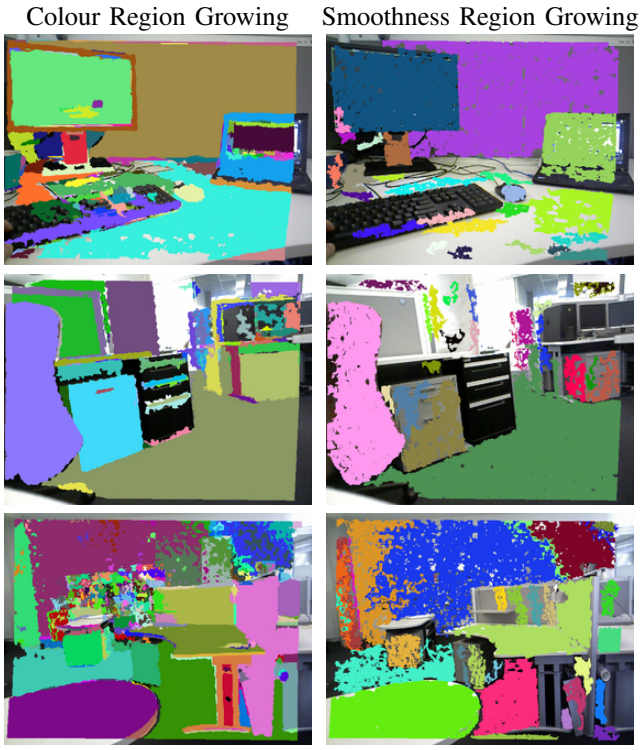


Fig. 4: These images show the segmentation results obtained using region growing based methods. The first column shows results obtained based on colour information. Smoothness of the depth data was used to obtain the results in the second column.

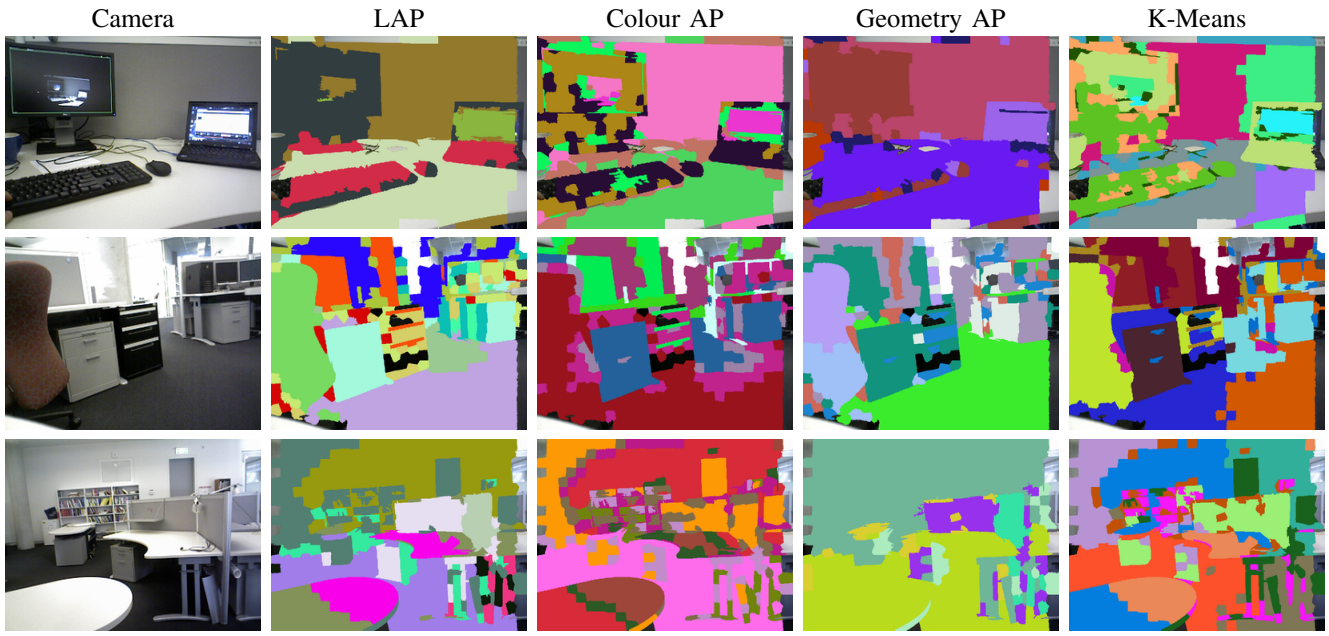


Fig. 2: Exemplary segmentation results, from left to right: raw image, LAP, colour affinity propagation, geometry affinity propagation and k-means. The colours indicate the cluster assignments made.

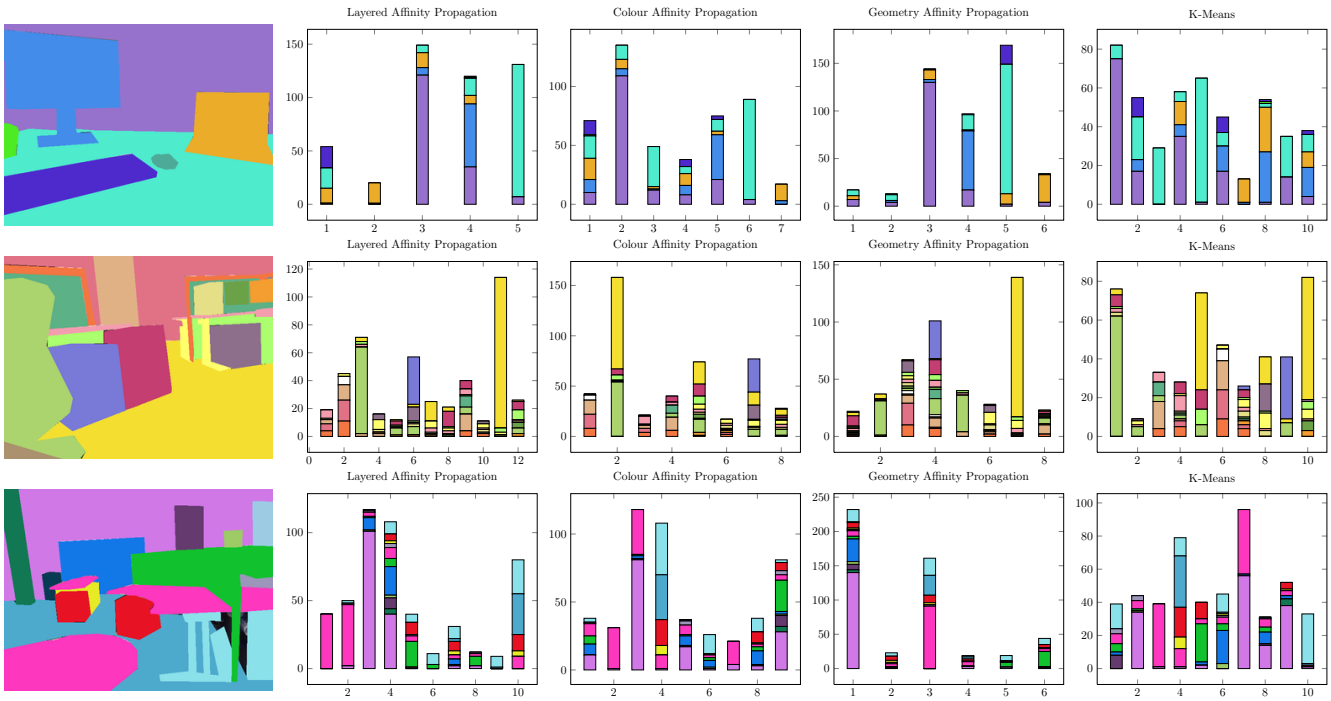


Fig. 3: Clustering statistics for the results shown in Figure 2. The first column contains ground truth labels for each of the scenes while the four plots to the right show the ground truth label distribution in the clusters for each of the four methods used. Each bar represents a single cluster and visualizes the number of data points in the cluster via its height. Homogeneity of a cluster is visualized by the number of different colours in the bar. Completeness can be assessed by the distribution of a single colour over all clusters

on the two histograms and is set to find 10 clusters. For a more direct comparison to K-Means we additionally run affinity propagation with a similarity matrix obtained from the Euclidean distance between the feature histograms. For both K-Means and affinity propagation we choose reasonable parameters but no search for the optimal parameter set is performed, as this wouldn't demonstrate the typical performance of the methods. To see how clustering methods perform in

Method	V-Measure	Homogeneity	Completeness
	Scene 1		
LAP	0.48	0.49	0.47
Colour AP	0.32	0.36	0.29
Geometry AP	0.52	0.53	0.51
Combined AP	0.41	0.51	0.34
K-Means	0.36	0.46	0.29
	Scene 2		
LAP	0.56	0.55	0.57
Colour AP	0.40	0.36	0.46
Geometry AP	0.45	0.41	0.51
Combined AP	0.54	0.57	0.52
K-Means	0.52	0.50	0.54
	Scene 3		
LAP	0.44	0.45	0.44
Colour AP	0.36	0.36	0.36
Geometry AP	0.36	0.30	0.44
Combined AP	0.45	0.47	0.44
K-Means	0.45	0.43	0.48
	Overall		
LAP	0.48 ± 0.09	0.49 ± 0.10	0.47 ± 0.09
Colour AP	0.40 ± 0.09	0.40 ± 0.10	0.40 ± 0.10
Geometry AP	0.43 ± 0.13	0.41 ± 0.14	0.45 ± 0.14
Combined AP	0.43 ± 0.10	0.46 ± 0.12	0.40 ± 0.12
K-Means	0.41 ± 0.06	0.48 ± 0.10	0.37 ± 0.08

TABLE I: V-Measure, homogeneity and completeness scores for the four methods evaluated for the three scenes shown in Figure 2 as well as all the recorded scenes.

comparison to dedicated segmentation approaches we also use colour and smoothness based region growing methods.

We collected several scenes in a typical office environment containing chairs, books, binders, desks, shelves, computers, etc. Typical segmentation results obtained with clustering methods are shown in Figure 2 and Figure 4 for region growing based methods. Each row contains the results obtained for the scene shown in the first column of Figure 2.

One big difference between clustering and region growing methods is that typically clustering methods do not consider spatial closeness and thus may group spatially distant but similar objects together. This can be seen in the second scene with the grey drawers or table surfaces in the third scene. Whether or not this is a desirable property depends on the application. However, adding spatial connectivity information into the clustering system would allow it to exhibit a more region growing like behaviour. Adapting a region growing approach to behave more like clustering methods though is not possible.

Figure 2 shows results obtained with the different methods in three scenes. Looking at colour AP and geometry AP it is obvious that the clusters they find correspond to the features used. However, this is problematic as for example the depth feature is unable to distinguish between the surface of a table and the floor. From the k-means results we see how having both modalities improves the results. However, the choice of the number of clusters to find can have a big impact on the result as it can lead to under or over segmentation. If we now look at results obtained with LAP we can see that clusters adhere well to object boundaries with most larger areas being successfully clustered as a single region when compared to k-means which often ends up splitting them. The numerical evaluation using V-Measure [11] in Table I shows a general trend where LAP outperforms K-Means while colour AP and geometry AP come in last. Comparing K-Means to Combined AP, which uses the Euclidean distance metric, we can see how AP outperforms K-Means even when using the same metric. However, LAP is still able to improve on these results indicating that a more principled way of combining the data is beneficial. Looking at the individual results of the three scenes we can see that geometry AP performs better when the scene is composed of a few large and distinct areas, as is the case in scene one.

For a more detailed look at the results we visualize the size, homogeneity and completeness of each cluster in Figure 3. Each bar represents a single cluster with the distribution of true labels in it, given by the labelled image to the left. The height indicates the cluster's size while the distribution of true labels within a bar represents its homogeneity. The cluster completeness can be assessed by the distribution of a single label over all clusters. These plots show how geometry AP tends to form a few large clusters which capture the major surface normals in the environment which explains the tendency to under segment scenes. For colour AP we can observe how most clusters contain multiple different labels such as the second cluster in the second scene which contains both the chair and floor. This visualizes how colour AP fails to separate areas that appear similar in the colour histograms. The k-means results exhibit a more uniform size then the other methods with a mixture of very homogeneous clusters and mixed clusters. Finally, LAP produces uniform clusters for large areas in the scenes and at times collapses multiple small classes into a single cluster.

Taking a look at the results of the two region growing methods in Figure 4 we see that they produce cleaner results compared to the clustering methods. Still, they each have their own set of drawbacks. The colour based version is prone to oversegmentation if there is no direct connectivity between components which is easily caused through occlusions.The smoothness based version has the same need for connectivity but additionally is much more sensitive to the choice of smoothness threshold. For example a different set of thresholds could obtain better results for the wall in scene three though this would cause other parts to be undersegmented.

B. KITTI Dataset

The KITTI dataset [4] is designed as a vision benchmark but also provides calibrated Velodyne and camera data recorded in urban scenes. This provides us with the same data modalities as the Kinect, depth and colour, but at different densities and no direct correspondence between the modalities. Figure 5 shows the type of coloured point clouds this dataset provides us with. In this experiment we are not clustering these raw point clouds but rather segments extracted from these. To this end we segment the data as follows:

Fig. 5: Visualization of exemplary point clouds of the KITTI dataset coloured using information from the camera images.

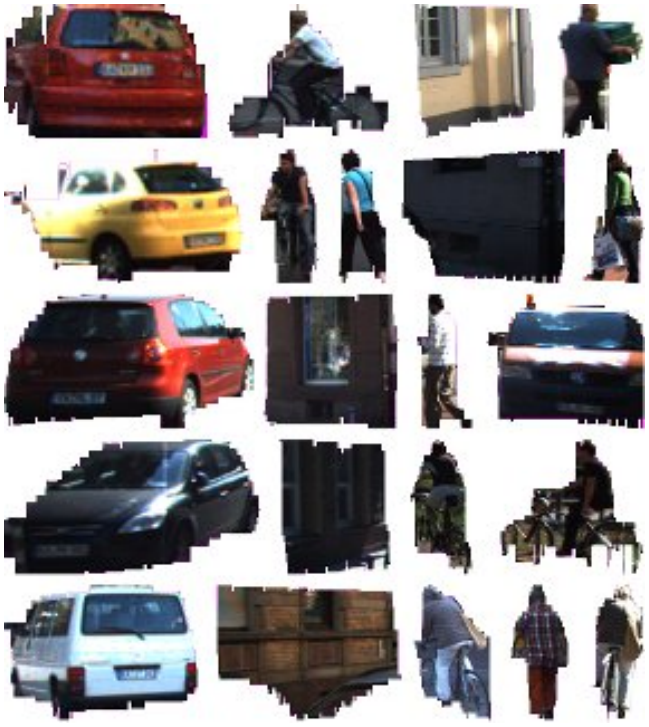


Fig. 6: Exemplary image data corresponding to point cloud segments extracted from the raw Velodyne data. Shown are cars, pedestrians, cyclists and wall segments.

1) remove the ground plane from the point cloud,
2) find segments in the point cloud using Euclidean distance clustering,
3) extract image parts corresponding to the point cloud segment.

This provides us with 3D point clouds with associated colour information. From this data we select segments which occur frequently, i.e. cars, cyclists, pedestrians and wall segments. Examples of segments found from this data are depicted in Figure 6. This shows the variability the data has in orientation, posture, colour and size, both within and between classes. From this collection of segments we choose random subsets to cluster. The features we extract are colour histograms and surface normal histograms. Bhattacharyya distance is used to compute pairwise similarities. We show the average results obtained from 20 runs in Table II. We can

Fig. 7: Examples of successful clustering results obtained with LAP. The individual groups show cars, pedestrians and cyclists respectively clustered together despite their different appearances.

see how combining the two modalities with LAP improves the results. The colour and geometry based affinity propagation methods produce decent results but are outperformed by LAP. K-means on the other hand struggles on this data set. Tweaking the number of clusters could improve the result somewhat but no single value would work for all runs. This reinforces the importance of methods that detect the appropriate number of clusters automatically. Another important observation is the high homogeneity score of LAP which means that using hierarchical methods can easily further improve the results.

To better understand the results we show segments successfully assigned to the same cluster by LAP in Figure 7. This shows how how cars, pedestrians and cyclists are grouped together even though they appear different in colour and posture. While at first this may be counter intuitive we have to remember that affinity propagation optimizes a global cost function and as such is influenced by within cluster similarity as well as inter cluster dissimilarity. For example cars tend to use a single colour and have one or two strong normals, whereas a pedestrian will have multiple main colours and more evenly distributed normals. Thus optimizing both the similarity within a cluster as well as the dissimilarity between clusters the algorithm is capable of finding the solutions shown. In Figure 8 we show an instance where the clustering failed to separate a van from wall segments. Since only the side of the van is visible it is easy to see why these segments were grouped together. While we ideally would like the clustering to provide us with four clusters representing our four classes it is unrealistic to achieve this directly. However, the homogeneous nature of LAP clusters should allow us to use hierarchical methods to improve the results.

VI. Conclusion

In this paper we have presented a novel clustering approach that combines information from multiple sensors in a principled way. The approach allows the user to define features that are appropriate for each sensor individually, without having to worry about how to best combine them. In experiments we have shown that this approach can be used perform segmentation by combining colour and depth

Fig. 8: This shows one of the more common clustering mistakes, the side view of a car being clustered together with wall segments. Since only the planar side of the van is visible and both walls and van have rather uniform colour distributions this type of error is not unsurprising.

Method	V-Measure	Homogeneity	Completeness
LAP	0.41 ± 0.01	0.82 ± 0.02	0.28 ± 0.01
Colour AP	0.35 ± 0.01	0.66 ± 0.02	0.26 ± 0.01
Geometry AP	0.37 ± 0.01	0.67 ± 0.02	0.26 ± 0.01
K-Means	0.14 ± 0.02	0.19 ± 0.02	0.11 ± 0.01

TABLE II: Average V-Measure, homogeneity and completeness scores with standard deviation of 20 clustering runs. LAP has the best overall V-Measure score but also produces much more homogeneous results. This is important as it indicates that further hierarchical processing is likely to further improve the results.

information of a Kinect. In a second experiment we evaluated the performance of the method when clustering point cloud segments with colour information obtained in urban scenes. While the experiments concentrated on depth and colour information nothing prevents the use of data from other sensors, such as accelerometers or hyperspectral cameras for example.

REFERENCES

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[2] R. Bekkerman and J. Jeon. Multi-modal Clustering for Multimedia Collections. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[3] B.J. Frey and D. Dueck. Clustering by Passing Messages Between Data Points. *Science*, 2007.

[4] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] I. Givoni and B. Frey. A Binary Variable Model for Affinity Propagation. *Neural Computation*, 2009.

[6] I. Givoni, C. Chung, and B. Frey. Hierarchical Affinity Propagation. In *Uncertainty in Artificial Intelligence*, 2011.

[7] A. Howard, M. Turmon, L. Matthies, B. Tang, A. Angelova, and E. Mjolsness. Towards Learned Traversability for Robot Navigation: From Underfoot to the Far Field. *Journal of Field Robotics*, 2006.

[8] I. Jebari and D. Filliat. Color and Depth-Based Superpixels for Background and Object Segmentation. *Procedia Engineering*, 2012.

[9] R. Katz, J. Nieto, and E. Nebot. Unsupervised Classification of Dynamic Obstacles in Urban Environments. *Journal of Field Robotics*, 2010.

[10] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 2001.

[11] A. Rosenberg and J. Hirschberg. V-Measure: A conditional entropy-based external cluster evaluation measure. In *Proc. of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007.

[12] J. Schoenberg, A. Nathan, and M. Campbell. Segmentation of Dense Range Information in Complex Urban Scenes. 2010.

[13] J. Sun, J. Rehg, and A. Bobick. Learning for Ground Robot Navigation with Autonomous Data Collection. Technical report, 2005.

[14] R. Triebel, R. Paul, D. Rus, and P. Newman. Parsing Outdoor Scenes from Streamed 3D Laser Data Using Online Clustering and Incremental Belief Updates. In *AAAI Conference on Artificial Intelligence*, 2012.

[15] C. Wang, J. Lai, C. Suen, and J. Zhu. Multi-Exemplar Affinity Propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[16] J. Xiao, J. Wang, P. Tan, and L. Quan. Joint Affinity Propagation for Multiple View Segmentation. In *International Conference on Computer Vision*, 2007.

[17] D. Zhang, C. Lin, S. Chang, and J. Smith. Semantic Video Clustering Across Sources using Bipartite Spectral Clustering. In *IEEE International Conference on Multimedia and Expo*, 2004.