Grid-based DBSCAN for clustering extended objects in radar data

Conference Paper · June 2012

DOI: 10.1109/IVS.2012.6232167

3 authors, including:

Dominik Kellner

13 PUBLICATIONS **126** CITATIONS

Jens Klappstein

Daimler

41 PUBLICATIONS **409** CITATIONS

Grid-Based DBSCAN for Clustering Extended Objects in Radar Data

Dominik Kellner, Jens Klappstein and Klaus Dietmayer

***Abstract*—The online observation using high-resolution radar of a scene containing extended objects imposes new requirements on a robust and fast clustering algorithm. This paper presents an algorithm based on the most cited and common clustering algorithm: DBSCAN [1]. The algorithm is modified to deal with the non-equidistant sampling density and clutter of radar data while maintaining all its prior advantages. Furthermore, it uses varying sampling resolution to perform an optimized separation of objects at the same time it is robust against clutter. The algorithm is independent of difficult to estimate input parameters such as the number or shape of available objects. The algorithm outperforms DBSCAN in terms of speed by using the knowledge of the sampling density of the sensor (increase of app. 40-70%). The algorithm obtains an even better result than DBSCAN by including the Doppler and amplitude information (unitless distance criteria).**

I. Introduction

In the automotive field, high resolution radar is required for roadside detection, lane prediction and classification of objects to make an intelligent interpretation of a traffic scene possible. In these systems, vehicles appear as laterally and longitudinally extended objects caused by reflections from the road surface. In radar systems, the complete bottom side of the car can be seen, allowing an accurate recognition and classification into different classes (car, van and truck). [2]

With high-resolution radar, an extended object results in more than one observation. A clustering algorithm is required to associate the single reflections (observations) with an unknown number of different objects. Otherwise, the following signal processing steps must deal with a large amount of data. E.g., conventional tracking algorithms are optimized for point targets. Dealing with a large amount of observations per object decreases their speed significantly and results in association problems. An additional step is required to sort and merge tracks. [3]

These steps can be avoided by using an adequate clustering algorithm. But most common cluster algorithms are based on an equidistant sampling density (recorded samples per unit distance). The same object results in the same number and distribution of points independent of its position inside the field of view.

Manuscript received January 24, 2012.

D. Kellner is with Departement of Measurement, Control and Microtechnology, University of Ulm, Germany. Contact: dominik.kellner@uni-ulm.de

J. Klappstein is with Daimler AG, Ulm, Germany

K. Dietmayer is with Departement of Measurement, Control and Microtechnology, University of Ulm, Germany

In contrast, the observation of a target in a radar signal is determined by an azimuth angle (θ) and the distance to the sensor (r). Its position in Cartesian coordinates can be calculated using trigonometric functions (Fig1-left). Both variables have a fixed sampling resolution, with r in units of length and θ in units of angle. The result is a non-equidistant sampling in Cartesian coordinates resulting from the trigonometric calculation of the x-y values. Therefore, the minimal azimuth distance between two points increases with range. Due to the fixed sampling resolution, a grid representation consisting of range and azimuth cells (Fig1-right) is easy to apply.

For the mentioned applications in the automotive field, [2] proposes a 77 GHz radar with a resolution of $1°$ in the azimuth direction and 1m in the range direction. This results in a strong variation of the sampling density in Cartesian coordinates, as seen in a simple example (Fig. 2).

The shape of the side of a car (length 5m, width 2m) at a distance of 3 meters causes over 150 possible observations compared to only 9 possible observations at a distance of 80 meters. Because of the fixed metric range resolution, the number of affected range cells is equal while the number of affected azimuth cells differs significantly. For the car at 3m, 59 azimuth cells are affected compared to only 3 cells for a car at 80m.

The CFAR method presented in [5] is used to suppress clutter and remove no-target points. The targets result in the points used for the clustering algorithm.

This paper is outlined as follows: Section II gives an overview of the most common and automotive-specific clustering algorithms and their disadvantages compared to a non-equidistant sampling density. In Section III, the standard

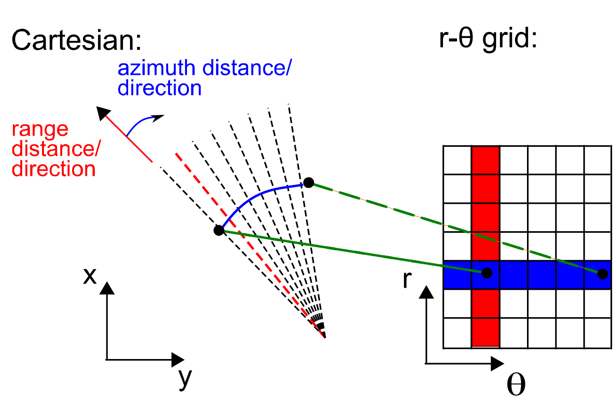

Fig. 1: Data output of the sensor (right) in r- and θ-grid and data transformation in common Cartesian (x-y) coordinates obtained from trigonometric functions

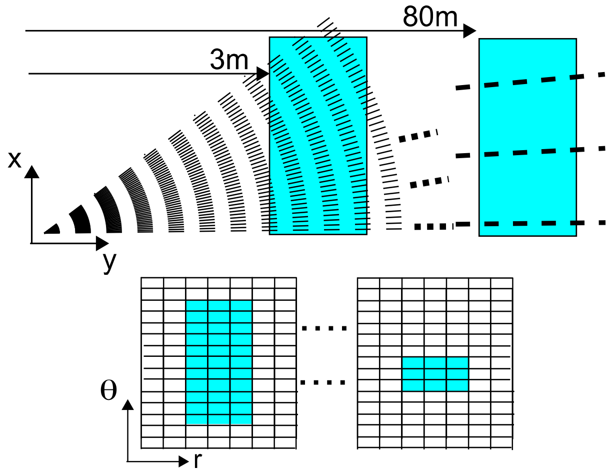

Fig. 2: Difference in number of possible observations for a car with dimensions 5x2m at a distance of 3m and 80m in front of a radar sensor (Δr = 1m, $\Delta\theta$ = 1°) in Cartesian coordinates (top) and in r-θ grid (button)

DBSCAN algorithm is analyzed in terms of clustering high-resolution radar data. The grid-based DBSCAN algorithm is presented in section IV and its advantages compared to DBSCAN are discussed. In Section V, the execution time on real data is compared. Finally, a conclusion is presented in Section VI.

II. Overview Clustering Algorithm

In general, clustering divides the data into different classes (clusters), with data in the same cluster showing a great similarity. The differences for data in different clusters are greater. In this paper, the similarity is based on the spatial information of each point (x-y or r-θ).

There are two main categories of cluster algorithms: hierarchical and partitioned. A good overview of algorithms with regards to radar data is presented in [6]. Only the most common algorithms and their disadvantages in terms of clustering high-resolution radar are discussed here:

Hierarchical algorithms, such as CURE, BIRCH, and CHAMELEON, can basically discover clusters of any shape and size. But the algorithm's complexity regarding space and time is high. Furthermore, the merge and split process needs a constant distance criterion, which is hard to estimate for a non-equidistant sampling density. [7]

Some of the most common algorithms in clustering are the k-mean based methods. They rely on prior knowledge of the number of clusters present and each point in the dataset must be assigned to one of the clusters. In radar datasets, there are an unknown number of extended objects and reflection centers, so this input parameter is hard to estimate [8]. Radar datasets usually contain clutter, so even those points are assigned to one of the clusters, deteriorating the performance of the algorithm. There are density approaches to exclude potential outliers, but they are time-intensive and can have restrictions on the noise distribution. [9]

All methods which assume, for example, an elliptical cluster, e.g. presented in [6], can't be used due to the varying object shapes. Pixel- and segment-based segmentation are not suitable because neither the amplitude nor Doppler value is significant enough to precisely cluster an object. Furthermore, the sampling density might be too small for an edge search [10]. DBSCAN, which uses a fixed density, can't be used due to the high variation in density. Variable density modified DBSCAN algorithms like OPTICS [11] are too time-consuming. SDDC [6], or the region growing approach in [6] handle a non-equidistant density just between features (like x- and y- value). But these algorithms fail if the feature itself has a high density variation.

III. DBSCAN For automotive Radar

A. DBSCAN

The general idea of density-based methods is to continue growing a cluster as long as the density in the neighborhood exceeds some threshold. This density criterion can be described as follows: "for each point of a cluster the neighborhood of a given radius has to contain at least a minimum number of points, i.e. the density in the neighborhood has to exceed some threshold" [1]. All points in a cluster which meet this criterion are marked as core points. A point is marked as a border point if it doesn't meet this criterion but is inside the search area of a core point. All other points are marked as outliers. The procedure of growing clusters and marking points remains unchanged for all algorithms presented. Therefore, only the density criterion is the subject of this paper.

B. DBSCAN on radar data

DBSCAN uses a fixed density threshold for the creation of a cluster. This means, each observation is examined if there are at least k observations within a radial search radius ε.

Applying DBSCAN on a dataset with a non-equidistant sampling density as mentioned in section I (Fig 2.) induces some disadvantages which are shown in Fig.3:

1) *Presence of clutter*
   There is a high sampling density in Cartesian coordinates close to the sensor, caused by the small azimuth distance between possible observations. With this large number of possible observations in the search radius ε, even widely distributed clutter is clustered to objects.
2) *Limited Range*
   Objects which are far away from the sensor aren't clustered because the search radius ε is smaller than the sampling resolution or the number of possible observations is smaller than the amount of required observations (k). In general, the number of possible observations in the search radius decreases with the range. The result is that small objects are only clustered close to the sensor.
3) *Separation resolution*
   The ability to separate two objects in the azimuth

direction varies significantly with the range. Close to the sensor, the search radius includes a large number of azimuth cells, so that a separation of two closely spaced objects is not possible.

4) *Object orientation*
Range and azimuth direction are not treated equally by DBSCAN. For close objects, the number of possible observations in the azimuth direction is significantly higher than in the range direction. Therefore, the algorithm prefers small and long objects in the azimuth direction and discriminates objects in the range direction.

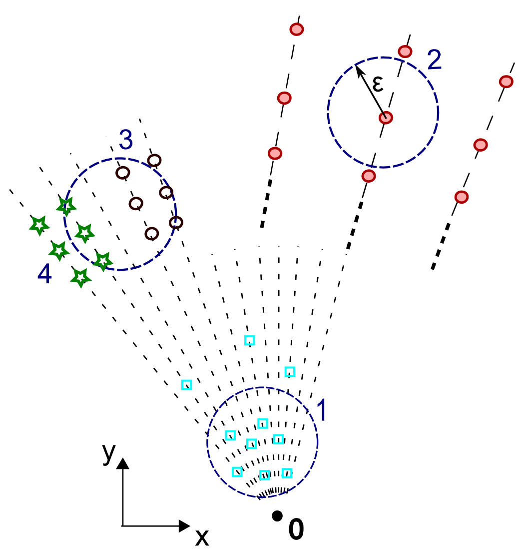

Fig. 3: Effects of original DBSCAN algorithm on two close objects (green, black), one object far away (red) and clutter (cyan)

Each of the mentioned disadvantages can be avoided separately by changing the input parameters k and ε. But they have a strong negative influence on each other. E.g., if the search radius is increased to cluster objects far away (2), more clutter will be clustered together (1) and the separation resolution of two objects decreases (3).

A possible improvement is the introduction of a variable threshold for the number of observations: $k(x,y)$. This parameter must show a high dependency on the number of possible observations inside the search radius. The threshold k has to decrease in the range direction. The results would be that close clutter isn't clustered (1) anymore. Objects far away (2), which have less possible observations in their search radius, are clustered due to the significant smaller threshold k.

Nevertheless, the separation resolution (3) and the disparity of both directions (4) can't be improved. Therefore, it is necessary to introduce spatial-variable parameters for both search radius and number of observations: $\varepsilon(x,y)$ and $k(x,y)$. A simple and fast approach of a spatial density criterion based on the r- and θ-grid (Fig. 1) is presented in the following sections.

IV. Grid Based DBSCAN

A. *Data representation in grid*

DBSCAN needs the x- and y-position of each point to calculate the distance between the examined point, and the rest of the dataset to determine the points inside the search radius ε. The modified grid-based DBSCAN algorithm is based on neighborhood relations and can use the r- and θ-cell-information as a grid (Fig 1-right).

To deal with the non-equidistant sampling density, it is unavoidable to obtain spatial dependant variables for the search area and for the number of required points. The grid-based DBSCAN uses only the ratio c between the radial and angular distance for each point. Since the range distance is always constant, this parameter is sufficient to calculate the spatial sampling density:

$$c_{i,j} = \frac{r_{i,j}}{2\Delta r}(\sin(\theta_{i,j+1} - \theta_{i,j}) + \sin(\theta_{i,j} - \theta_{i,j-1})) \qquad (1)$$

with: i,j index of grid in r / θ - direction
$r_{i,j}$ radial distance
Δr radial resolution (constant)
$\theta_{i,j}$ azimuth angle

This is a sensor-specific ratio, which can be calculated in advance and stored in a look-up-table. During the clustering process, this local ratio is used to determine the optimal number of relevant r- and θ-neighbors (search area), as discussed in the following section.

B. *Determination of spatial density criterion compared to DBSCAN*

As discussed in III) for DBSCAN, the major difference between DBSCAN and grid-based DBSCAN is the density criterion. In this section, the calculation of the density criterion of DBSCAN is presented first and then the modification for grid-based DBSCAN.

DBSCAN starts the process of testing this criterion on all observations by determining the x-y values of all observations (Fig. 4). Due to the fixed sampling, these values are sensor-specific and can be calculated in advance, A position-dependant search radius $\varepsilon(x,y)$ and threshold $k(x,y)$ has to be determined at runtime for each current observation. Then the Euclidean distance to all other observations is calculated. In the final step, the number of observations inside the search radius is compared to the threshold $k(x,y)$.

Compared to the grid-based DBSCAN algorithm (Fig 5.), only the ratio c is calculated in advance for each cell. At runtime, this ratio is used to determine the search area in cells. The area has a constant value in r-direction (constant width h) and depends on c (width $w(c)$) only in θ-direction. The threshold k is calculated as the simple percentage of possible observations inside the search area.

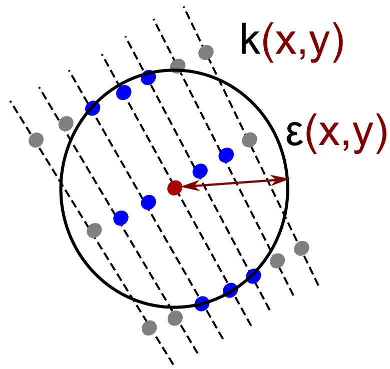

Fig. 4: Calculation of the density criterion using DBSCAN with an adaptive search radius ε and adaptive threshold k in Cartesian coordinates

The advantages of grid-based DBSCAN is a simple comparison in both directions to determine all observations inside the search area. Compared to the calculation of the Euclidean distance between all observations, it is significantly less time-intensive. Furthermore, the data format can be simple integers, compared to float values in DBSCAN. In section V the computation time is examined on real-world data.

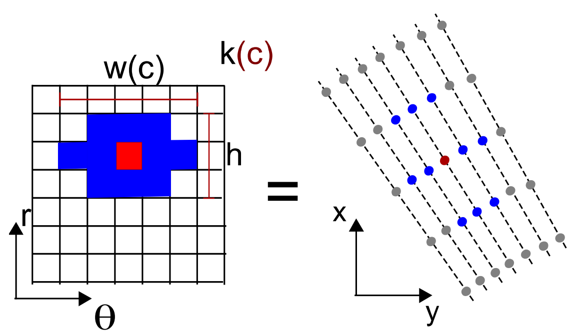

Fig. 5: Calculation of the density criterion using grid-based DBSCAN (left), with the search area specified by the parameters h and w and the adaptive threshold k. The equal representation in Cartesian coordinates is on the right side.

C. *Cluster size and separation of clusters*

The only mandatory input parameter is the ratio of present observations to possible observations for creating a cluster. This parameter depends strongly on the sensor and the desired cluster size.

If no other input parameter is set, the search area is equal to a constant search radius in DBSCAN. Therefore, the second input parameter f (default 1) adjusts the search distance in θ-direction. The search distance in θ-direction can be calculated as follows:

$$w_{i,j}(c) = \frac{1}{f \cdot c_{i,j}} \tag{2}$$

For example, using the factor $f = 2$ results in a circular search area for an equal r- and θ-distance. If the θ-distance gets smaller than the r-distance, the search area becomes an ellipse which gets narrower and narrower for points closer to the sensor. The result is an improved separation of objects in the θ-direction. An example for $f = 2$ is shown in Fig 6.

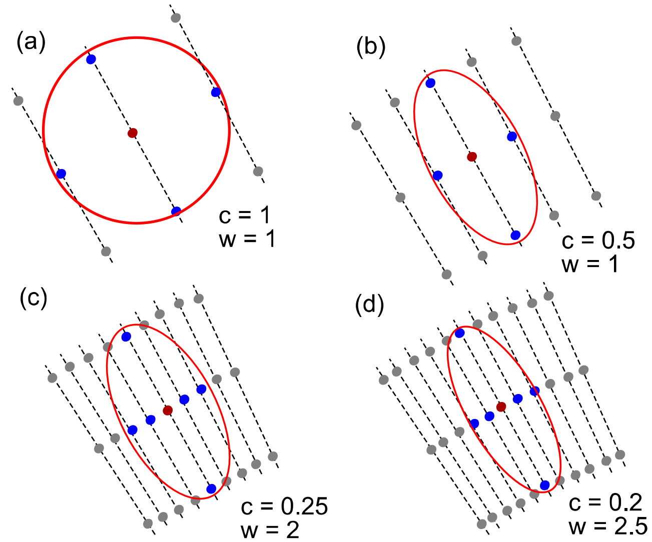

Fig. 6: Influence of different sampling density in the θ-direction of the search area (expressed by w – number of azimuth cells in each direction) of grid-based DBSCAN ($f = 2$). If the sampling resolution is equal in both directions (r and θ), the result is a constant search radius (a). If the point is closer to the sensor, the θ-distance and thus ratio c decrease (b-d).

Typically, the search area in the r-direction includes one point in each direction. This is the minimum value to cluster extended objects. A larger value is normally not applied since the r-distance is quite large. But with parameter g, more than one possible observation in the r-direction can be taken into account. Parameter g adjusts the number of examined r-cells. Then the search distance in the θ-direction can be calculated as follows:

$$w_{i,j}(c) = \frac{g}{f \cdot c_{i,j}} \tag{3}$$

D. *Advantages of DBSCAN in terms of the clustering result*

Regarding the problems mentioned in III-A with respect to the modifications of the density criterion in the grid-based version (Fig 7):

1) *Presence of clutter*
   Clutter close to the senor isn't clustered due to a percentage-based threshold and a small search area in the azimuth direction.
2) *Limited Range*
   The search area always contains at least one cell on each side and in each direction. Furthermore, the number of required observations is a percentage-based threshold and therefore depends on the possible observations in this area.
3) *Separation resolution*

A main advantage is that the search area is adjusted to the local sampling density. This results in a variable separation distance of two objects. For objects close to the sensor, the separation ability (in the azimuth direction) of the modified DBSCAN can be significant higher compared to far objects. This ability depends on the input parameter *f*.

4) *Object orientation*
The search distance in *r*- and θ-direction is independent. The ratio of the search distances depends on the local resolution at the examined point and on the parameter *f*. This means that the points inside the search can be determined independently for θ- and *r*-direction. This results in high execution efficiency.

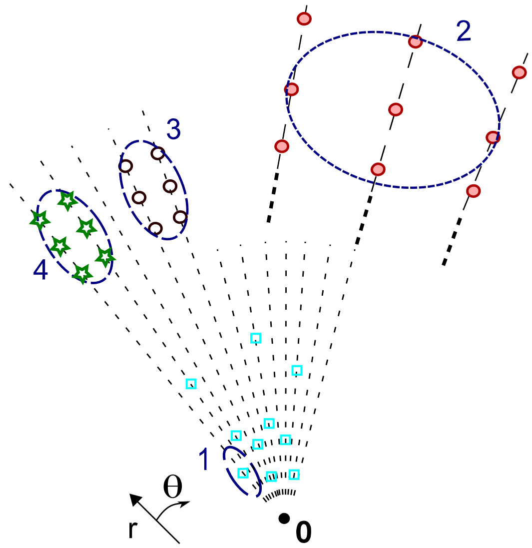

Fig. 7: Effects of the grid-based DBSCAN algorithm on two close objects (green, black), one object far away (red) and noise (cyan)

V. EXPERIMENTAL RESULTS

A. Clustering results on real data

A real data set is used to show the different cluster results of DBSCAN (Fig. 8) and grid-based DBSCAN (Fig. 9). The dataset contains clutter (1), one pedestrian (2), three vehicles (3-5) and a crash barrier (6). The data set is chosen to demonstrate the disadvantages of DBSCAN clustering of non-equidistant data.

The DBSCAN results (Fig. 8) show that the clutter close to the sensor (1) has clustered together. The pedestrian (2) next to the car (3) can not be separated and the two form a cluster. The reason for this is that minimal distance of both objects is app. 0.8m, whereas the search distance is larger ($\varepsilon = 1$m). A car (5) heading towards the sensor results in an object orientation in the *r*-direction. DBSCAN is not able to cluster the car since its search radius covers more possible observations in the azimuth than in the radial direction. The furthest object (6) is not clustered, due to the small number

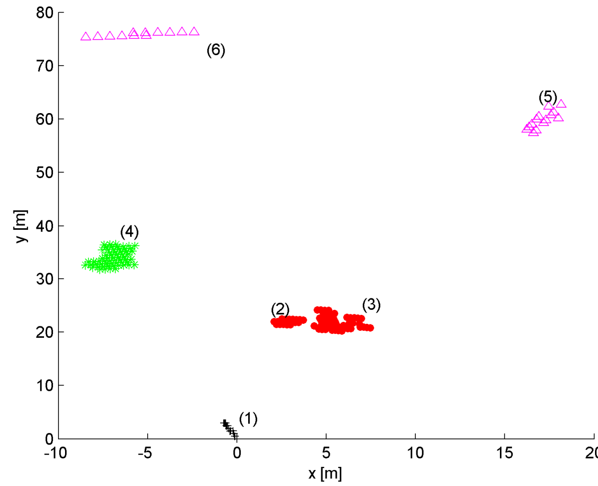

Fig. 8: Clustering results with original DBSCAN (ε = 1m, k = 6) showing outliers (pink triangle) and 3 clusters (red, green, black) for clutter (1), a pedestrian (2) next to a car (3), two other vehicles (4-5) and a barrier (6)

of possible observation compared to the fixed threshold of DBSCAN.

Compared to grid-based DBSCAN (Fig. 9) the clutter (1) is marked as outliers. The pedestrian (2) next to the car (3) can be separated. The ratio *c* in this point is 0.23, so the search distance can be calculated using equation (2), which results in a search distance of 2 cells (= 0.4m) in each azimuth direction. Grid-based DBSCAN is able to cluster the car (5) and the barrier (6) properly. Since the search area contains only one cell in each direction, even objects in radial direction (5) and thin objects can be clustered (6). Furthermore, the search radius always contains the neighboring points and the threshold depends on the possible observations in the search area.

The execution time of grid-based DBSCAN using an ordinarily personal computer decreases by 43%. A detailed

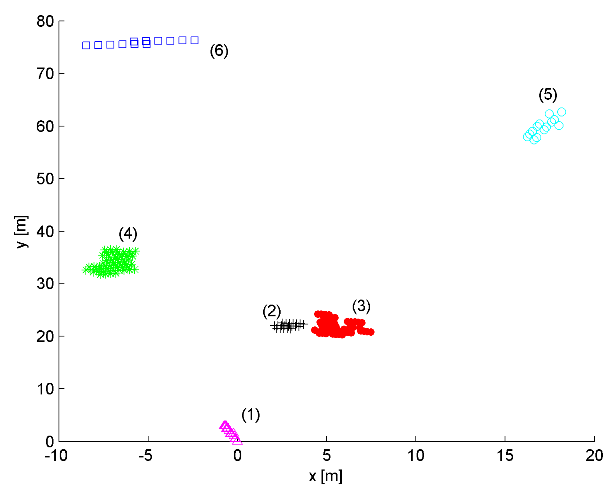

Fig. 9: Clustering results for grid-based DBSCAN (f = 2) showing outliers (pink triangle) and 5 clusters (red, green, black, blue, cyan) for clutter (1), a pedestrian (2) next to a car (3), two other vehicles (4-5) and a barrier (6)

analysis of the execution time is presented in the next section.

B. Execution Efficiency

In this section the runtime of both algorithms is compared using real data. To obtain a significant test, the parameters of the grid based DBSCAN algorithm are chosen to have the same search area as DBSCAN ($f = 1$, $g = 1$). Further, a part of the data is chosen that does not contain front clutter and has a limited range distance, so that all algorithms have the same cluster result. The data shows different objects recorded with our image radar and contains 125, 250 and 500 points. The only difference between DBSCAN and grid-based DBSAN is another calculation of the density criterion, the rest of the algorithm is identical. The third algorithm is a speed optimized version. Instead of calculating the distance between all points, it has a look-up-table representing the grid where each point is registered. With a simple look-up of the search area, the algorithm can determine the indices of all points in the search area. The disadvantage is that for a large database, the look up table increase its size by a factor $O(n^2)$, whereas the other algorithm increases by only a factor O(n).

The corresponding execution time on an ordinary personal computer in Matlab was determined for the three data sets. The grid-based DBSCAN has a significant decrease in execution time compared to the original DBSCAN algorithm (125 points: -60%, 250: -44%, 500: -58%). The fast implementation of the modified algorithm decreases the execution time for large datasets even more (125: -56%, 250: -61%, 500: -69%). The results are shown in Fig. 10.

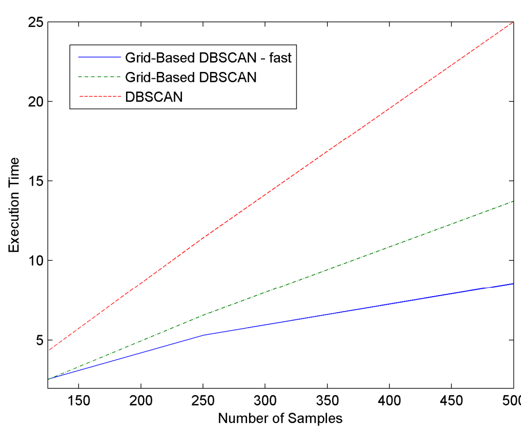

Fig. 10: Execution time for DBSCAN, grid-based DBSCAN and a fast implementation of grid-based DBSCAN, normalized on grid-based DBSCAN for 125 points

VI. Conclusion

This paper presents a density-based algorithm to cluster high-resolution radar data. It not only outperforms the classic DBSCAN algorithm in terms of execution time with app. 40-70%, but increases the separation resolution of two objects. In addition, it is robust against clutter and a non-equidistant sampling density. Especially in high-resolution radar systems, the sampling density is highly non-equidistant and using common cluster algorithms results in a number of disadvantages. With its optional input parameter, the improved clustering is flexible and can be adjusted not only for the sensor but also for the desired cluster size or separation resolution of two close objects.

Since a unitless distance criterion is used, the algorithm can be enhanced by the velocity or amplitude information. A constant parameter, similar to g for the r-direction, has to be introduced to adjust the search area for this new feature. Another possibility is to fill the r-θ-grid with the amplitude values. Then the density criterion could be a comparison of the mean amplitude in the search area to an amplitude threshold.

References

[1] M. Ester, H.-P. Kriegel, J Sander, and X. Xu, "A density-based algorithm for discovering cluster in large spatial databases with noise", Proc. 1996 Int. Conf. Knowledge Discovery and Data Mining. Portland, USA. Aug. 1996, pp.226-231.

[2] R. Schneider, and J. Wengen, "High resolution radar for automobile applications", Advances in Radio Science 2003, Volume 1, 2003 pp. 105-111.

[3] V. Leonhardt, G. Wanielik, and S. Kälberer "A region-growing based clustering approach for extended object tracking", Radar Conference IEEE 2010, Washington DC, USA., May 2010

[4] R. Schneider, "Automotive Radar – Status and Perspectives", Compound Semiconductor Integrated Circuit Symposium, 2005. CSIC '05. IEEE, Nov 2005.

[5] H. Rohling, "Radar CFAR Thresholding in Clutter and Multiple Target Situations", IEEE Transactions on Aerospace and Electronic Systems, July 1983.

[6] F. Pauling, M. Bosse, and R. Zlot, "Automatic Segmentation of 3D Laser Point Clouds by Ellipsoidal Region Growing", Australasian conference on robotics and automation (ACRA), Sydney, Autralia, December 2-4, 2009.

[7] K. Qin, M. Xu, Y. Du, and S. Yue, "Cloud Model and hierarchical clustering based spatial data mining method and application", The international archives of the Photogrammerty, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2, Beijing 2008.

[8] D. Comaniciu, and P. Meer, "Mean Shift: A robust approach toward feature space analysis", IEEE Transactions on pattern analysis and machine intelligence, vol 24. no. 5, May 2005.

[9] D. Rajesh, "Characterization and detection of noise in clustering", Pattern Recognition Letters 12, Nov 1991.

[10] M. Ankerst, M. Breunig, H-P. Kriegel, and J. Sander "OPTICS: Ordering points to identify the clustering structure", Proc. ACM SIGMOD'99 Int. Conf. on Management of Data, Philadelphi PA, USA., 1999.

[11] Z. Li, and X. Wang, "High resolution radar data fusion based on clustering algorithm", Radar Conference IEEE 2010, Washington DC, USA, May 2010.