Extracting Path Graphs from Vehicle Trajectories

Fritz Ulbrich[1], Simon Rotter[1], Daniel Goehring[1] and Raúl Rojas[1]

***Abstract*— In this paper we present an approach for building a graph of drivable paths from the reconstructed trajectories of vehicles detected by lidar and radar sensors mounted in an autonomous car. The perceived objects are tracked, and their trajectories are merged, clustered and labeled with meta information. A graph of the underlying road infrastructure can be generated with this information. We report on the results of testing the validity and accuracy of the method. The generated path graph can be used either to update high precision maps or for generating local temporary maps, both of them useful for autonomous driving.**

I. INTRODUCTION

Most current path planning algorithms for autonomous vehicles are based on prerecorded maps [3]. In general, obtaining a safely drivable path through the environment consists of following the predefined routes in such maps [16]. Interaction with dynamic obstacles is implemented in a low level controller [6].

A map based approach offers many advantages for autonomous driving. The car can plan ahead and use, for example, the shortest path through the road network. The velocity in each segment can be adapted optimally to the known geometry of the road ahead. On the other hand, there is a major disadvantage in map-based approaches. Although road maps are available for most parts of the world, in many cases they are not accurate enough or are outdated.

Map data can become obsolete very fast: Roads are closed down due to construction works, while new roads can be unfinished. Furthermore, there are streets where the number or the direction of drivable lanes changes based on time, weather or volume of traffic. Such frequent challenges are frequently met using dynamic maps, which can be distributed to cars through vehicle-to-infrastructure communication technology.

Another fundamental problem is, that almost all maps available today lack the necessary level of detail. A very detailed map is needed for autonomous driving – it can be at least very helpful. For safe and efficient path planning it is essential to know details like the number and width of available lanes, or the exact geometry of an intersection.

The approach described in this paper can be used to address those shortcomings. A local path graph, representing the drivable trajectories in the immediate surroundings of the ego vehicle, is generated automatically from lidar and radar measurements. The underlying road geometry can be easily extracted from such a graph and be used as a temporary local map, or for updating dynamic maps.

II. RELATED WORK

A lot of research has been performed on automatic or semi-automatic generation of static or dynamic maps using environment sensors of vehicles. More recent approaches rely on multiple scans of the same area with low-cost sensors [15], while others use existing coarse maps as a basis for generating accurate lane-level maps [9]. Research effort has also been put on more specific temporary scenarios, such as road construction sites [17]. All of these approaches rely heavily on the detection of static features, in particular lane markings and other road boundaries. While this is reasonable in an environment where drivers are using the contemplated lanes, the approach presented in this paper is usable in scenarios where the geometry of the road not necessarily defines the lanes actually used by the road users (e.g. due to weather or traffic conditions).

In cases where lane markings are not available - either since they do not exist or because they are temporarily invisible - autonomous vehicles can orient themselves based on convoy or platooning behavior of other traffic participants, which form temporary lanes. Recent work in this field focuses on explicit intervehicle communication [4]. Other approaches form convoys with distributed cooperative control over several lanes and are aiming to incorporate human drivers through maneuvering instructions on their HMI [10]. In contrast, we rely only on observation, i.e. implicit communication, between traffic participants, which is feasible in a near-future mixed environment of autonomous and manually driven vehicles.

Examples for reusing the tracks of other individuals for navigation can also be found in nature, most prominently in ant colonies, adapted to mark the path to food sources through the deposition of pheromones [8]. The principle of "digital pheromones" has been also used in a robotics context, e.g. for UAVs [12]. The word "stigmergy" is used to describe such a behavior, i.e., how agents can communicate and coordinate implicitly through their actions and the resulting changes in the environment [5]. Corne et al. emphasize in [2], that this kind of communication is indirect and compare its fundamental "stigmergy structure" with a notepad which is used by the swarm members to leave cues.

The idea of using the tracks provided by traffic participants to extract further information, like region detection [7] or driving directions [18], seems natural.

However, tracks generated by GPS receivers installed in many vehicles lack the required accuracy to distinguish the different lanes in a single road. Therefore, the autonomous car "MadeInGermany" (see Fig. 1), instrumented with a

[1] DCMLR, Computer Science Institute, Freie Universität Berlin, Germany, {fritz.ulbrich | simon.rotter | daniel.goehring | raul.rojas}@fu-berlin.de

highly accurate GPS navigation unit, was used for data collection for the experiments described in this paper. High precision navigation systems, as well as several lidar and radar sensors allow the tracking of surrounding traffic participants with an accuracy in the range of centimeters. More details about the AutoNOMOS project can be found in [11].

Fig. 1. The autonomous car “MadeInGermany” at a roundabout in Berlin

III. EXTRACTING DRIVABLE PATHS

The basic idea of the approach presented in this paper is to extract a graph of drivable paths from the trajectories of nearby vehicles perceived by lidar and radar sensors. Fig. 2 shows on overview of the extraction process. The individual steps of the process are explained in what follows.

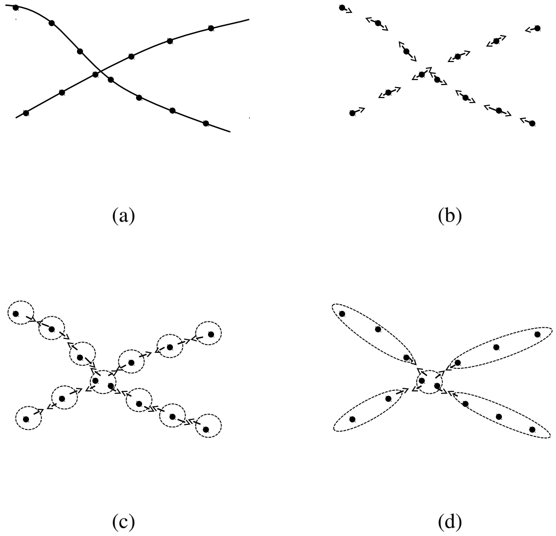


Fig. 2. Overview of the extraction process: (a) The trajectories of surrounding vehicles detected by sensor fusion modules are represented as a list of waypoints. (b) The waypoints have information on their predecessors and successors. (c) Close-by waypoints are merged. Neighborship is preserved. (d) The merged waypoints are clustered according to the ID of the objects involved. Connections between the clusters are constructed from the existing neighborship between the merged waypoints (adapted from [13]).

A. Detecting Surrounding Vehicles

The first step in the processing pipeline of the approach described in this paper is the detection of the surrounding traffic participants, that is, mostly moving cars. For the experiments in IV a sensor setup as depicted in Fig. 3 was used.

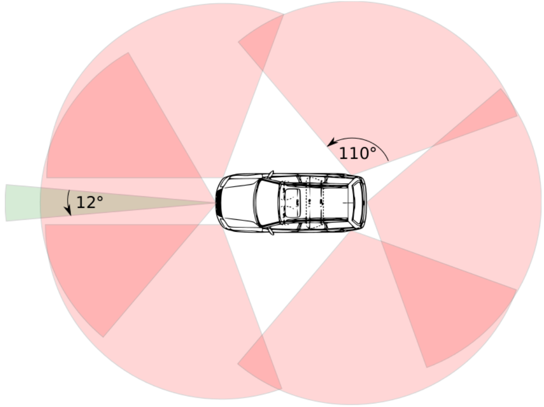


Fig. 3. Sensor configuration used for the experiments: six single Ibeo Lux lidar sensors (red) and one TRW radar (green) [14].

The position and velocity of objects around the ego vehicle are tracked with a Kalman filter. Based on the position of the objects detected by the lidar sensors, the velocities detected by the radar sensor are used to improve the overall accuracy of object recognition and tracking. The update frequency of the sensor fusion modules is 12.5 Hz. The object detection and tracking based on lidar/radar sensor fusion is explained in detail in [14].

The global coordinates of the objects detected by the sensors are calculated using the information provided by an Applanix POS LV 510 Positioning System, installed in our vehicle. The position accuracy provided by the system at 200 Hz is under 0.3 meters when using real time GPS correction data.

The result of this detection process is a list of objects with their corresponding unique IDs used for the tracking process. Each object has an associated velocity, size and boundary box. Furthermore, a list of the waypoints representing the trajectory of the object (compare Fig. 4) is generated by our software. Detected objects are classified into static, dynamic or unknown, based on their observed velocities. This is useful for the extraction of the road geometry, because only moving objects can provide information about drivable paths.

The objects received from the sensor processing modules are filtered according to their state of movement, size and classification. For the next processing step the following information is stored for each object:

- The ID of the object;
- The trajectory represented as list of waypoints;
- The predecessor and successor of each waypoint;
- The velocity of the object at the respective waypoints.

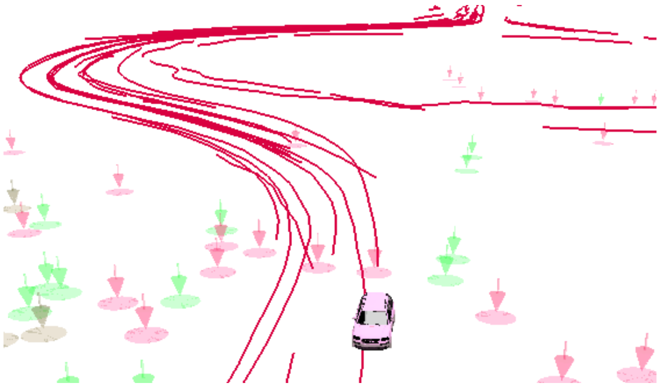


Fig. 4. Detected objects and their trajectories in a roundabout [13]. Dynamic objects are shown in red, static objects are shown in green, and unknown objects are shown in gray. The filtered trajectories of objects that are candidates for being vehicles are shown in red.

While the IDs and the neighborship relations of the waypoints are essential for the clustering process described below, the corresponding velocities constitute additional information which is useful, for example, to provide a recommendation about the optimal or maximal speed limits in later applications.

B. Merging Waypoints

The second step during the processing of the perceived objects is to merge neighboring waypoints of the trajectories - possibly of different objects - to find representative merged waypoints for drivable paths. This has the advantage of reducing the overall number of waypoints used in the clustering described in the next section. We adapted an algorithm proposed by Guo et al. in [7] to this task. The six steps of the method are:

1) Start from any waypoint s and let $C = \emptyset$ be the set of representatives.
2) Find all the waypoints within a distance d to s that are not represented by any existing representatives in C. Calculate the centroid c of such points (including s).
3) Find the waypoints p_i within distance d to c. For each point p_i:
   - If p_i is not represented yet, assign p_i to c (i.e. p_i will be represented by c).
   - If p_i is already assigned to another representative q but p_i is closer to c, re-assign p_i to c (i.e. p_i will be represented by c instead of q).
4) Choose the next point s, which is a neighbor to any point in p_i and is not yet represented. If all neighbors of p_i are represented, then randomly choose s from the remaining not-represented points;
5) Repeat steps (2)-(4) until all waypoints are represented
6) Move each representative to the center (centroid) of the points that it represents.

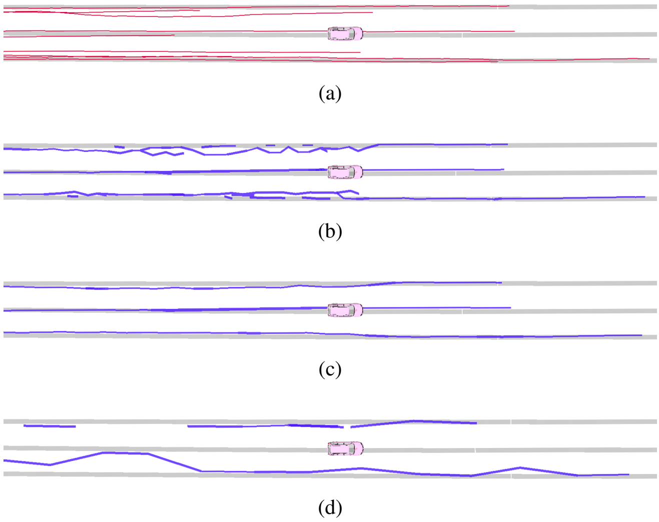
The distance d used in this algorithm can be chosen according to the expected width of the lanes. It is important to preserve the IDs of the objects from which the waypoints are originating, as well as the information about the neighborship relations of each waypoint. This information is used to apply clustering in the next processing step. Therefore, a merged waypoint can have more than one predecessor and more than one successor. Also the velocities associated with each waypoint are preserved with a mimimum, maximum, and average velocity for each merged waypoint.

C. Clustering Merged Waypoints

Finally, the merged waypoints are clustered according to their neigborship relations and the IDs of the objects involved. A merged waypoint is a candidate to be assigned to a cluster if it is a successor or predecessor of one of the waypoints already in the cluster. The other criterion for the assignment to a certain cluster is whether the IDs of the objects contributing to the neighboring merged waypoint match. Only if those two conditions are met, the waypoint is added to an existing cluster - otherwise a new cluster containing the waypoint is created. The waypoints of each cluster created in such a way all share the same set of IDs. Connections between the clusters can be established using the neighborship relations of the first and last point in each cluster (compare Fig. 2). A graph, which represents the drivable paths, can be created by traversing those connected clusters.

IV. EXPERIMENTAL RESULTS

We use the ARND map format developed in the AutoNOMOS project [11] to evaluate the validity and accuracy of the path graph extracted from the perceived objects. The expected trajectory of vehicles driving in a specific lane of the road network is represented as an Akima spline [1]. It is created semi-manually by averaging the data of several recorded test drives. Methods are provided to calculate the



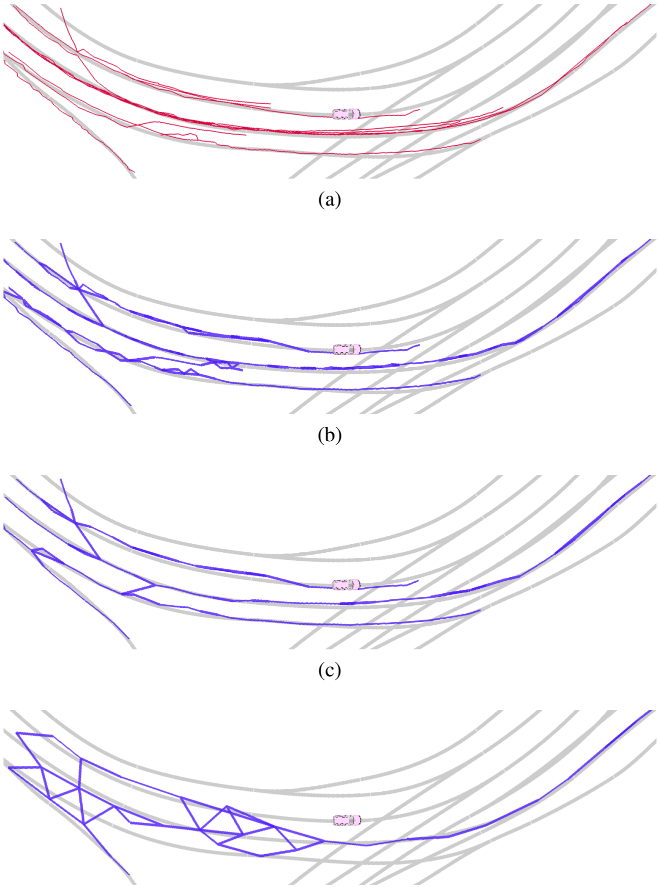
(a)

(b)

(c)

(d)

Fig. 5. Scenario A: city highway A100, Berlin. Expected trajectories extracted from the ARND map are shown in grey for comparison. (a) The trajectories of surrounding vehicles. (b) The generated path graph with a merge distance of 1 meter. (c) The generated path graph with a merge distance of 2 meters. (b) The generated path graph with a merge distance of 5 meters.



(a)

(b)

(c)

(d)

Fig. 6. Scenario B: Siegessäule roundabout, Berlin. Expected trajectories extracted from the ARND map are shown in gray for comparison. (a) The trajectories of surrounding vehicles. (b) The generated path graph with a merge distance of 1 meter. (c) The generated path graph with a merge distance of 2 meters. (b) The generated path graph with a merge distance of 5 meters. Lane changes of the detected vehicles produce some artifacts which could be post-processed, but which also provide useful information.

distance to the nearest lane spline - given a position - in an efficient way. Details on the format and map creation can be found in [3].

The experiments presented in this paper involve two scenarios: (A) a city highway (A100, Berlin), and (B) an inner city roundabout (Siegessäule, Berlin). Compare Fig. 5 and Fig. 6 respectively for a visualization of the road layout and generated graphs.

Scenario A comprises three lanes – the ego vehicle is driving in the center lane. Another eight vehicles were tracked within a radius of 50 meters around the ego vehicle. There were no lane changes for the observed vehicles. The accuracy of the localization of the ego vehicle was better than 0.1 meters. Scenario B comprises four to five lanes in a roundabout with several lanes forking out. The ego vehicle is driving in the second lane from the inside. Ten other vehicles were tracked in a radius of 50 meters around the ego vehicle.

Table I summarizes the results of the experiments, where n_v denotes the count of observed vehicles, d the merge distance used, μ_{dist} the mean of the distance of the merged waypoints to the nearest expected trajectories from the ARND map, σ_{dist} the standard deviation of the distance of the merged waypoints to the nearest expected trajectories, n_{mwp} the overall count of merged waypoints, and $n_{mwp/c}$ the count of merged waypoints per cluster. In the scenario B* clusters whose first and last merged waypoint do not share the same nearest expected trajectory were excluded. This provides results without considering the lane changes of some of the observed vehicles. Taking into account that the largest source of error is that the other vehicles are actually not driving exactly on the expected trajectories, we can neglect the measurement errors (between 0.072 to 0.094 meters for object detection [14] and less than 0.1 meters for localization). The most relevant parameter here is the distance d used in the merging process. As the results and visualization of the experiments clearly show, a value set to approximately half of the actual lane width gives the best results.

Another relevant aspect is the influence of lane changes of observed vehicles. As can be seen in Table I there is a difference of about 15% in the accuracy of scenario B and B* (where lane changing clusters were excluded). Nonetheless those lane changes can provide hints for autonomous path planning or can be excluded from the graph for map updates (compare [13]).

TABLE I
EXPERIMENTAL RESULTS

Scenario	n_v	d	μ_{dist}	σ_{dist}	n_{mwp}	$n_{mwp/c}$
A	8	1 m	0.3963 m	0.3252 m	207	2.7970
		2 m	0.3355 m	0.2072 m	68	7.4756
		5 m	0.6256 m	0.4103 m	38	1.9972
B	10	1 m	0.4443 m	0.3216 m	276	2.9360
		2 m	0.4138 m	0.2970 m	122	3.3870
		5 m	0.6570 m	0.4807 m	54	2.0753
B*	10	2 m	0.3528 m	0.2380 m	114	3.6774

V. CONCLUSIONS

The approach presented in this chapter provided the expected results. Our approach was implemented and tested for the generation of temporary local maps for path planning for an autonomous vehicle in real traffic [13]. The path graph and the velocity information stored for each merged waypoint were used to calculate a desired trajectory and velocity for the subsequent control modules, which generate the commands for gas, brake, and steering.

The approach can be adapted to specific criteria, e.g. by including more information in the merged waypoint structure, such as the type or size of the detected object. A possible improvement would be to consider the direction of the trajectories in the merging process. Doing so, the trajectories can be merged over the whole estimated lane width while still preserving a smaller distance between the point in longitudinal direction. Furthermore, the approach profits greatly from more vehicles being observed. The clusters tend to converge to the center of the lane, when there are more than one trajectories per lane - as can be seen clearly on the left side of Fig. 5 and Fig. 6 respectively. Therefore a major improvement would be the inclusion of more trajectories and observed vehicles respectively. The experiments presented in this paper were carried out on data perceived in a single drive-through. More vehicles could be used to collect the data, possibly commercial vehicles connected by vehicle-to-infrastructure communication.

Another obvious future application would be the creation or update of static maps. Although approaches which rely on static landmarks, such as lane markings, are certainly more accurate with respect to the exact geometry of the lanes, our approach is also applicable in situations where those landmarks are not available. Furthermore it could be detected where the geometry of the road contradicts the behaviour of the human drivers, i.e. where lane markings are often ignored, and road construction planning could be improved.

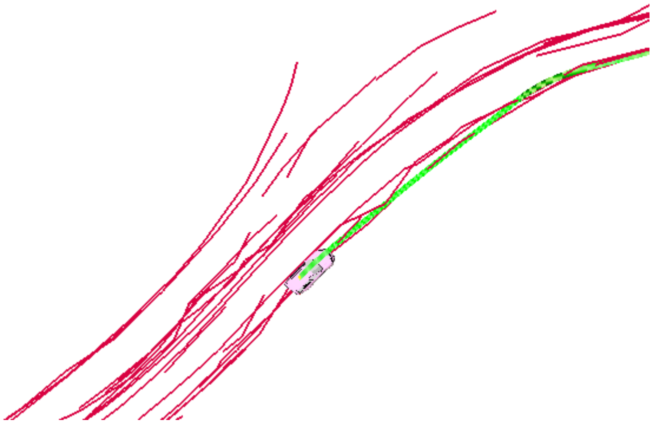


Fig. 7. Pathplanning with a temporary local map [13]. The generated trajectory is shown in green, the underlying trajectories are shown in red.

Acknowledgment

This work was supported by the Deutsche Forschungsgemeinschaft (DFG), SPP 1835 - Kooperativ interagierende Automobile.

References

[1] H. Akima. A new method of interpolation and smooth curve fitting based on local procedures. *Journal of the ACM (JACM)*, 17(4):589–602, 1970.

[2] D. W. Corne, A. Reynolds, and E. Bonabeau. Swarm intelligence. In G. Rozenberg, T. Bäck, and J. N. Kok, editors, *Handbook of Natural Computing*, pages 1599–1622. Springer Berlin Heidelberg, 2012.

[3] P. Czerwionka. A three dimensional map format for autonomous vehicles. Master's thesis, Freie Universität Berlin, 2014.

[4] P. Fernandes and U. Nunes. Multiplatooning leaders positioning and cooperative behavior algorithms of communicant automated vehicles for high traffic capacity. *IEEE Transactions on Intelligent Transportation Systems*, 16(3):1172–1187, June 2015.

[5] S. Garnier, J. Gautrais, and G. Theraulaz. The biological principles of swarm intelligence. *Swarm Intelligence*, 1(1):3–31, 2007.

[6] D. Göhring. Controller architecture for the autonomous cars: Madeingermany and e-instein. Technical report, Freie Universität Berlin, 2012.

[7] D. Guo, S. Liu, and H. Jin. A graph-based approach to vehicle trajectory analysis. *Journal of Location Based Services*, 4(3-4):183–199, 2010.

[8] B. Hölldobler and E. O. Wilson. *The ants*. Harvard University Press, 1990.

[9] A. Joshi and M. R. James. Generation of accurate lane-level maps from coarse prior maps and lidar. *IEEE Intelligent Transportation Systems Magazine*, 7(1):19–29, Spring 2015.

[10] A. Marjovi, M. Vasic, J. Lemaitre, and A. Martinoli. Distributed graph-based convoy control for networked intelligent vehicles. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 138–143, June 2015.

[11] AutoNOMOS Labs. *AutoNOMOS Project*. AutoNOMOS, 2016. `http://www.autonomos-labs.de/`.

[12] D. H. V. D. Parunak, L. M. Purcell, and M. R. OConnell. Digital pheromones for autonomous coordination of swarming uav's. *AIAA's 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, S 20-23 May 2002, Portsmouth, Virginia*, 1001, 2002.

[13] S. Rotter. Swarm behaviour for path planning. Master's thesis, Freie Universität Berlin, 2014.

[14] M. Schnürmacher, D. Göhring, M. Wang, and T. Ganjineh. High level sensor data fusion of radar and lidar for car-following on highways. In *Recent Advances in Robotics and Automation*, pages 217–230. Springer, 2013.

[15] M. Schreiber, A. M. Hellmund, and C. Stiller. Multi-drive feature association for automated map generation using low-cost sensor data. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1140–1147, June 2015.

[16] M. Wang. *A Cognitive Navigation Approach for Autonomous Vehicles*. mbv, 2012.

[17] A. Wimmer, T. Jungel, M. Glck, and K. Dietmayer. Automatic generation of a highly accurate map for driver assistance systems in road construction sites. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 281–286, June 2010.

[18] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: Driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 99–108, New York, NY, USA, 2010. ACM.