Proceedings of 7th Transport Research Arena TRA 2018, April 16-19, 2018, Vienna, Austria

Semantic 3D Models from Real World Scene Recordings for Traffic Accident Simulation

Ludwig Mohr a, Martin Öttl c, Michael Haberl b, Matthias Rüther a,c, Horst Bischof a

[a]*Institute of Computer Graphics and Vision, Graz University of Technology, Inffeldgasse 16, 8010 Graz, Austria*
[b]*Institute of Highway Engineering and Transport Planning, Graz University of Technology, Rechbauerstraße 12 , 8010 Graz, Austria*
[c]*Holistic Imaging Meixner & Rüther OG, Nikolaiplatz 4, 8020 Graz, Austria*

Abstract

We propose a novel extension to traffic accident simulation by means of semantic 3D environment information allowing for a broader view by incorporating the entire close-by environment. In this course, the effect of Advanced Driver Assistance Systems (ADAS) can be simulated, as well as the visibility of objects from people's perspectives. We present an enclosed pipeline generating 3D objects, their extents and relative positions as well as their semantic class from a combination of photogrammetric recordings and LiDAR (Light Detection And Ranging) scans. By adjusting the desired level of detail, these objects are suitable for both direct integration into the 3D scene reconstruction for use in the accident simulation software PC-Crash, as well as for fine tuning parameters in traffic flow simulations and for convincing visualization and presentation of simulation results, be it in courts or to policy makers in urban planning.

Keywords: traffic accident simulation; semantic model; environment model; semantic segmentation; LiDAR; PC-Crash; VISSIM; city models; urban planning; traffic flow simulation.

1. Introduction and Motivation

We propose a novel extension of traffic accident simulation and traffic flow simulation by means of semantic 3D environment information. While traditional simulation workflows focus on the key contributors to an accident, we allow for a broader view by incorporating the entire close-by environment. In this course, the effect of ADAS systems can be simulated, as well as the visibility from people's perspectives of objects and other traffic participants at intersections. Another benefit is the convincing 3D visualization of an accident in its surroundings.

Instead of creating the 3D environment manually, or editing LiDAR (Light Detection and Ranging) scans to assign semantic classes like building, vehicle, vegetation, road etc., we propose a novel pipeline yielding 3D objects, their extents and relative positions, as well as their semantic class, from a combination of photogrammetric recordings and LiDAR scans. By adjusting the desired level of detail, these objects are suitable for both direct integration into the 3D scene reconstruction for accident simulation in PC-Crash (DSD, 2017) as well as for convincing visualization and presentation of simulation results in courts or for instance in presentations of traffic flow simulation results and urban planning proposals to policy makers.

Typically, microscopic traffic flow simulation models (e.g. SUMO, VISSIM, AIMSUN, PARAMICS etc.) are developed for traffic analyses which require far less consideration to detailed driver behaviour and infrastructure environment than that is essential for safety assessment. These models use time step and behaviour-based approaches that usually utilise psycho-physical car-following models for longitudinal vehicle movement and a rule-based algorithm for lateral movement. For modelling the road network geometry data from orthophotos, signal layout maps and other GIS sources are gathered (e.g. location of stop lines at intersections, location of pedestrian/cyclist facilities, location of bus stops, location of signals) to build up the model environment. Nevertheless, microscopic simulations tend to simplify certain circumstances ignoring the fact that these circumstances can have an impact on the results of the investigations. Inclination of the road networks is seldom incorporated in microscopic simulation models, although they can strongly influence efficiency of roadways, intersections, traffic signal control and also emissions due to different acceleration and deceleration behaviours. In terms of traffic safety there should be a closer focus on visibility and available sight distances subject to certain obstacles, such as buildings, other driving and also parking vehicles, vegetation and various road equipment's like signal posts or road signs. Visibility of objects from people's perspectives is strongly neglected in microscopic simulation models so far. Within this study the consideration of 3D environments to enhance microscopic simulation models should be tested.

To our knowledge, this is the first pipeline for road accident simulation and traffic flow simulation facilitating automatic creation of 3D semantic scene models from easily obtainable real world data. Since our proposed solution spares the need for laborious and time consuming manual editing and data conditioning, it offers fast and affordable modelling of scenes to a wide range of users and projects.

This paper presents intermediate results of ongoing research.

2. Model Generation Pipeline

Our model generation pipeline expects as input registered LiDAR point clouds with accompanying RGB images covering the 360° field of view around the respective reference point of the LiDAR scanner. As long as camera matrix, camera calibration as well as relative position and orientation with respect to the LiDAR point cloud (intrinsic and extrinsic camera calibration) are known, the choice of imaging system and respective mapping of the surrounding view is up to the user.

Additionally, the choice of the view point in the data acquisition is within reasonable bounds free, enabling the application of the proposed pipeline in a multitude of scenarios in which not every mode of acquisition may be feasible. As long as the resulting data is available in the correct format it may stem from aerial capturing by an Unmanned Aerial Vehicle (UAV), ground based capturing utilizing a car, it may even stem from a handheld reconstruction scanner.

RGB imagery provides on one hand highly dense semantic information on the scene, images with resolutions of tens of megapixels are easily obtainable. On the other hand, single images only provide directional information on the scene geometry given a camera calibration, but no depth. LiDAR data in contrast provides highly accurate depth information, yet due to the comparatively slow measurement process of LiDAR scanners densely capturing a scene is usually impractical in urban environments. For capturing systems mounted on a car which has to participate in regular traffic this is entirely impossible. Therefore, accurate depth information from LiDAR point clouds usually is sparse in contrast to the information available in colour imagery.

The overall goal now is twofold: leveraging semantic information from RGB imagery for automatic creation of semantic scene models as well as increasing the spatial resolution of the depth information for generation of detailed 3D models, ideally increasing the resolution of the depth to match that of the RGB imagery.

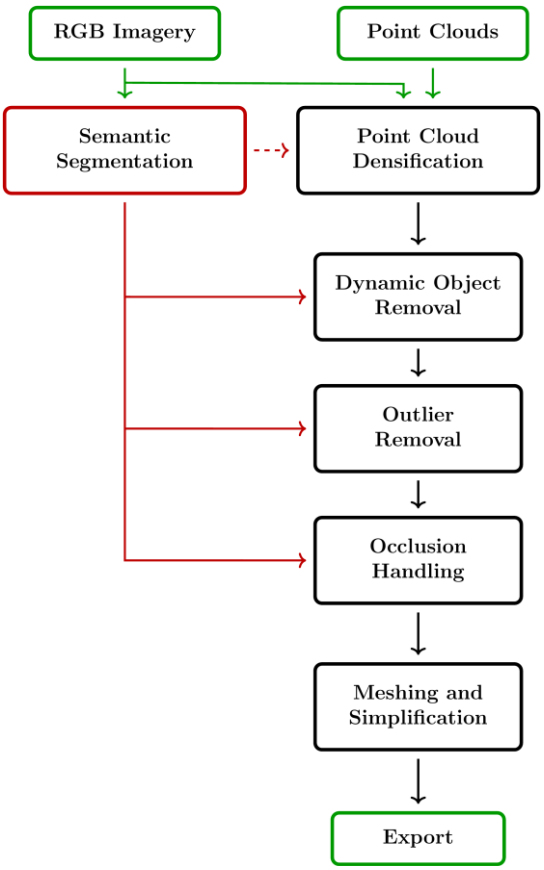

Fig. 1 Flowchart of the model creation pipeline. The colour red marks semantic information, black marks point cloud manipulations, and green marks I/O operations

In order to achieve this, after data acquisition, the pipeline implements fully automatic processing of the following steps as outlined in the flow diagram in Fig. 1:

- Semantic segmentation: Semantic segmentation is the process of assigning each pixel in photogrammetric recordings of a scene a label from the set of classes relevant to the given task. Classes relevant in this case are, among others, buildings, vehicles, vegetation, and roads (vehicle lanes and sidewalk). Utilizing state-of-the-art semantic scene segmentation of urban scenes gives important information for both model creation and model refinement.
- Densification of the sparse 3D point cloud: LiDAR scans from moving capturing systems produce sparse point clouds in general. In order to obtain dense depth information suitable for model generation we leverage the scene information from registered photogrammetric recordings. If LiDAR scanning is not feasible, this processing block can be replaced by either stereo matching methods or Structure from Motion (SfM) approaches if suitable data is provided instead, without impeding the functionality of other processing steps.

- Dynamic Object Removal: In general, it is not feasible or even impossible to close off whole streets for data acquisition, meaning that other traffic participants are almost always present in the raw dataset. Due to their independent motion they will be visible multiple times at different locations, thereby creating artefacts in the depth reconstruction. To remedy this we implement automatic detection and removal of traffic participants utilizing semantic segmentation.
- Outlier Removal: Measurement errors of the LiDAR scanner, depth measurements on dynamic objects which have moved rather far until the acquisition of the RGB image and occasional measures on extremely thin structures such as power lines, contact wires, or wire mesh fences all produce depth measurements unsuitable for reconstruction. The latter measurements, albeit being correct scene measures, are not feasible in reconstruction due to their scarcity in combination with the limited spatial extent of their respective structure. Another source of outliers are scattered points arising from artefacts in the densification step.
- Detection of Occlusions: Stationary vehicles as well as dynamic traffic participants occlude static structures behind them. The information on these parts has either to be filled in from previous or subsequent acquisitions or artificially generated if the corresponding area is not observed within the data.
- Point Cloud meshing: Meshing generates 3D models from the segmented objects allowing for easy adjustment of the level of detail appropriate for the desired task while still retaining high accuracy where needed.
- Export: The last stage exports the final model with all necessary attributes into compatible formats for direct import into PC-Crash for road accident simulation or into VISSIM (PTV, 2017) for traffic flow simulations or presentation of their results. The 3D scene can be used directly for simulations without any additional adjustments concerning the scene.

2.1. Semantic Segmentation on RGB Imagery

The task of semantic segmentation of an image can be defined as follows: assign each pixel in the image a semantic meaning. More precisely assign a label from a fixed set to which it belongs with the highest likelihood, thus producing a map with a pixel-wise dense classification. The set of possible labels usually depends on the task. For segmentation of urban scenes it often contains labels for the classes “car”, “truck”, “bus”, “house”, “pedestrian”, and “vegetation”, among several others.

In recent years, immense progress has been made in assigning correct semantic labels. Especially with the advent of Convolutional Neural Networks (CNNs) accelerated by General Purpose Graphics Processing Units (GPGPUs), the speed of development has further increased. A change in network architecture, called the Fully Convolutional neural Network (FCN), enables now the segmentation of images with arbitrary input sizes independent of the image size used for training, enabling this technology to be applied more easily within complex frameworks.

Table 1. Categories and classes in the Cityscapes Dataset

Category	Classes
flat	road · sidewalk · parking · rail track
human	person · rider
vehicle	car · truck · bus · on rails · motorcycle · bicycle · caravan · trailer
construction	building · wall · fence · guard rail · bridge · tunnel
object	pole · pole group · traffic sign · traffic light
nature	vegetation · terrain
sky	sky
void	ground · dynamic · static

Drawbacks of these deep learning techniques are their demand of significant amounts of pre-labeled training data. Furthermore, devising network architectures which generalize well to modalities not encoded in the training data is often challenging. Considering urban scene segmentation, these unseen modalities may arise from different architectural styles in different cities or different countries as well as for instance a change of view point between training data and target data. With the release of the Cityscapes Dataset in 2016 (Cordts 2016), there is now a large dataset providing dense pixel annotations of street scenes of 50 European cities with 30 semantic classes in eight

categories (cf. Table 1 for a complete overview over available classes in this dataset). Due to this high variability, segmentation networks trained on this dataset tend to generalize well on other European cities not contained in the dataset. FCNs trained for semantic segmentation on this dataset lend themselves as a starting point for the segmentation task at hand. In this spirit, we employ the network architecture Resnet38 (Wu 2016), one of the most successful network architectures for this task, trained on the Cityscapes Dataset. At this state in building the pipeline we employ this network without any adaptations as we found that it already delivers suitable segmentations despite the dataset not covering our general testing region of southern Austria and having a different view point (i.e., mounting height of the acquisition setup) than our data.

Planned future work on the pipeline contains fine-tuning of the pre-trained segmentation network with a small amount of labelled data from our dataset. This leverages the rich semantic information available in the Cityscapes Dataset with its thousands of labelled images, but at the same time adapts the model to the view point of our dataset using only some tens of labelled images.

Despite the enormous advances in segmentation performance neural networks brought, there are still challenging scene structures. These are mainly the correct distinction between highly similar classes, e.g. correct segmentation of roads and broad bicycle lanes and sidewalks on one hand, the correct classification of extremely thin structures like power lines or wire mesh fences on the other hand. Fig. 2 shows two examples of semantic segmentation results overlaid onto the original image.

Fig. 2 Segmentation examples, classes are colour coded and overlaid onto the original RGB images. On the right side and example of failed segmentation is visible, parts of the sidewalk and bicycle lane are incorrectly classified as road

2.2. Point Cloud Densification

A popular and well researched approach for creating 3D structures from two dimensional photogrammetric image series is Structure from Motion (SfM). SfM techniques exploit overlap in monocular image sequences by associating corresponding features between consecutive captures and estimating the relative camera motion as well as the geometric structure producing these recordings. These techniques work well for high acquisition framerates resulting in small relative camera movements, yet suffer from degrading results in presence of larger camera movements, as well as when dynamic objects are present in the scene. Furthermore, a high data overhead when recording at high framerates in urban environments can be expected due to frequent stops and variable driving speeds. For these reasons we focus on sensor systems equipped with LiDAR scanners for depth acquisition which produce highly accurate depth measurements and allow for a drastic reduction of the capture rate of photogrammetric data. A major drawback of LiDAR scanners is their comparatively slow acquisition rate. This results in a dataset with sparse yet highly accurate depth measures from the LiDAR scanner in combination with registered dense semantic information from associated RGB imagery.

Ferstl et. al. (2013) propose in their work on depth-image super-resolution the formulation within the framework of Total Generalized Variation with a L2 data term (TGV-L2) which favours piece-wise affine structures in the interpolation. Furthermore, they incorporate a regularizer derived from the anisotropic diffusion of radiometric images depicting the same scene as a guidance. This directs the interpolation to produce affine structures within

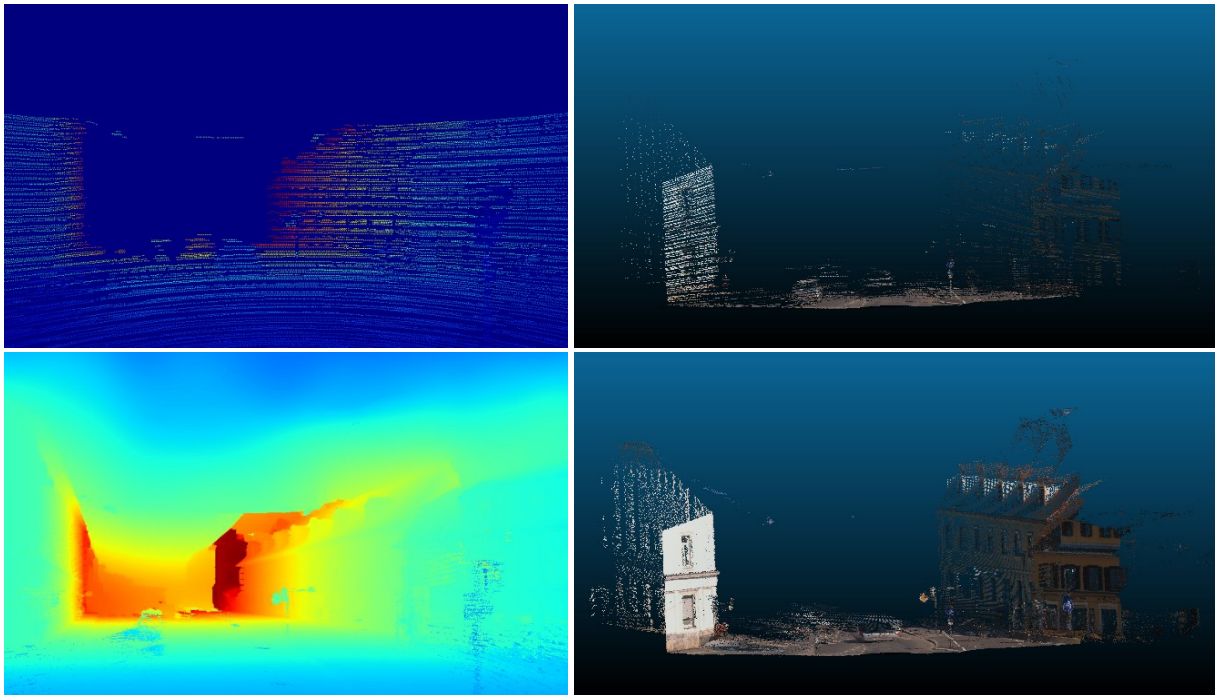

Fig. 3 Original sparse depth map (left) and corresponding point cloud (right) in the first row, super-resolved dense depth map (left) and corresponding point cloud (right) in the bottom row. Depth maps are color coded: blue hues encode close by depths, red hues points far away

areas without textural changes, while encouraging different structures at texture discontinuities. In man-made environments, such texture discontinuities often coincide with depth discontinuities. The combination with the TGV-L2 formulation favouring piecewise affine solutions makes this approach therefore especially suitable for urban scenes.

We implement the point cloud densification utilizing the approach of Ferstl et. al. (2013) by projecting the available LiDAR depth measurements back into the image to create the sparse depth-map, encoding the depth of the scene at a current pixel location in the pixel value. This generates pairs of depth-map and guidance image, which are subsequently super-resolved with anisotropic guided TGV-L2. The resulting dense depth-map is then back projected to create a dense 3D point cloud. Fig. 3 shows a sparse point cloud, the corresponding sparse depth-map as well as the super-resolved point-cloud and depth map.

2.3. Dynamic Object Removal

In urban environments the category of dynamic objects is comprised almost entirely of traffic participants such as cars and pedestrians. They pose a further challenge to the model creation since they move independently from the acquisition vehicle. Therefore, in consecutive acquisitions they are reproduced multiple times at different locations, thereby creating undesirable artefacts. Furthermore, for accident simulation all traffic participants should be removed entirely from the model anyways.

Removing these objects is done by leveraging the already available information from semantic segmentation. If for an image area the segmentation classifies the corresponding pixels as belonging to any of the respective categories, we restrict back projection of pixels into the scene in order to omit these pixels. This of course leaves holes in the model of a single acquisition. These holes are however filled by data from adjacent acquisitions in the vast majority of times.

The only cases where filling of holes fails is for the road below parked vehicles or walls next to them, and pedestrians waiting while the acquisition vehicle passes. Furthermore, in rare cases when large vehicles like lorries are driving just in the right way to conceal an area for the whole time the acquisition vehicle is in range, permanent occlusions may occur. In these cases, the respective areas have never been observed and therefore cannot be reconstructed from data. Yet at least for such holes left on the road automatic detection and removal by interpolation is straight forward. This however is left as a future task at this stage.

2.4. Outlier Removal

Albeit outliers in the point cloud can stem from wrong LiDAR measurements, most true outliers arise as artefacts from depth super resolution at depth discontinuities. Discontinuities highly susceptible to this are those at extremely thin structures like power lines or thin poles. We implement detection and removal of outliers based onto two approaches: statistical suppression and semantic filtering.

Semantic filtering is realized by rejecting all depth points in image areas classified as sky. This gets rid of false interpolations at transitions from solid objects to the sky as well as depth values from very thin structures for which not enough information for correct modelling is available from neither LiDAR measurements nor colour imagery.

Statistical outlier suppression is applied in three steps. First, we remove all interpolated depth points outside of the area in which true LiDAR measurements are projected. For these points extrapolation is actually performed which is an under-constrained task and produces therefore highly unreliable results. In a second step we analyse the normal directions of depth points and remove those with normal vectors perpendicular to the viewing direction of the sensor. This is based on the observation that points from the point cloud with normal vectors perpendicular to the viewing direction of the sensor are artefacts from the densification step with an extremely high likelihood. In a third step we remove sparse scattered points by analysing the mean distance of a depth point to a subset of its nearest neighbours, rejecting all points with a mean distance larger than a defined threshold. This last step removes points where depth values have been incorrectly interpolated between a foreground object and a background object.

2.5. Occlusion Detection and Handling

Considering occlusions two sources have to be considered. One kind emanates from dynamic objects in the scene being removed, the other kind is caused by parallax resulting from the capturing system moving through the scene. Resolving occlusions caused by dynamic objects is again implemented utilizing the semantic information segmented from the associated RGB image: since we know where and when in the scene the occlusion happened and therefore depth information is lacking due to dynamic object removal, we keep track of the area and corresponding view point. After model creation, we check if merging multiple views has automatically filled the area, otherwise we mark it as unobserved area for later handling. This will occur most often for stationary instances from the category of dynamic objects, most commonly parked cars.

Fig. 4 Errors due to parallax when warping LiDAR points captured after the colour images back to a common trigger point. The street sign and the traffic light in the foreground are projected onto the facade in the background. Additionally, artefacts from the densification step can be seen as points scattered between the facade in the background and the post in the foreground. The effect has been exaggerated for visualization

Generally, LiDAR point clouds are recorded continuously, whereas the accompanying colour imagery is recorded only at fixed time or distance intervals. LiDAR points collected before and after the acquisition are warped to match the exact time of image acquisition, due to parallax it may occur that LiDAR points now resting on a foreground object were actually measured at an object behind it. This happens most likely at edges of traffic signs

which are naturally rather close to a capturing vehicle in comparison to a facade in the background. Therefore, they suffer most from parallax induced errors. Fig. 4 shows an example of this kind of error.

This second source of errors induces artefacts by "smearing" depth values between facades in the background and edges of street signs, or by forcing the image of the sign to be projected onto the facade. These artefacts can be handled by a combination of intelligent filtering of depth points aided again by the semantic segmentation as well as statistical outlier removal suppressing depth points with only far away neighbours.

2.6. Point Cloud Meshing and Model Generation

Up until now we still have separate point clouds for each acquisition. This last step fuses all point clouds into a single file from which we then again extract overlapping blocks for meshing, thereby collecting all points within the same volume in the scene into the same point cloud. Meshing is performed with screened Poisson surface reconstruction, proposed by Kazhdan and Hoppe (2013), which creates watertight surfaces from oriented point sets. This approach incorporates the points as hard constraints, thereby assuring that the known depths are accurately accounted for in the meshed model. After meshing, we apply Laplacian-based mesh simplification (Zhang et. al. 2012) to reduce the model size and complexity in uniform regions while preserving detailed features where necessary. Meshing and simplification are performed for each desired class separately, thereby retaining semantic information and allowing object-specific model complexities.

Functionality for exporting a model for simulation compatible with PC-Crash or VISSIM is planned but at this stage not yet implemented.

3. Prototype Evaluation

We test the prototype of the pipeline using data of a 400 meters stretch of street in the city of Graz, Austria. The dataset was captured using the UltraCam Mustang of Vexcel Imaging producing georeferenced LiDAR point clouds and images of the surrounding view at equidistant intervals of 4 meters. This results in a dataset of 100 captures for the evaluated stretch. Each capture consists of a LiDAR point cloud of roughly 60,000 data points and RGB images of the six faces of a cube presenting the surrounding view (front, right, back, left, down, and up) with a resolution of 9 megapixels each. Fig. 5 shows an example of a recorded point cloud and the associated imagery.

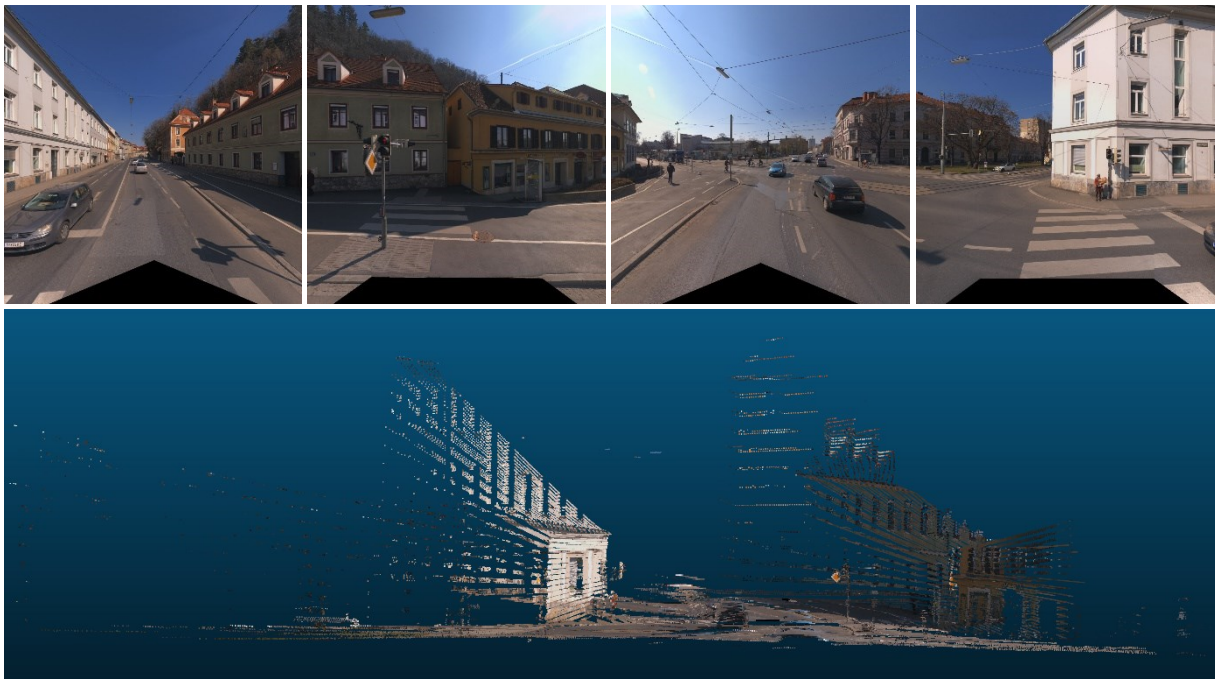

Fig. 5 A single captured data point. In the top row the forwards, rightwards, backwards, and leftwards facing colour images (top and bottom facing images have been omitted) are shown. In the bottom row the associated sparse point cloud can be seen

The pipeline is implemented in Python with the OpenCV computer vision library (super-resolution, filtering, outlier and dynamic object removal, occlusion handling), semantic segmentation using the MXNet deep learning framework (Chen et. al. 2015), and meshing and mesh simplification in C++. At this stage, densification and estimation of point orientations accounts for the vast majority of the processing time (about 4 hours per acquisition), with semantic segmentation (about 1 minute per acquisition) and meshing and mesh simplification (ca. 2 hours for the whole model) attributing almost negligible processing time in comparison. This results in an overall model creation time of roughly 400 hours in a sequential single threaded implementation on a desktop machine equipped with an Intel Core i7-4790 running Debian Sid. The reason behind the vast processing time for super-resolution is twofold: on one hand the formulation as numeric optimization problem is computationally demanding in and of itself, on the other hand the implementation is still in a prototype stadium without any optimizations. Since all processing steps up until point cloud consolidation and meshing are performed separately for each acquisition. This lends itself for parallel processing, reducing the overall processing time significantly.

In this stage, we restrict the model creation to three semantic classes, namely: street (including sidewalks and bicycle lanes), facades, and a third generic class collecting all remaining objects. Creation of more detailed models is left to future work.

An overview of the reconstructed street is shown in Fig. 6, whereas Fig. 7 shows partial views of the resulting model in more detail. Parts of the scene classified as “street” and “sidewalk” are shown on the left of Fig. 7. It can be seen that some over segmentation occurs for parts of the vegetation on the left side of the street. The result of adding the facade are shown in Fig. 7 centre. Again parts of the vegetation on the left have been misclassified, this time as facade. On the right of Fig. 7, the complete model of the scene can be seen. Since dynamic objects have already been removed, as expected most of the addition to the model is vegetation. The figures are best viewed in the digital version of this paper on the computer.

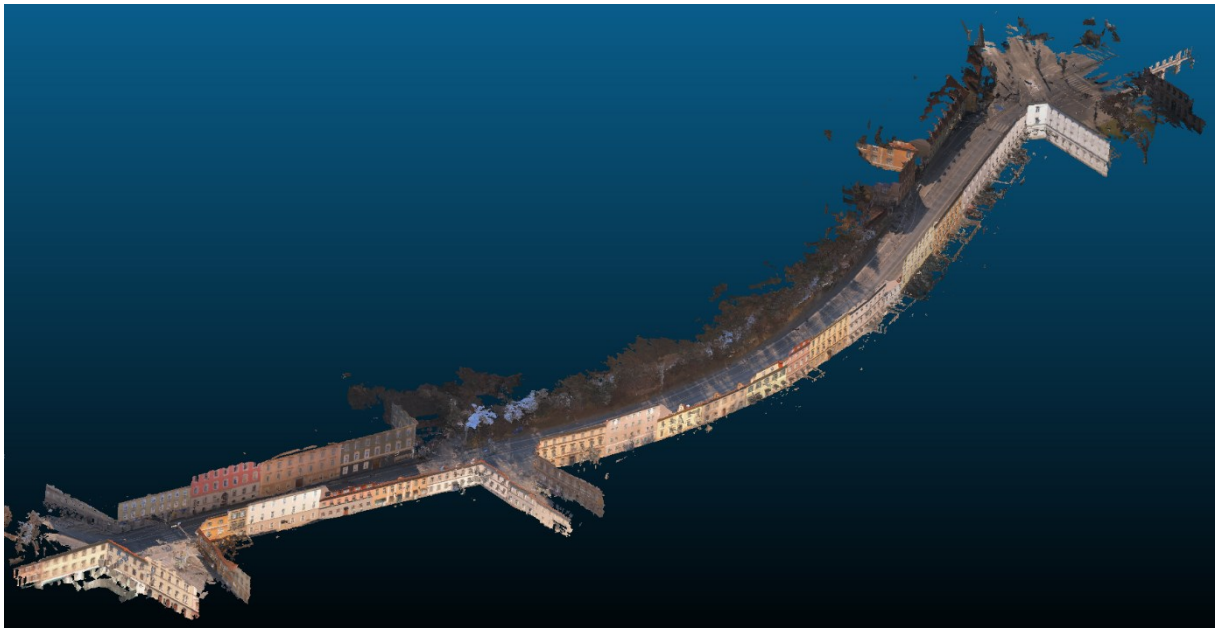

Fig. 6 Overview of the complete model of the captured street

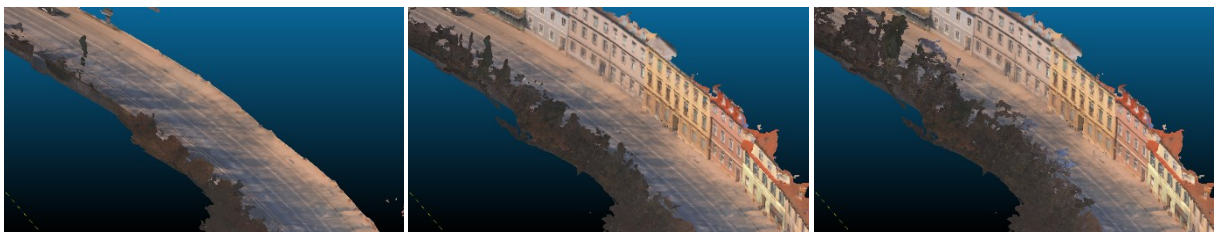

Fig. 7 Detailed view of a section of the reconstructed street. On the left the part segmented as "street", in the middle "street" and "facade", on the right side the complete reconstruction

4. Conclusion and Outlook

In this work we present a prototype of a novel pipeline creating accurate metric 3D models of urban scenes enriched with semantic information from LiDAR point clouds and registered photogrammetric recordings with minimal manual intervention. The results show that automatic creation of flexible 3D models for simulation and presentation from easily obtainable real world data is feasible. The resulting model can be used in the accident simulation software PC-Crash, where due to the sematic labels of the model appropriate parameters can automatically be set. If anything, only minimal manual augmentations may be necessary, thereby reducing the effort for recreating the environment of a scene of an accident. Since objects like traffic signs, vegetation, etc. are modelled together with their precise location and extent, the visibility among traffic participants can be assessed more accurately, thereby producing more meaningful simulation results. Traffic flow simulation may profit from these models, since for example visibility at junctions is a key factor for correct parameterization of e.g. approach velocity, yet this may not be accurately inferable from available road maps. From the model, they can be automatically inferred and set appropriately. Further information relevant to traffic flow simulation available in the 3D model is, among others, the inclination and the cross slope of roadways. Evaluation of the impact of ADAS systems on traffic flow or on the circumstances of an accident will profit again from accurate visibility information, especially in conjunction with semantic information. Depending on the ADAS technology used, for instance vegetation may be invisible to some ADAS systems but not others.

As open work for improving and completing the pipeline the following tasks have been identified:

- Improving the segmentation result by fine-tuning the network on data recorded by us to account for the different view point of UltraCam Mustang.
- Consolidation of the segmentation results if semantic labels change with distance to the object.
- Improving the densification performance of the pipeline, adopting a deep-learning approach with a Deep Primal-Dual Network implementation as proposed by Riegler et. al. (2016) or leveraging state of the art solutions following the approach of Uhrig et. al. (2017) with a sparsity invariant CNN.
- Augmenting the resulting model to fill in holes in streets or facades when they have not been observed during capturing due to occlusions.
- Augmenting the model for presentation purposes by a) texturing it using the RGB image with the best fitting view and b) augmenting buildings in the model with roofs from e.g. readily available aerial map data since they cannot be observed by ground based capturing systems.

Acknowledgements

This study is part of the project IMPROVE and is financially supported by the FFG, the Austrian Research Promotion Agency of the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT).

5. References

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. 2016. *"The Cityscapes Dataset for Semantic Urban Scene Understanding"*. In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

Zifeng Wu, Chunhua Shen, Anton van den Hengel. 2016. *"Wider or Deeper: Revisiting the ResNet Model for Visual Recognition"*. arXiv preprint: 1611.10080

D. Ferstl, C. Reinbacher, R. Ranftl, M. Ruether and H. Bischof. 2013. *"Image Guided Depth Upsampling Using Anisotropic Total Generalized Variation"*. IEEE International Conference on Computer Vision, Sydney, NSW, 2013, pp. 993-1000. doi: 10.1109/ICCV.2013.127

Michael Kazhdan and Hugues Hoppe. 2013. *"Screened Poisson Surface Reconstruction"*. In ACM Transactions on Graphics, 32, 3, Article 29 (July 2013), 13 pages. DOI: https://doi.org/10.1145/2487228.2487237

Lin Zhang, Zhen Ma, Zhong Zhou, and Wei Wu. 2012. *"Laplacian-Based feature preserving mesh simplification"*. In Proceedings of the 13th Pacific-Rim conference on Advances in Multimedia Information Processing (PCM'12), Weisi Lin, Dong Xu, Anthony Ho, Jianxin Wu, and Ying He (Eds.). Springer-Verlag, Berlin, Heidelberg, 378-389. DOI=http://dx.doi.org/10.1007/978-3-642-34778-8_35

Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, Zheng Zhang, 2015. *"MXNet: A Flexible and Efficient Machine Learning Library for Heterogeneous Distributed Systems"*. E-print arXiv:1512.01274

DSD Dr. Steffan Datentechnik, 2017. PC-Crash – Collision & Trajectory Accident Reconstruction Software. Linz, Austria.

PTV (Planung Transport Verkehr AG), 2017. VISSIM 9.00 User Manual. Karlsruhe, Germany, PTV.

UltraCam Mustang by Vexcel Imaging, http://www.vexcel-imaging.com/products/ultracam-mustang/

OpenCV, 2017. Open Source Computer Vision Library. http://opencv.org/

Gernot Riegler, David Ferstl, Matthias Rüther and Horst Bischof. 2016. *"A Deep Primal-Dual Network for Guided Depth Super-Resolution"*. In Richard C. Wilson, Edwin R. Hancock and William A. P. Smith, editors, Proceedings of the British Machine Vision Conference (BMVC), pages 7.1-7.14. BMVA Press, September 2016.

Uhrig, Jonas and Schneider, Nick and Schneider, Lukas and Franke, Uwe and Brox, Thomas and Geiger, Andreas. 2017. *"Sparsity Invariant CNNs"*. In Proceedings of International Conference on 3D Vision (3DV), 2017.