Urban 3D Semantic Modelling Using Stereo Vision

Sunando Sengupta[1] Eric Greveson[2] Ali Shahrokni[2] Philip H. S. Torr[1]
{ssengupta, philiptorr}@brookes.ac.uk, eric@greveson.co.uk, Ali.Shahrokni@2d3sensing.co.uk
[1]Oxford Brookes University [2]2d3 Ltd.

Abstract— **In this paper we propose a robust algorithm that generates an efficient and accurate dense 3D reconstruction with associated semantic labellings. Intelligent autonomous systems require accurate 3D reconstructions for applications such as navigation and localisation. Such systems also need to recognise their surroundings in order to identify and interact with objects of interest. Considerable emphasis has been given to generating a good reconstruction but less effort has gone into generating a 3D semantic model.**

The inputs to our algorithm are street level stereo image pairs acquired from a camera mounted on a moving vehicle. The depth-maps, generated from the stereo pairs across time, are fused into a global 3D volume online in order to accommodate arbitrary long image sequences. The street level images are automatically labelled using a Conditional Random Field (CRF) framework exploiting stereo images, and label estimates are aggregated to annotate the 3D volume. We evaluate our approach on the KITTI odometry dataset and have manually generated ground truth for object class segmentation. Our qualitative evaluation is performed on various sequences of the dataset and we also quantify our results on a representative subset.

I. INTRODUCTION

In this paper we propose a robust computer vision algorithm that uses images from stereo cameras mounted on a vehicle to generate a dense 3D semantic model of an urban environment. In our 3D model, every voxel is either assigned to a particular object category like road, pavement, car, etc., free space or object's interior. We are motivated by the fact that autonomous robots navigating in an urban environment need to determine their path and recognise the objects in the scene [1], [21], [5] .

Currently most autonomous vehicles rely on laser based systems which provide sparse 3D information [15] or a locally metric topological representation of the scene [12]. Sparse laser-based systems lack the details that are required for classifying objects of interest [6], [18] and for accurate boundary predictions [20]. To obtain an accurate understanding of the scene, a dense metric representation is required [16]. We show that such a representation can be obtained using a vision based system. Moreover, compared to normal cameras, the laser sensors are expensive and power hungry, can interfere with other sensors, and have a limited vertical resolution [9].

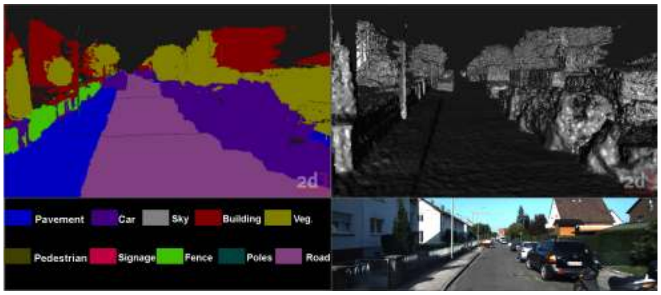

Fig. 1: *3D semantic reconstruction.* The figure shows a sample output of our system. Dense 3D semantic reconstruction along with class labels is shown in the left and the surface reconstruction is shown in the right. Bottom right shows one of the image corresponding to the scene.

Recently, Newcombe et al. [16] proposed a system for dense 3D reconstruction using a hand-held camera. The dense model is generated from overlapping depth-maps computed using every image pixel instead of sparse features, thus adding richness to the final model. Geiger et al. [9] proposed a method for fast dense reconstruction of road scenes from stereo images. Their method uses a point cloud representation which is updated by averaging the estimated 3D points and as a consequence, can quickly suffer from accumulated drift.

In the context of object class segmentation, computer vision algorithms have been effectively applied to the semantics of road scenes [3]. These algorithms work in the image domain where every pixel in the image is classified into an object label such as car, road, pavement etc. Object class segmentation in the image domain was extended to generate a semantic overhead map of an urban scene from street level images [20], or a coarse 3D interpretation in the form of blocks in [10], and with a stixel representation in [7]. The most related to our work is [14], where a joint representation of object labelling and disparity estimation is performed in the image domain. However, none of these methods deliver a dense and accurate 3D surface estimation.

In this paper we perform a 3D semantic modelling for large scale urban environments. Our approach is illustrated in Fig. 2 and a sample output of our system is shown in Fig. 1. The input to our system is a sequence of calibrated, stereo image pairs rectified so that the image scan lines correspond to epipolar lines. We use a robust visual odometry method with effective feature matching to track the camera poses (§ II-A). The estimated camera poses are used to fuse the depth-maps generated from stereo pairs, producing a volumetric 3D representation of the scene. This

[2]2d3 Ltd. has been funded by the Defence Science and Technology Laboratory (DSTL), Futures and Innovation Domain. DSTL is part of the UK Ministry of Defence.

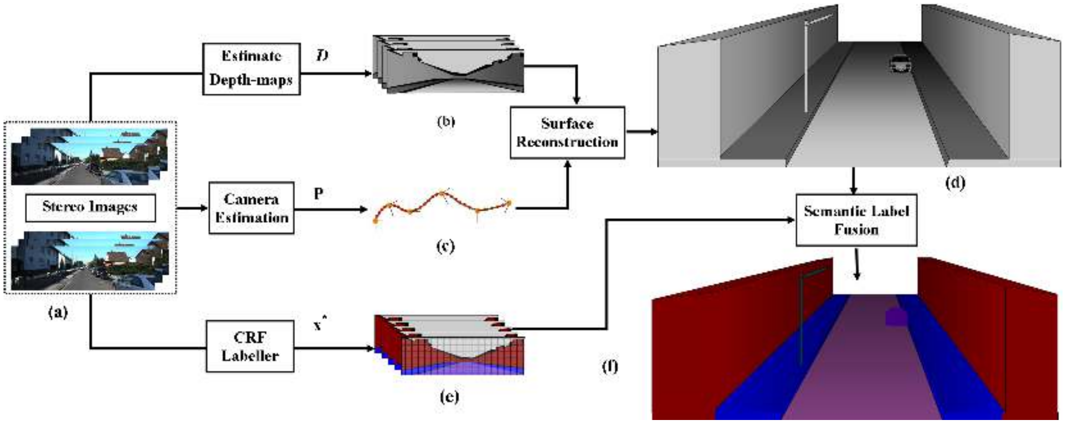

Fig. 2: *System Overview.* (a) Shows the input to our method which is a sequence of rectified image pairs. The disparity map (b) is computed from the images and (c) is the camera track estimation. The outputs of (b) and (c) are merged to obtain a volumetric representation of the scene (d). (e) shows the semantic segmentation of the street images which is then fused into a 3D semantic model of the scene (f). Best viewed in colour.

is done online to enable reconstruction over long street image sequence (§ II-B). In parallel, the pixels in the input views are semantically classified using a CRF model. The label predictions are aggregated across the sequence in a robust manner to generate the final 3D semantic model (§ II-C). We evaluate both object labelling and odometry results our method on the KITTI [8] dataset.

II. Semantic 3D Reconstruction of the World

In this section we explain the individual stages of the semantic 3D reconstruction pipeline in detail.

A. Camera Pose Estimation

The camera pose estimation has two main steps, namely feature matching and bundle adjustment. We assume calibrated stereo cameras positioned on a rigid rig.

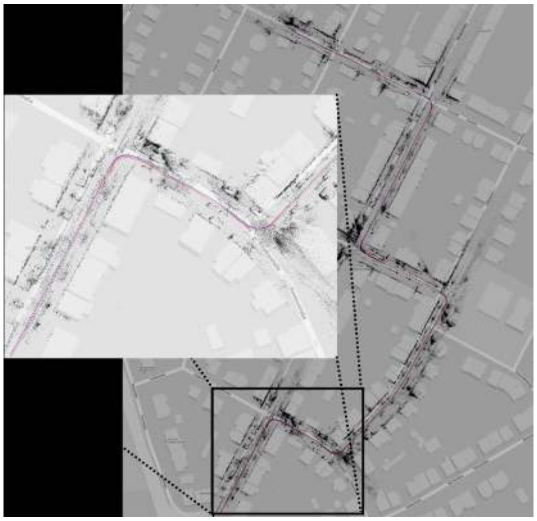

Fig. 3: Bundle adjustment results, showing camera centres and 3D points, registered manually to the Google map.

Feature matching: The feature matching comprises of two stages, stereo matching and frame by frame matching. For stereo matching we try to find potential matches across the epipolar line based on the sum of squared differences score of an 8×8 patch surrounding the candidate pixel (for an image of resolution 1241×376). Each potential match is cross-checked to ensure it lies in a valid range (i.e. minimum depth/maximum disparity magnitude) and the fact that points must be in front of both cameras. The image matches are cross-checked for both left-right and right-left pairs and the agreed matches are kept. After the list of stereo matches is obtained, we perform frame to frame matching for both left and right images. The basic approach is similar to the stereo matching framework, except that we do not rely on epipolar constraints.

Once the matches are computed, the corresponding feature tracks are generated. All the stereo matches which also have corresponding frame-to-frame matches are kept in the track. Having this agreement between both the stereo and ego-motion helps the bundle adjuster to estimate the camera poses and feature points more accurately by rejecting false matches, and simplifies the feature point initialisation phase in the bundle adjuster. We use a bundle method where our optimiser estimates camera poses and the associated features viewed by the last n cameras, leading to lower accumulated drift by reducing noise over n frames. In our experiments we set $n = 20$ which we found to be a good compromise between speed and accuracy. The example result of bundle adjustment is shown in Fig. 3, where the camera track and the 3D points are overlayed manually on the Google map demonstrating a near perfect match between the tracked camera positions and the actual street layout.

B. Surface Reconstruction

For generating the surface, we first estimate the depth maps from stereo pairs. These are merged using the Truncated Signed Distance Function (TSDF) and finally a mesh is created using marching tetrahedra algorithm. The individual steps are described in detail below.

Depth Map Generation: Given a rectified stereo image pair, a depth map is computed from the disparity image as: $z_i = B.f/d_i$, where z_i and d_i are the depth and the disparity corresponding to the i^{th} pixel respectively. The terms B and f are the camera baseline and the focal length, which are computed from the camera matrix. The disparity is computed using the OpenCV implementation of Semi-Global Block Matching (SGBM) method [11]. The depth values are clipped based on a permissible disparity range.

TSDF *Volume Estimation:* Each depth map with estimated camera parameters is fused incrementally into a single 3D reconstruction using the volumetric TSDF representation [4]. A signed distance function corresponds to the distance to the closest surface interface (zero crossing), with positive values corresponding to free space, and negative values corresponding to points behind the surface. The representation allows for the efficient registration of multiple surface measurements, by globally averaging the distance measures from every depth map at each point in space.

We assume that the depth of the true surface lies within $\pm\mu$ of the observed values from the depth maps. So the points that lie in the visible space at a distance greater than μ are truncated to μ. The points beyond μ in the non-visible side are ignored. The TSDF values are computed for each depth map. They are merged using an approach similar to [16] where an averaging of all TSDF's is performed. This smoothens out the irregularities in the surface normals of the individual depth estimate computed from a single stereo pair.

Online Volume Update: As we are reconstructing road scenes which can run from hundreds of meters to kilometers, we use an online grid update method. We consider an active $3\times3\times1$ grid of voxel volumes at any time of the fusion. We allow for only one volume in the vertical direction assuming minor elevation changes compared to the direction of motion. For every new depth map, the grid is updated. As the vehicle track goes out of the range of current grid, the current grid blocks are written to memory and a new grid is initialised. This allows us to handle arbitrarily long sequence without losing any granularity of the surface.

Triangulated Meshing using Marching Tetrahedra: In order to obtain a complete meshed surface, we first infer an iso-surface from the TSDF field by finding all the zero crossings. Then we use the Marching Tetrahedra algorithm [17] to extract a triangulated mesh of the zero valued iso-surface. Fig. 4 shows an example output of the surface reconstruction. The reconstructed model captures fine details which is evident with the pavements, cars, road and vegetation. Once the rendered mesh is obtained, the faces of the mesh are associated with object class labellings. This is more efficient than performing labelling of all the 3D points and still produces a dense labelling.

C. Semantic Model Generation

We use a Conditional Random Field (CRF) based approach that performs a pixel-wise classification on the street level images similar to [13] which is briefly described below. Consider a set of random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$, where each variable $X_i \in \mathbf{X}$ takes a value from a pre-defined label set $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$. A labelling $\mathbf{x}$ refers to any possible assignment of labels to the random variables and takes values from the set $\mathbf{L} = \mathcal{L}^N$. The random field is defined over a lattice $\mathcal{V} = \{1, 2, \ldots, N\}$, where each lattice point, or pixel, $i \in \mathcal{V}$ is associated with its corresponding random variable X_i. Let $\mathcal{N}$ be the neighbourhood system of the random field defined by sets $\mathcal{N}_i, \forall i \in \mathcal{V}$, where $\mathcal{N}_i$ denotes the set of all neighbours (usually the 4 or 8 nearest pixels) of the variable X_i. A clique c is defined as a set of random variables $\mathbf{X}_c$ which are conditionally dependent on each other. The corresponding energy $E(\mathbf{x})$ is given by: $E(\mathbf{x}) = \sum_{c\in\mathcal{C}} \psi_c(\mathbf{x}_c)$, where the term $\psi_c(\mathbf{x}_c)$ is known as the potential function of the clique c, and $\mathcal{C}$ is the set of all the cliques. The most probable or maximum a posteriori labelling $\mathbf{x}^*$ of the CRF is defined as: $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbf{L}} E(\mathbf{x})$. The energy minimisation problem is solved using a graph-cut based Alpha Expansion algorithm [2].

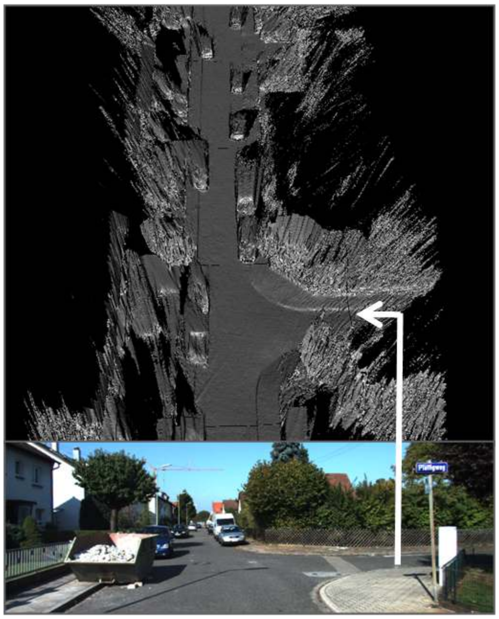

Fig. 4: *Volumetric surface reconstruction.* Top figure shows the 3D surface reconstruction over 250 frames (KITTI sequence 15, frames 1-250) with street image shown at the bottom. The arrow highlights the relief of the sidewalk which is correctly captured in the 3D model.

Street Level Image Segmentation: For our application, the label set is $\mathcal{L}$ = {pavement, building, road, vehicle, vegetation, signage, pedestrian, wall/fence, sky, post/pole}. We used the associative hierarchical CRF [13] which combines features and classifiers at different levels of the hierarchy (pixels and superpixels). The Gibbs energy for a street-level image is:

$$E(\mathbf{x}) = \sum_{i\in\mathcal{V}} \psi_i(x_i) + \sum_{i\in\mathcal{V}, j\in\mathcal{N}_i} \psi_{ij}(x_i, x_j) + \sum_{i\in\mathcal{V}, j\in\mathcal{N}_i} \psi^d_{ij}(x_i, x_j) + \sum_{c\in\mathcal{C}} \psi_c(\mathbf{x}_c) \qquad (1)$$

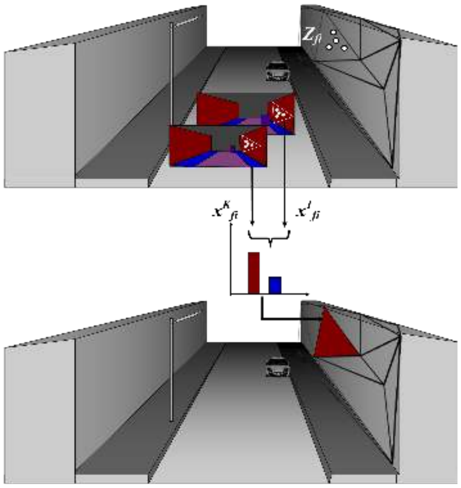

Fig. 5: *Label fusion scheme.* The white dots Z_{f_i} are the sampled points on the face of a mesh triangle. The corresponding image points $x^j_{f_i}$ are obtained by projecting the face points onto the labelled street image. The mesh is labelled with the class label with the most votes on the mesh face.

Unary potential: The unary potential ψ_i describes the cost of a single pixel taking a particular label. We have used the multi-feature variant of the TextonBoost algorithm [13].

Pairwise potentials: The pairwise term ψ_{ij} is an edge preserving potential which induces smoothness in the solution by encouraging neighbouring pixels take the same label. It is defined over an neighbourhood of eight pixels taking the form of a contrast sensitive Potts model:

$$\psi_{ij}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j, \\ g(i,j) & \text{otherwise}, \end{cases} \tag{2}$$

, where the function $g(i,j)$ is an edge feature based on the difference in colours of neighbouring pixels [13], defined as:

$$g(i,j) = \theta_p + \theta_v \exp(-\theta_\beta ||I_i - I_j||_2^2), \tag{3}$$

where I_i and I_j are the colour vectors of pixels i and j respectively. $\theta_p, \theta_v, \theta_\beta \geq 0$ are model parameters set by cross validation. The disparity potential $\psi^d_{ij}(x_i, x_j)$ takes the same form as the pairwise potential but operates on the disparity image, where neighbouring pixels with similar disparity are encouraged to take same labels. Adding information from both image and disparity domain helps us to achieve more consistent results (we give equal importance to both these terms). An alternative potential based on the full depth map could be considered, however the back projected points can be sparse in the image domain, which is not suitable for the per-pixel inference used here.

Higher Order Potential: The higher order term $\psi_c(\mathbf{x}_c)$ describes potentials defined over overlapping superpixels as described in [13]. The potential encourages the pixels in a given segment to take the same label and penalises partial inconsistency of superpixels. This captures longer range contextual information.

Semantic Label Fusion : Once the street level image segmentations are obtained, the label predictions are fused as follows: for each triangulated face f in the generated mesh model, we randomly sample i points (Z_{f_i}) on the face. The points are projected back in to K images using the estimated camera pose (P^k), resulting in a set of image points ($x^k_{f_i}$). The label predictions for all those image points are aggregated and the majority label is taken as the label of the face in the output model. The label histogram Z_f for the face f is given as:

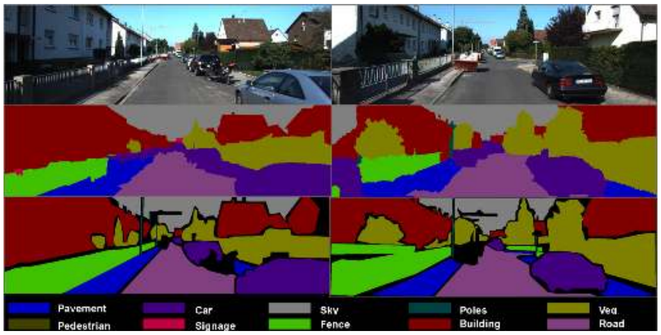

Fig. 6: *Semantic image segmentation*: The top row shows the input street-level images and the middle row shows the output of the CRF labeller. The bottom row shows the corresponding ground truth for the images.

$$Z_f = \frac{1 + \sum_{i \in Z_{f_i}} \sum_{k \in K} \delta(x^k_i = l)}{|L|} \tag{4}$$

where l is a label in our label-set $\mathcal{L}$ and $|L|$ is the number of the labels in the label set. This provides a naive probability estimation for a mesh face taking label l. We set $i = 10$ in our experiments. The fusion step is illustrated in Fig. 5. The white dots in the model (top) are projected back to the images. The class label prediction of all those image points are aggregated to generate the histogram of labels and the final label prediction is made accordingly. Instead of considering all the points in the face triangle, sampling a few random points is fast, provides robustness to noise and avoids aliasing problems. This is better than considering only the face vertices label in the image because in the final model the face vertices' labels are likely to coincide with the object class boundary.

III. Experiments

To demonstrate the effectiveness of our proposed system, we have used the publicly available KITTI dataset [8] for our experiments. The images are 1241×376 at full resolution. They are captured using a specialised car in urban, residential and highway locations, making it a varied and challenging real world dataset. We have manually annotated a set of 45 images for training and 25 for testing with per-pixel class labels. The class labels are road, building, vehicle, pedestrian, pavement, tree, sky, signage, post/pole, wall/fence[1] .

We evaluate our camera pose estimation using two metrics, translation error (%) and rotation error (degrees/m) over an increasing number of frames with the ground truth provided by [8] of sequence 8 (see table I). We evaluate our sliding window bundle method (*full*) and a *fast* variant of that. The

[1] available at http://cms.brookes.ac.uk/research/visiongroup/projects.php

TABLE I: Odometry results: Translational and rotational error with increasing number of frames.

	Trans. Error (%)		Rot. error (degs/m)	
Length (frames)	fast	full	fast	full
5	12.2	12.15	0.035	0.032
10	11.84	11.82	0.028	0.026
50	8.262	8.343	0.021	0.018
100	4.7	4.711	0.019	0.013
150	3.951	3.736	0.017	0.01
200	3.997	3.409	0.015	0.009
250	4.226	3.209	0.013	0.007
300	4.633	3.06	0.012	0.007
350	5.057	2.939	0.011	0.006
400	5.407	2.854	0.01	0.004

fast method performs the Levenberg-Marquardt minimisation for two successive frame pairs to estimate camera pose and the feature points. As expected the average error for the *full* method reduces with increasing number of frames. Also the absolute magnitude of error for the *fast* method is larger than compared to the *full* method. Our *full* bundle method runs takes around 3.5 seconds per frame on a single core machine. However the *fast* method runs at approximately 4 fps. The feature extraction takes about 0.02 seconds, feature matching (both stereo and frame to frame) takes 0.2 seconds per frame. For disparity map extraction we use OpenCV implementation of semi-global block matching stereo [11] which takes around 0.5 seconds for the full sized 1280×376 image. The TSDF stage is highly parallelisable as each voxel in the TSDF volume can be treated separately. Currently, our implementation considers around 10 million voxels per $3 \times 3 \times 1$ grid of TSDF volume, running on a single core. All these steps can be optimised using the GPU implementation [19].

Fig. 6 shows the qualitative results of the street level image segmentation using our CRF framework. The first column shows the street-level images captured by the vehicle. The second and the third column show the semantic image segmentation of the street images and the corresponding ground truth. A qualitative view of our 3D semantic model is shown in Fig. 7. The arrows relate the positions in the 3D model and the corresponding images. We can see in (a), both fence and pavement are present in the model as well as the associated images. The model can capture long and thin objects like posts as shown in (b). The circle in the image (c) shows the car in the image, which has been captured correctly in the final model. In (d) arrows show the vegetation and the car respectively. In Fig. 8, a large scale semantic reconstruction, comprising of 800 frames from KITTI sequence 15, is illustrated. An overhead view of the reconstructed model is shown along with the corresponding Google Earth image. The inset image shows the actual path of the vehicle (manually drawn).

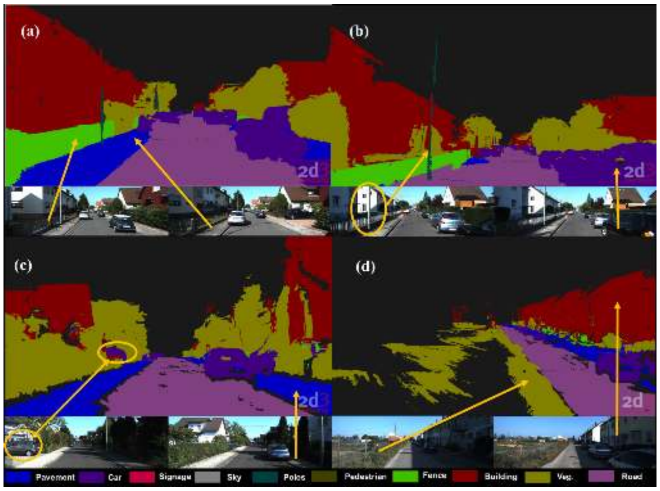

Fig. 7: *Closeup view of the 3D model.* The arrows relate the image locations and the positions in the 3D model.

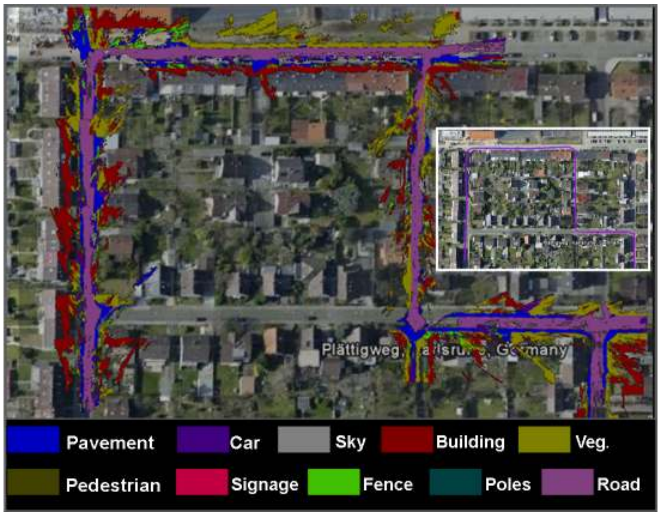

Fig. 8: Semantic model of the reconstructed scene overlayed with the corresponding Google Earth image. The inset image shows the Google earth track of the vehicle.

Next we describe the quantitative evaluation. For object level classification, we use an approach similar to [20]. As generating ground truth data for large sequences is expensive, we evaluate our model by projecting the semantic labels of the model back into the image domain using the estimated camera poses. Points in the reconstructed model that are far away from the particular camera (> 20m) are ignored. The projection is illustrated in Fig. 9. We show quantitative results on two metrics, recall and intersection *vs* union measures, for both street image segmentation and semantic model. Our results are summarised in table II. 'Global' refers to the overall percentage of pixels correctly

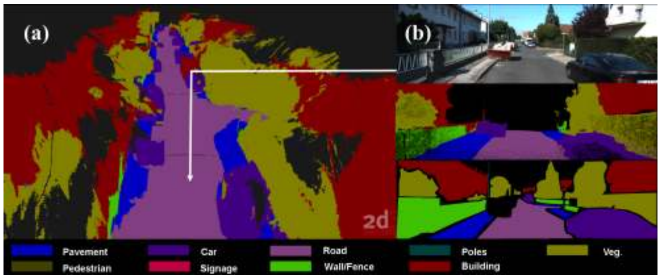

Fig. 9: *3D semantic model evaluation.* (a) shows the 3D semantic model. (b) shows the input image (top), corresponding image with labels back-projected from the 3D model (middle) and the ground truth image (bottom).

Method	Building	Vegetation	Car	Road	wall/fence	Pavement	Pots/Pole	**Average**	**Global**
Recall									
Image Segmentation	97.0	93.4	93.9	98.3	48.5	91.3	49.3	81.68	88.4
Semantic Model	96.1	86.9	88.5	97.8	46.1	86.5	38.2	77.15	85
Intersection *vs* Union									
Image Segmentation	86.1	82.8	78.0	94.3	47.5	73.4	39.5	71.65	
Semantic Model	83.8	74.3	63.5	96.3	45.2	68.4	28.9	65.7	

TABLE II: Semantic Evaluation: Pixel-wise percentage accuracy on the test set, $\textit{Recall}=\frac{\text{True Positive}}{\text{True Positive + False Negative}}$, *Intersection* vs *Union*$=\frac{\text{True Positive}}{\text{True Positive + False Negative + False Positive}}$

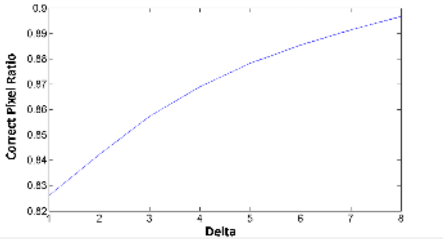

Fig. 10: Model Depth Evaluation

classified, and 'Average' is the average of the per class measures. In this evaluation we have not considered the class 'sky' which is not captured in the model. Due to lack of test data we have also not included the classes 'pedestrian' and 'signage' in our evaluation. As expected, after back-projection the classification accuracy for the model reduces due to errors in camera estimate, when compared with street image segmentation results. This would especially affect the thin object classes like 'poles/posts' where small error in projection leads to large errors in the evaluation. Classes like 'vegetation', where the surface measurement tends to be noisy, have increased error in classification. Our system is designed to model static objects in the scene, which causes an adverse effect when considering moving objects such as cars which is reflected in the results. To evaluate the accuracy of the structure, we use the ground truth depth measurement from Velodyne lasers as provided in [8]. The depth measurements from both the Velodyne lasers (δ_i^g) and our generated model (δ_i) are projected back into the image and evaluated. We measure the number of pixels that satisfy $|\delta_i - \delta_i^g| \geq \delta$, where δ is the allowed error in pixels. The results of our method are shown in Fig. 10. δ ranges between 1 to 8 pixels. It can be noted that the estimated structural accuracy at $\delta = 5$ pixels is around 88% which indicates the performance of the structure estimation.

IV. CONCLUSION

We have presented a novel computer vision-based system for 3D semantic modelling and reconstruction of urban environments. The input to our system is a stereo video feed from a moving vehicle. Our system robustly tracks the camera poses which are used to fuse the stereo depth-maps into a TSDF volume. The iso-surface in the TSDF space corresponding to the scene model is then augmented with semantic labels. This is done by fusing CRF-based semantic inference results using the input frames. We have demonstrated desirable results both qualitatively and quantitatively on a large urban sequence from the KITTI dataset [8]. In future we would like to perform semantic labelling and reconstruction jointly, where we would like to exploit the depth while performing object labelling. We believe this will improve the overall performance of our system.

ACKNOWLEDGMENTS

The authors wish to thank David Capel, Paul Sturgess and Vibhav Vineet for their valuable input. P. H. S. Torr is in receipt of Royal Society Wolfson Research Merit Award.

REFERENCES

[1] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE RAM*, 13(3):108–117, 2006.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE PAMI*, 23:2001, 2001.

[3] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV (1)*, pages 44–57, 2008.

[4] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, pages 303–312, New York, NY, USA, 1996. ACM.

[5] H. Dahlkamp, G. Bradski, A. Kaehler, D. Stavens, and S. Thrun. Self-supervised monocular road detection in desert terrain. In *RSS*, Philadelphia, 2006.

[6] B. Douillard, D. Fox, F. Ramos, and H. Durrant-Whyte. Classification and semantic mapping of urban environments. *IJRR*.

[7] F. Erbs, U. Franke, and B. Schwarz. Stixmentation - probabilistic stixel based traffic scene labeling. In *BMVC*, 2012.

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, Providence, USA, June 2012.

[9] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IV*, pages 963 –968, june 2011.

[10] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: image understanding using qualitative geometry and mechanics. In *ECCV*, ECCV'10, pages 482–496, Berlin, Heidelberg, 2010. Springer-Verlag.

[11] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *CVPR*, pages 807–814, Washington, DC, USA, 2005.

[12] K. Konolige, E. Marder-Eppstein, and B. Marthi. Navigation in hybrid metric-topological maps. In *ICRA*, pages 3041–3047, 2011.

[13] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009.

[14] L. Ladicky, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. Torr. Joint optimisation for object class segmentation and dense stereo reconstruction. In *BMVC*, 2010.

[15] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *IV*, pages 163 –168, june 2011.

[16] R. Newcombe, S. Lovegrove, and A. Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, pages 2320 –2327, nov. 2011.

[17] B. Payne and A. Toga. Surface mapping brain functions on 3d models. In *Computer Graphics and Applications*, 1990.

[18] I. Posner, M. Cummins, and P. Newman. Fast probabilistic labeling of city maps. In *RSS*, Zurich, Switzerland, June 2008.

[19] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA*, Shanghai, China, May 9-13 2011.

[20] S. Sengupta, P. Sturgess, L. Ladicky, and P. Torr. Automatic dense visual semantic mapping from street-level imagery. In *IROS*, oct 2012.

[21] K. M. Wurm, R. Kümmerle, C. Stachniss, and W. Burgard. Improving robot navigation in structured outdoor environments by identifying vegetation from laser data. In *IROS*, pages 1217–1222, 2009.