

What Lies Behind: Recovering Hidden Shape in Dense Mapping

Michael Tanner[†]

Pedro Piniés[†]

Lina María Paz[†]

Paul Newman[†]

Abstract—In mobile robotics applications, generation of accurate static maps is encumbered by the presence of ephemeral objects such as vehicles, pedestrians, or bicycles. We propose a method to process a sequence of laser point clouds and back-fill dense surfaces into gaps caused by removing objects from the scene – a valuable tool in scenarios where resource constraints permit only *one* mapping pass in a particular region. Our method processes laser scans in a three-dimensional voxel grid using the Truncated Signed Distance Function (TSDF) and then uses a Total Variation (TV) regulariser with a Kernel Conditional Density Estimation (KCDE) “soft” data term to interpolate missing surfaces. Using four scenarios captured with a push-broom 2D laser, our technique infills approximately 20 m² of missing surface area for each removed object. Our reconstruction’s median error ranges between 5.64 cm - 9.24 cm with standard deviations between 4.57 cm - 6.08 cm.

I. INTRODUCTION AND PRIOR WORK

In robotics applications, it is vital to maintain an accurate map of the local environment. When in the presence of ephemeral objects (e.g., vehicles, pedestrians, bicycles, etc.), this map can quickly become cluttered and inconsistent. These inconsistencies make the map difficult to use.

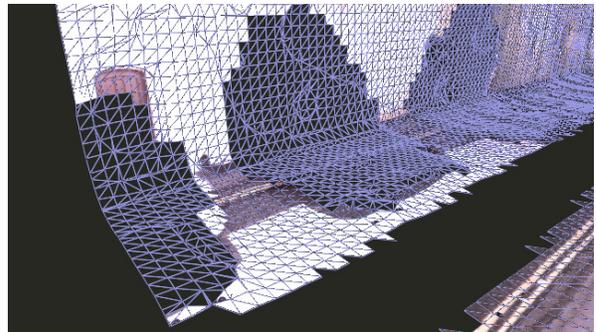
If a static map has been created from a previous data run, these difficulties can be avoided by removing ephemeral objects. Removing these objects from the map and infilling the missing background data results in an accurate static map which then enables more accurate real-time processing algorithms. For example, one can then quickly identify which laser points in the current scan are affiliated with the static map and which points are from new ephemeral objects. This is useful in distraction suppression [1] scenarios where a mask is applied to the input data to avoid bias in dependent localisation or perception algorithms. Alternatively, to complement the recent “hard negative” mining research [2], this can be used as an automated method to generate “hard positives” for standard detector or classification training.

However, removing ephemeral objects traditionally requires manually labelling positive and negative data sets to train a detector. And, once removed, it is even more difficult to back-fill the gaps (Figure 1) left in the laser data because, without a prior, there is ambiguity as to which areas should be free space and which should contain interpolated surfaces.

There has been a variety of work in this area. Wang and Oliveira [3] use a least-squares method to estimate a locally-smooth surface for holes. However, this requires user



(a) Before: The *sparse* laser point cloud input after removing ephemeral objects (automobiles). The resulting gaps are highlighted by red arrows.



(b) After: Our *dense* reconstruction with interpolated surfaces.

Fig. 1. Example scenario (a) where three large gaps appear in the laser data after vehicles are removed. These gaps must be filled in (b) to generate an accurate map which could later be used for algorithms such as localisation or ephemeral-object detection.

intervention to identify the holes and can only reconstruct simple surfaces. Vasudevan *et al.* [4] use a neural-network based Gaussian process to model large-scale, outdoor environments. They produce visually appealing results for landscape-type environments. However, this is not applicable in the city-street environments typical for our mobile robotics platform. Davis *et al.* [5] use a small, local signed distance function to represent the surface around the hole. Then, based on a number of heuristics, they select a “class model” to approximate the missing surfaces. Their algorithm was designed and tested in a small scale environment and only operates on gaps which are surrounded on all sides by distance data - the latter assumption is routinely violated in our data runs with tall ephemeral objects or when objects pass near to the laser sensor. Häne *et al.* [6] use a regularisation

[†]Mobile Robotics Group
Department of Engineering Science
University of Oxford
17 Parks Road, Oxford
OX1 3PJ, United Kingdom
mtanner,ppiniés,linapaz,pnewman@robots.ox.ac.uk

term to both smooth out noisy range data and to interpolate small holes in the dense reconstruction with the assumption that the surfaces are piecewise planar.

Much of the recent research in the field of general dense reconstruction is based on RGB or RGB-D (e.g., Microsoft Kinect) inputs [7] [8] [9] [10]. However, we seek to produce an accurate, dense model solely from sparse range-data input (laser) and then infill ephemeral-object gaps via a post-processing step.

This paper’s contribution is the formulation of a new Kernel Conditional Density Estimation (KCDE) data term for the energy functional in dense reconstruction. With no directly observed data in the interpolation regions, the KCDE makes the energy sympathetic to local structure by using a Gaussian-based likelihood model dependent upon neighbouring surfaces. This helps avoid the tendency for Total Variation (TV) to produce a saddle-shaped surface when reconstructing multiple planes.

A naive approach to missing surface reconstruction might use pure interpolation or a simple affine prior. However, the sophisticated regularisation approach we use allows us to model more complicated surfaces without explicitly enumerating all possible priors.

We begin in Section II by framing the problem in the context of an implicit 3D function: the TSDF. In Section III, we derive our regularisation method from Bayes’ theorem and outline the key equations needed for an implementation. Quantitative and qualitative analysis of our regulariser’s performance in an urban environment are presented in Section IV. Finally, we outline our conclusions in Section V.

II. VOXEL GRID STRUCTURE AND FUSION

The voxel grid model is a discretised version of a Truncated Signed Distance Function (TSDF) $f : \Omega \rightarrow \mathbb{R}$ where $\Omega \subset \mathbb{R}^3$ represents a subset of points in 3D space and f returns the corresponding truncated distance to surfaces in the scene [11]. The TSDF is constructed in such a way that zero is the surface of an object, positive values represent empty space, and negative values correspond to the occluded side of surfaces, as shown in Figure 2. Thus, by finding the zero-crossing level set, $f = 0$, we arrive at a dense representation of surfaces in a workspace.

The outputs of this fusion process are:

- 1) The weighted average TSDF value along with an associated weight (i.e., confidence) (Section II-A)
- 2) The Ω set in which the voxel belongs (Section II-B), indicating if the voxel contains valid surface data or if the surface data must be interpolated.

A. Data Fusion

Consider the case of a single laser ray with an origin at \mathbf{p}_o and a termination (i.e., reflectance) point at \mathbf{p}_r . $\vec{\mathbf{p}}$ is fused into the voxel grid by tracing the ray and updating each voxel with the signed distance to \mathbf{p}_r . Specifically, the following operations are performed on all voxels which intersect the ray from \mathbf{p}_o to \mathbf{p}_r :

- 1) Calculate the voxel’s centroid $\mathbf{p} = [x, y, z]^T$

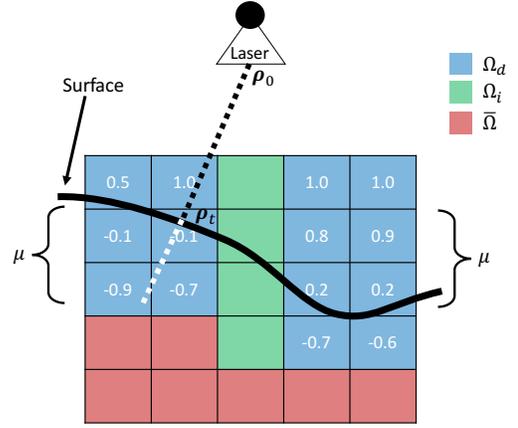


Fig. 2. A graphical depiction of how a single ray-traced laser is fused into a two-dimensional ‘voxel’ grid. The TSDF values represent the zero-crossing surface (black line) which has only been partially observed (Ω_d) while the remaining portion of the surface must be interpolated (Ω_i). All voxels with TSDF values are part in Ω_d (blue voxels), voxels near the interpolated surface are in Ω_i (green voxels), and the remaining voxels are in $\tilde{\Omega}$ (red voxels). These TSDF values $f \in [-1, 1]$ are a linear mapping from the signed distance a surface is from a given voxel centroid. There is no f value when the signed distance is less than $-\mu$, however when the signed distance is greater than μ (i.e., observed free space in front of a surface) all those voxels are updated with $f = 1$.

- 2) Evaluate v_{SDF} as the signed difference between \mathbf{p}_r and \mathbf{p} . If $v_{SDF} > 0$, the voxel is between the surface and the laser sensor, whereas $v_{SDF} < 0$ indicates the surface occludes the laser sensor’s view of the voxel.
- 3) Evaluate v_{TSDF} as a linear scale-and-clamp of v_{SDF} such that any voxel for which $v_{SDF} \geq -\mu$ lies in the interval $[-1, 1]$.
- 4) Update the voxel’s current f (TSDF value) and w (weight or confidence in the given TSDF value) at time k ,

$$w_k = \begin{cases} w_{k-1} + 1 & v_{SDF} \geq -\mu \\ w_{k-1} & v_{SDF} < -\mu \end{cases} \quad (1)$$

$$f_k = \begin{cases} \frac{v_{TSDF} + w_{k-1}f_{k-1}}{w_k} & v_{SDF} \geq -\mu \\ f_{k-1} & v_{SDF} < -\mu \end{cases}$$

where w_{k-1} and f_{k-1} are the previous values of f and w for that voxel.

These calculations are highly data-independent and thus suitable for parallel processing. However, one must ensure the memory update operations are atomic since multiple laser rays may simultaneously intersect any given voxel.

B. Ω -Domain Inclusion

Building on our approach described in [12], we use the ‘ Ω domain’ to limit regularisation to areas in which we have either directly observed laser data or where we desire to interpolate a surface. All other regions ($\tilde{\Omega}$) must be excluded to avoid spurious surface generation by the regulariser.

The Ω domain employs a principal similar to Neumann boundary conditions - i.e., carefully define and limit an operation (regularisation) by taking into account special boundary conditions for the operating domain. To leverage the Ω domain within our regulariser, we subdivide our voxel grid into three distinct sets:

- 1) Ω_d – These voxels were directly intersected by a laser ray, so the regulariser will only smooth the already-existing data.
- 2) Ω_i – These voxels were not intersected, but they are in a region in which we will interpolate missing surfaces. Since the automatic object detector [13] indicates where ephemeral objects exist, we remove the objects and add those areas to Ω_i to interpolate the surfaces. In other words, for each laser ray that intersected an ephemeral object, we extend the ray further outwards to include the background area in Ω_i .
- 3) $\bar{\Omega}$ – These voxels were neither directly intersected by a laser ray nor were they in a region targeted for interpolation. These voxels are excluded from regularisation.

Therefore, our Ω domain is defined as $\Omega = \Omega_d \cup \Omega_i$. These subsets are graphically depicted in Figure 2.

III. VARIATIONAL METHOD REGULARISATION

At its core, our method calculates the *maximum a posteriori* (MAP) estimator,

$$\operatorname{argmax}_u (P(u|f)) \quad (2)$$

with $P(u|f)$ defined as Bayes' theorem,

$$P(u|f) = \frac{P(f|u)P(u)}{P(f)} \quad (3)$$

where the scalar u is the denoised or interpolated result, f is the noisy TSDF data, $P(f)$ is a constant, $P(u)$ is the prior or “regularisation” term, and $P(f|u)$ is our likelihood model or “data” term.

From this foundation, we derive a traditional TV energy functional (Section III-A) and incorporate a KCDE to create a more sophisticated regulariser (Section III-B).

A. Bayesian Perspective

$P(u)$ and $P(f|u)$ can be modelled, respectively, as Laplace and Gaussian distributions,

$$\begin{aligned} P(u) &\propto \prod_{u_i \in \Omega} \exp\left(\frac{\|\nabla u_i\|_1}{2\sigma_u^2}\right) \\ P(f|u) &\propto \prod_{(f_i, u_i) \in \Omega} \exp\left(\frac{\|f_i - u_i\|_2^2}{2\sigma_f^2}\right) \end{aligned} \quad (4)$$

The numerator in the Laplace and Gaussian exponential's fraction are the L_1 and L_2 norms. If these distributions are substituted into Equation 2, we can reframe this as a minimising optimisation problem,

$$\begin{aligned} &\operatorname{argmin}_u (-\log(P(u|f))) = \\ &\operatorname{argmin}_u \sum_{(f_i, u_i) \in \Omega} \|\nabla u_i\|_1 + \lambda \|f_i - u_i\|_2^2 \end{aligned} \quad (5)$$

where λ encapsulates the ratio of the constant scalar values in Equations 2 and 4.

Equation 5 can be mapped to the continuous domain as a variational method's cost functional in the form [14],

$$\begin{aligned} E(u) &= E_{\text{regularisation}}(u) + E_{\text{data}}(u, f) \\ E(u) &= \int_{\Omega} \|\nabla u\|_1 d\Omega + \lambda \int_{\Omega} \|f - u\|_2^2 d\Omega \end{aligned} \quad (6)$$

where $E(u)$ is the energy (which we seek to minimise) of the denoised (u) and noisy (f) data. The *data* energy term seeks to minimise the difference between the u and f . The *regularisation* energy term, commonly known as a TV regulariser, seeks to fit the solution (u) to a specified prior. When data is non-existent, the regularisation term acts an interpolator and its form serves as a prior. This is a convex energy minimisation problem that can be solved using Primal-Dual techniques [15].

In our initial experiments, we found both the TV and Total *Global* Variation (TGV) regularisers tend to create a saddle-shaped surface when interpolating a complex junction between multiple planes. We therefore propose modifying Equation 4 with a bipartite likelihood, one computation for smoothing and another for interpolation, as explained in the next section.

B. Proposed Likelihood Function

For voxels where data is missing (i.e., Ω_i), we cannot calculate the corresponding $P(f|u)$. Our novel data term adds a KCDE likelihood estimator for the interpolated voxels (Ω_i),

$$P(f|u, v) \propto \begin{cases} \exp\left(\frac{\|f - u\|_2^2}{2\sigma_u^2}\right) & v \in \Omega_d \\ L(u) & v \in \Omega_i \end{cases} \quad (7)$$

where v is the current voxel and $L(u)$ is a KCDE which provides a prior data term to guide the regularisation as it interpolates surfaces [16],

$$\begin{aligned} L(u) &= \frac{\sum_{i=1}^M K_u(u - u_i) K_d(d_i)}{\sum_{i=1}^M K_d(d_i)} \\ K_u(\alpha) &\propto \exp\left(\frac{-\alpha^2}{2\sigma_\alpha^2}\right), \quad K_d(\beta) \propto \exp\left(\frac{-\beta^2}{2\sigma_\beta^2}\right) \end{aligned} \quad (8)$$

where M is the number of cardinal neighbouring voxels in Ω_d , u_i is the neighbour's TSDF value, and d_i is the distance between the current voxel and the neighbour. The $K_u(\cdot)$ kernel incorporates the neighbour's TSDF value while $K_d(\cdot)$, in concert with the denominator, provides a confidence in the neighbour's value based on the distance between voxels.

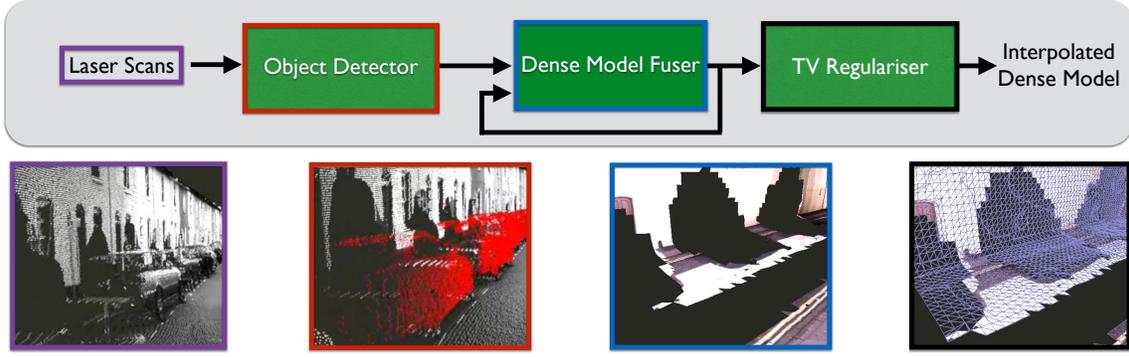


Fig. 3. Software Pipeline Overview: Our software pipeline consists of three stages to convert a sequence of laser scans into an interpolated dense model. First, we process the laser scans to automatically detect ephemeral objects in the 3D point cloud. Next, the laser rays not associated with an object are fused into a dense 3D model (Section II). Finally, we apply our KCDE-modified TV regulariser to interpolate surfaces behind the previously-removed objects (Section III-C).

This can be thought of as a memory-efficient method to accomplish a similar effect as the approach described in [17].

The minimisation version of Equation 7 now becomes,

$$\delta(u, f) = \begin{cases} \|f - u\|_2^2 & v \in \Omega_d \\ -\ln(L(u)) & v \in \Omega_i \end{cases} \quad (9)$$

which results in the energy function,

$$E(u) = \int_{\Omega} \|\nabla u\|_1 d\Omega + \lambda \int_{\Omega} \delta(u, f) d\Omega \quad (10)$$

However, the KCDE term makes $\delta(u, f)$ non-convex and cannot be solved with traditional techniques. We therefore approximate it with a second-order Taylor Series expansion which guarantees convergence when the Hessian is a positive-semidefinite matrix [18]. In addition, the non-convexity of the data term contains multiple local minima which we account for in Section III-D.

C. Implementation

Figure 3 is a summary of our overall software pipeline, outlining the steps required to process a sequence of laser scans to ultimately generate an interpolated dense model. In this section, we describe the algorithm to solve Equation 10 which is required in the last stage of our software pipeline. There are substantial steps required to derive this implementation. We provide an overview of the key concepts of our contributions and point the reader to [19] [15] [20] for additional details.

The L1 term in Equation 10 is not differentiable; therefore, it cannot be minimised using standard techniques. We use the Legendre-Fenchel Transform [20] [15] to transform it into a differentiable form,

$$\min_u \int_{\Omega} \|\nabla u\|_1 d\Omega = \min_u \max_{\|\mathbf{p}\|_{\infty} \leq 1} \int_{\Omega} u \nabla \cdot \mathbf{p} d\Omega \quad (11)$$

where the primal scalar u is the current denoised and interpolated TSDF solution and $\nabla \cdot \mathbf{p}$ is the divergence of the dual vector field \mathbf{p} defined as $\nabla \cdot \mathbf{p} = \nabla p_x + \nabla p_y + \nabla p_z$. Applying this transformation to Equation 10, the original energy minimisation problem now becomes a saddle-point

(min-max) problem with a new dual variable \mathbf{p} along with the original primal variable u ,

$$\min_u \max_{\|\mathbf{p}\|_{\infty} \leq 1} \int_{\Omega} u \nabla \cdot \mathbf{p} + \lambda \int_{\Omega} \delta(u, f) d\Omega \quad (12)$$

The solution to this regularisation problem is found with a Primal-Dual optimisation algorithm [15] which we briefly summarise in the following steps:

- 1) \mathbf{p} , u , and \hat{u} are initialised to 0. \hat{u} is a temporary variable which reduces the number of optimisation iterations required to converge.
- 2) To solve the maximisation, we update the dual variable \mathbf{p} ,

$$\mathbf{p}^k = \frac{\tilde{\mathbf{p}}}{\max(1, \|\tilde{\mathbf{p}}\|_2)} \quad (13)$$

$$\tilde{\mathbf{p}} = \mathbf{p}_{k-1} + \sigma_p \nabla \hat{u}$$

where σ_p is the dual variable's gradient-ascent step size.

- 3) For the minimisation problem, the primal variable u is updated by,

$$u_k = \begin{cases} \psi(\tilde{u}, w, f) & v \in \Omega_d \\ \xi(\tilde{u}, u_{k-1}, w, f) & v \in \Omega_i \end{cases} \quad (14)$$

with,

$$\tilde{u} = u_{k-1} - \tau \nabla \cdot \mathbf{p}$$

$$\psi(\cdot) = \frac{\tilde{u} + \tau \lambda w f}{1 + \tau \lambda w} \quad (15)$$

$$\xi(\cdot) = \frac{\tilde{u} + \tau \lambda_i (\delta''(u_{k-1}, \cdot) u_{k-1} - \delta'(u_{k-1}, \cdot))}{1 + \tau \lambda_i \delta''(u_{k-1}, \cdot)}$$

where τ is the gradient-descent step size and w is the weight of the f TSDF value.

The $\xi(\cdot)$ calculation is a numerical approximation of the second-order Taylor series expansion of $\delta(u, f)$ from Equation 9. This uses a modified λ_i which is

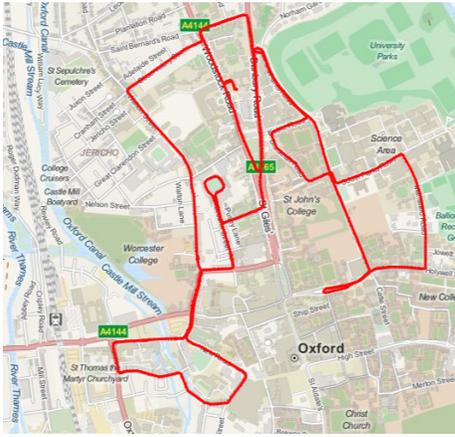


Fig. 4. The 10 km data collection route was traversed twice: once for reference with few vehicles on the roads and a second time for evaluation data with a large variety of vehicles visible. Map generated with OpenStreetMap [22].

further described in Section III-D. To meet the positive-semidefinite matrix constraint, since $\delta''(u_0, \cdot)$ is a scalar, we restrict its value to ≥ 0 .

- 4) Finally, to converge in fewer iterations, we apply a “relaxation” step,

$$\hat{u} = u + \theta(u - \hat{u}) \quad (16)$$

where θ is a parameter to adjust the relaxation step size.

While the derivation in Section III may seem intimidating at first glance, the preceding four steps are all that a user must implement to take advantage of our approach.

D. Adaptive Regularisation

As previously mentioned, the non-convexity of the KCDE data term contains multiple local minima. We initially set λ_i to a small value and then increase it after the regulariser converges.

Initially setting λ_i to a value orders of magnitude lower than λ ensures the KCDE data term only guides rather than dominates the regularisation process. This effectively removes the non-convexity of the energy functional and allows the regulariser to select solutions outside of the current local minimum. Once the regulariser converges, λ_i is increased to further refine the solution. This two-stage process minimises the global energy and therefore reduces the impact of non-convexity of the KCDE data term.

Our approach of utilising different values for λ_i , based on confidence in the local data, was inspired by image pyramids and the 2D depth-map regularisation in [21].

IV. RESULTS

To evaluate the performance of our algorithm, we ran an automated ephemeral object detector on a 10 km data-collection route in Oxford, UK, as depicted in Figure 4. Our vehicle had a Bumblebee 2 stereo camera for Visual Odometry (VO) [23]; a Ladybug 2 for colouring the laser

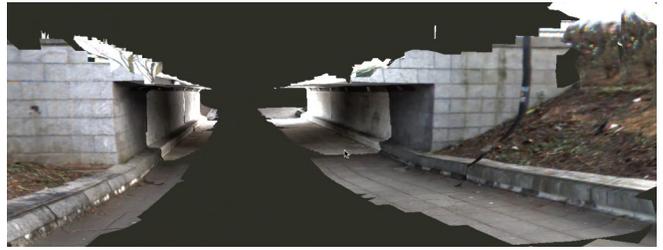


Fig. 5. An example dense reconstruction produced by fusing laser data into a voxel grid and applying texture to the surfaces from camera data.

points [24] and to add textures to the dense reconstructions; and a SICK LMS-151 2D laser oriented for push-broom collection.

The data-collection route was driven on two separate occasions. The reference data run was accomplished before dawn to minimise the number of vehicles parked on the roadside. The evaluation data run took place near noon on the same day, a time where there were a maximum number of vehicles on the roads or parked nearby.

We selected four 30 m segments from the two routes to access our dense reconstruction. The segments contained two or more vehicles in the second data run with no overlapping vehicles in the reference data run. This allowed us to automatically detect cars in the second data run, remove them, apply our dense reconstruction technique, and then compare our results with the original data run. We used the object detector described in [13].

We generated metrically consistent local 3D swathes from the 2D push-broom laser using a subset of laser-to-world pose estimates $T_{wl} \in SE(3)$ in 30 m windows,

$$M_\ell = \zeta(T_{wl}, \mathbf{x}_\ell)$$

where $\zeta(\cdot)$ is a function of the set of laser points (\mathbf{x}_ℓ) in the same 30 m interval. The resulting 3D point cloud (M_ℓ) was used as ground truth for our assessment.

From the output of our reconstruction algorithm, we extracted a point cloud sampling of the zero-crossing isosurface (e.g., Figure 5) in our voxel grid [25] and compared this to the reference data with CloudCompare [26]. These results are quantitatively summarised in Table I and Figures 6 - 9.

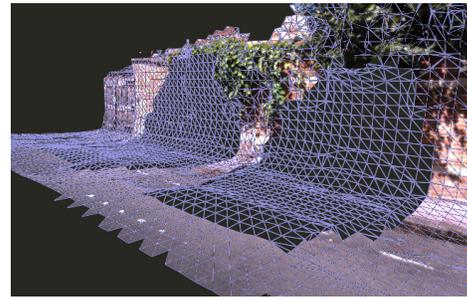
Each automobile removed required approximately 20 m² of surface area to be interpolated by the regulariser. Our reconstructions had median accuracies ranging from 5.64 cm - 9.24 cm with standard deviations between 4.57 cm - 6.08 cm.

TABLE I
SCENARIO ERROR STATICS

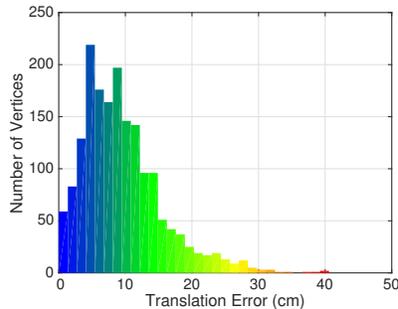
#	Median Error (cm)	σ (cm)	Eval Dims (m)	Vol. (m ³)
1	8.48	6.00	3.8 x 2.8 x 11.4	121.3
2	9.24	6.08	3.7 x 2.1 x 10.7	83.1
3	5.64	4.64	5.6 x 2.8 x 22.9	359.1
4	5.97	4.57	3.8 x 3.7 x 13.8	194.0



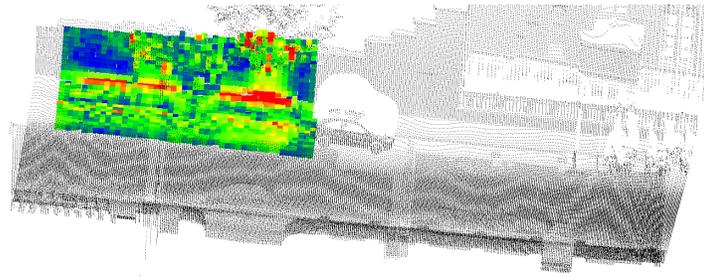
(a) The raw pointcloud used as the input to our regularisation algorithm.



(b) The dense reconstruction with infilled background data. Since there is no texture data for the obstructed background, a triangle mesh is overlaid to visualise the reconstruction's 3D model.



(c) The distribution of the translation error for the isosurface extracted from the dense reconstruction.



(d) A pointcloud of the translation errors for the dense reconstruction. Each colour in this figure correspond to the same translation error in (c).

Fig. 6. Scenario 1: Reconstruction of area behind two automobiles where the surrounding laser data includes a tree (top right of reconstructed area). Error: median = 8.48 cm, $\sigma = 6.00$ cm

The scenarios which were most challenging included complex surfaces such as trees (Figure 6), six separate intersecting planes (Figure 7), or features (e.g., door or window) obscured by the removed vehicle (Figure 9). However, a more typical city-street environment where the reconstruction is dominated by two planes (the pavement and building façade) resulted in the best reconstructions (Figure 8 and 9).

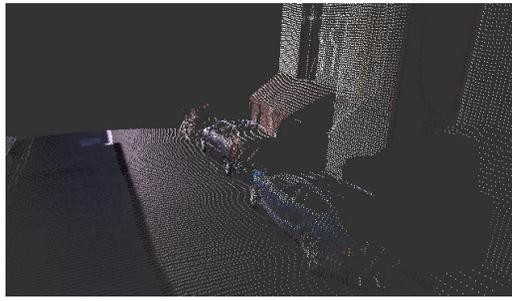
Our previous work [19] demonstrated that other regularisation terms can produce better results than TV. However, once the KCDE data term was added to our pipeline, the TV and TGV regularisers converged upon nearly identical solutions - therefore the additional computational and memory requirements of TGV are not justified.

V. CONCLUSIONS

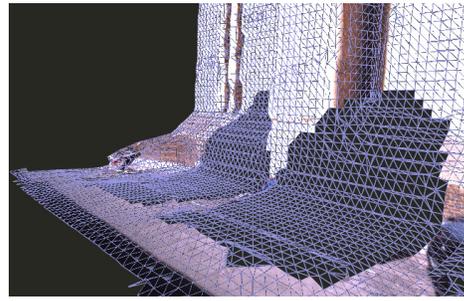
In this paper, we presented a new approach to reconstruct large-scale environments from laser scans, remove ephemeral objects, and then back-fill the gaps in the laser data. We created accurate reconstructions by fusing laser data into a voxel grid and isolating the regions for TV regularisation with a KCDE data term to assist when interpolating the missing surfaces. When reconstructing approximately 20 m² of surface area, our proposed method's median accuracy was under 10 cm with a standard deviation of 6 cm or less.

REFERENCES

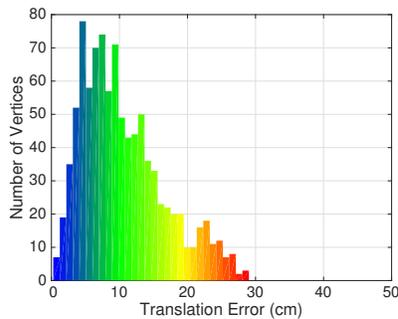
- [1] C. McManus, W. Churchill, A. Napier, B. Davis, and P. Newman, "Distraction suppression for vision-based pose estimation at city scales," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.
- [2] J. Hawke, C. Gurau, C. H. Tong, and I. Posner, "Wrong today, right tomorrow: Experience-based classification for robot perception," in *Field and Service Robotics (FSR)*, Toronto, ON, Canada, June 2015.
- [3] J. Wang and M. M. Oliveira, "A hole-filling strategy for reconstruction of smooth surfaces in range images," in *Computer Graphics and Image Processing, 2003. SIBGRAPI 2003. XVI Brazilian Symposium on*. IEEE, 2003, pp. 11–18. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1240986
- [4] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, "Gaussian process modeling of large-scale terrain," *Journal of Robot Robotics*, vol. 26, no. 10, pp. 812–840, 2009. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/rob.20309/abstract>
- [5] J. Davis, S. R. Marschner, M. Garr, and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," in *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*. IEEE, 2002, pp. 428–441. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1024098
- [6] C. Häne, C. Zach, B. Zeisl, and M. Pollefeys, "A patch prior for dense 3d reconstruction in man-made environments," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*. IEEE, 2012, pp. 563–570.
- [7] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*. IEEE, 2011, pp. 127–136. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6162880
- [8] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald, "Real-time large-scale dense RGB-d SLAM



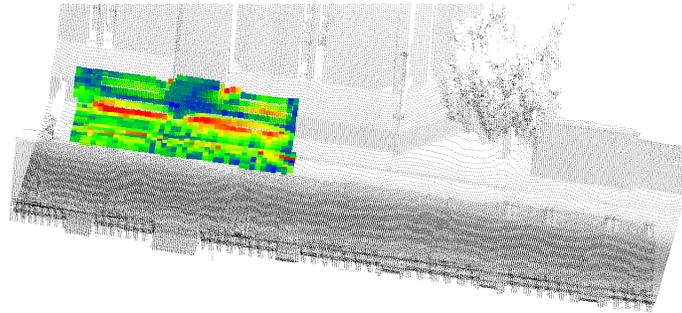
(a) The raw pointcloud used as the input to our regularisation algorithm.



(b) The dense reconstruction with infilled background data. Since there is no texture data for the obstructed background, a triangle mesh is overlaid to visualise the reconstruction's 3D model.



(c) The distribution of the translation error for the isosurface extracted from the dense reconstruction.



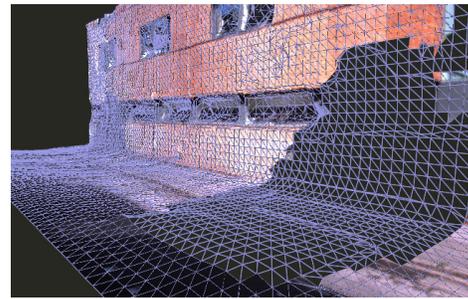
(d) A pointcloud of the translation errors for the dense reconstruction. Each colour in this figure correspond to the same translation error in (c).

Fig. 7. Scenario 2: Reconstruction of area behind two automobiles where six separate intersecting planes must be interpolated. Error: median = 9.24 cm, $\sigma = 6.08$ cm

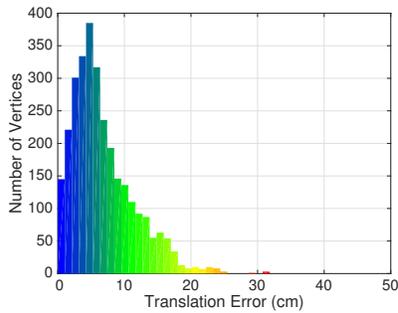
- with volumetric fusion,” *The International Journal of Robotics Research*, p. 0278364914551008, Dec. 2014. [Online]. Available: <http://ijr.sagepub.com/content/early/2014/12/07/0278364914551008>
- [9] V. Pradeep, C. Rhemann, S. Izadi, C. Zach, M. Bleyer, and S. Bathiche, “MonoFusion: Real-time 3D reconstruction of small scenes with a single web camera,” in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Oct. 2013, pp. 83–88.
- [10] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb, “Real-time 3d reconstruction in dynamic scenes using point-based fusion,” in *3DTV-Conference, 2013 International Conference on*. IEEE, 2013, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6599048
- [11] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 303–312. [Online]. Available: <http://dl.acm.org/citation.cfm?id=237269>
- [12] M. Tanner, P. Piniés, L. M. Paz, and P. Newman, “BOR2G: Building optimal regularised reconstructions with GPUs (in cubes),” in *International Conference on Field and Service Robotics (FSR)*, Toronto, ON, Canada, June 2015.
- [13] D. Wang, I. Posner, and P. Newman, “What could move? finding cars, pedestrians and bicyclists in 3D laser data,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012.
- [14] D. Cremers, “Multiple view geometry,” 2013. [Online]. Available: <http://vision.in.tum.de/teaching/ss2013/mvg2013>
- [15] A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, May 2011. [Online]. Available: <http://dx.doi.org/10.1007/s10851-010-0251-1>
- [16] M. P. Holmes, A. G. Gray, and C. L. Isbell, “Fast nonparametric conditional density estimation,” *arXiv preprint arXiv:1206.5278*, 2012. [Online]. Available: <http://arxiv.org/abs/1206.5278>
- [17] P. Piniés, L. M. Paz, and P. Newman, “Dense mono reconstruction: Living with the pain of the plain plane,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015.
- [18] D. Ferstl, G. Riegler, M. Rüther, and H. Bischof, “CP-census: A novel model for dense variational scene flow from RGB-d data,” in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.
- [19] P. Piniés, L. M. Paz, and P. Newman, “Too much TV is bad: Dense reconstruction from sparse laser with non-convex regularisation,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015.
- [20] R. T. Rockafellar, *Convex Analysis*. Princeton, New Jersey: Princeton University Press, 1970.
- [21] P. Piniés, L. M. Paz, and P. Newman, “Dense and swift mapping with monocular vision,” in *International Conference on Field and Service Robotics (FSR)*, Toronto, ON, Canada, June 2015.
- [22] “OpenStreetMap.” [Online]. Available: <http://www.openstreetmap.org>
- [23] W. Churchill and P. Newman, “Practice makes perfect? managing and leveraging visual experiences for lifelong navigation,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, USA, May 2012.
- [24] A. Stewart and P. Newman, “LAPS - localisation using appearance of prior structure: 6-DOF monocular camera localisation using prior pointclouds,” in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, USA, May 2012.
- [25] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '87. New York, NY, USA: ACM, 1987, pp. 163–169. [Online]. Available: <http://doi.acm.org/10.1145/37401.37422>
- [26] “CloudCompare.” [Online]. Available: <http://www.danielgm.net/cc/>



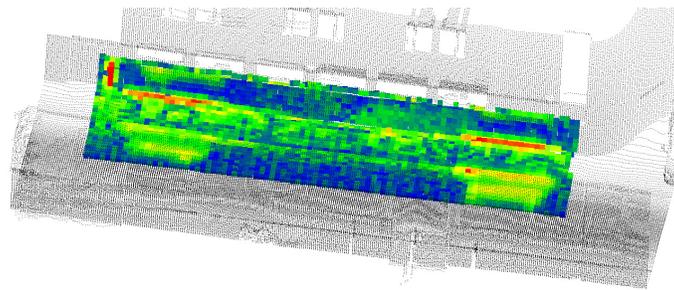
(a) The raw pointcloud used as the input to our regularisation algorithm.



(b) The dense reconstruction with infilled background data. Since there is no texture data for the obstructed background, a triangle mesh is overlaid to visualise the reconstruction's 3D model.

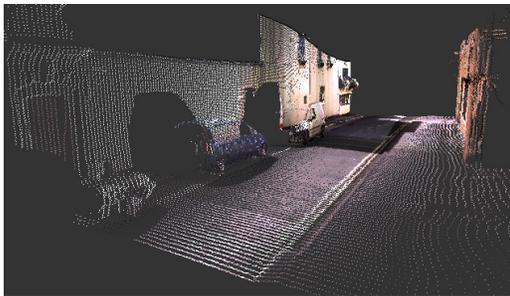


(c) The distribution of the translation error for the isosurface extracted from the dense reconstruction.

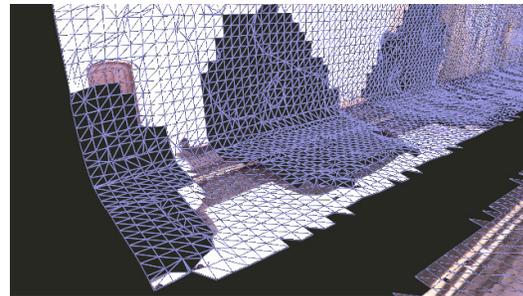


(d) A pointcloud of the translation errors for the dense reconstruction. Each colour in this figure correspond to the same translation error in (c).

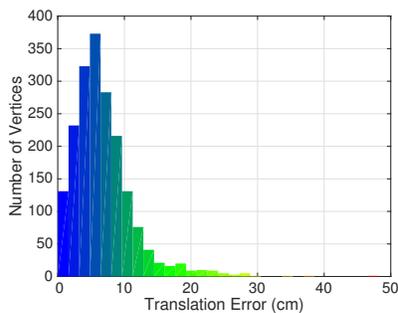
Fig. 8. Scenario 3: Reconstruction of area behind two automobiles in a typical two-plane-dominated street environment. Error: median = 5.64 cm, $\sigma = 4.64$ cm



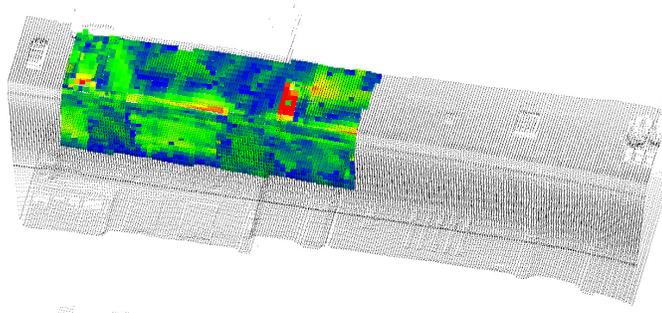
(a) The raw pointcloud used as the input to our regularisation algorithm.



(b) The dense reconstruction with infilled background data. Since there is no texture data for the obstructed background, a triangle mesh is overlaid to visualise the reconstruction's 3D model.



(c) The distribution of the translation error for the isosurface extracted from the dense reconstruction.



(d) A pointcloud of the translation errors for the dense reconstruction. Each colour in this figure correspond to the same translation error in (c).

Fig. 9. Scenario 4: Reconstruction of area behind three automobiles in a typical two-plane-dominated street environment. Error: median = 5.97 cm, $\sigma = 4.57$ cm