Too Much TV is Bad: Dense Reconstruction from Sparse Laser with Non-convex Regularisation

Pedro Piniés Lina María Paz Paul Newman

Abstract— **In this paper we address the problem of dense depth map estimation from sparse noisy range data to reconstruct large heterogeneous outdoor scenes. We propose a surface inpainting solution through energy minimisation with an adaptive selection of surface regularisers among a set of well known convex and non-convex regularisers. In fact, the selection of norm is pivotal with respect to the intrinsic surface characteristics. Our goal is to show how dense interpolation of sparse range data can be leveraged of more exotic and non-convex regularisers such as the log and logTGV [1] which can better capture the scene geometry. In contrast to state of the art solutions, we do not restrict ourselves to this set of norms, instead we search for the most apt norm for each semantically segmented part of the scene. Our energy model selection use Bayesian optimisation to learn the best choice of free parameters. This results in an adaptive model selection and the generalisation of well studied regularisation norms. We conclude with a detailed experimental analysis of our approach using a basis of four norms over a set of challenging outdoor scenes.**

I. Introduction

The task of workspace reconstruction remains a vital concern to computer vision and robotics. Given noisy, often sparse data from single camera or range images how does one construct a dense representation of the workspace which contains at once discontinuities, rips, smooth surface patches and holes. This paper is concerned with just this problem and, as is common practice, approaches it from an energy optimisation perspective. We seek a representation which balances a data term (how well does the observed data fit a solution) and a regularisation term (how well does the solution fit a prior belief on the characteristics of the surface).

The choice of the algebraic form of these two terms, driven by a choice of norm, has far reaching consequences and deserves careful attention. We show using an intuitive example on 1D signals the relative merits of a set of well studied and suitable convex and non-convex norms successfully used for image processing tasks [1]. We extend their application to a more challenging problem: dense reconstruction of large outdoor scences from a single sparse range image. A consequence of our analysis and optimisation construction is the conclusion that new more complex norms should be given a central role in dense reconstruction of large outdoor scenes - in contrast to the usual standard L1 and Huber norms successfully applied in small environments [2], [3]. The input to our pipeline consists of a point cloud obtained from push broom laser data and a corresponding set of images used to calculate the vehicle trajectory using Visual Odometry. From these, sparse depth images are obtained. Both sensors are mounted on a car that traverses a city environment.

Several alternative methods have been brought to bare on this kind of problem in the literature. In [4] we find a method to integrate high-res image data into low-res but sufficiently regular and dense range data. As a result, it recovers range data at the same resolution as the image data. The core of the method relies on an Markov Random Field formulation with a quadratic prior (or L_2 norm) which favors fronto-parallel surfaces in indoor scenes. Other approaches have been applied in the context of modeling to generate multiresolution 3D maps from sparse LIDAR measurements [5], [6], [7] or Kinect dense depth images [8]. In all these works Gaussian Process are employed. While these works share the same application and demonstrations, their models are restricted to the set of continuous and differentiable functions which prevents the explicit formulation of discontinuities.

Our task is to assign a depth for every pixel starting from the sparse seeding. Due to the heterogeneous structures that are present in the scene (buildings, vegetation, roads, walls,...), we do not apply a single energy model for the whole scene. Instead, for each distinctive region of the image, we search for a regulariser and its corresponding parameters that best captures the underlying 3D structure. Regions are acquired via an interactive multilabel segmentation of the images, as it is a common practice in many computer vision applications [9].

II. Problem Structure

Consider the example of finding a surface, parametrised by a function $x : \Omega \rightarrow \mathbb{R}$, that best explains a sparse set of noisy measurements given by a function $z : \Omega_z \rightarrow \mathbb{R}$, where $\Omega_z \subseteq \Omega$. This problem is in general ill-posed since there exist many surfaces that can fit the data. A prior belief on the surface properties (piecewise constant, smooth, affine,...) can be introduced as a regularisation term to reduce the space of valid solutions and, indeed, improve the stability of the problem. As such, we can represent our task as a minimisation over x of an energy $E(x)$ defined by

$$\min_x E(x) = \min_x R(x) + \lambda D(x, z) \quad (1)$$

where R is a regularisation prior, D is a data term capturing the compatibility between x and our measurements z and $\lambda > 0$ is a weight trading off R and D. For now we will proceed with a simple L_2 norm for D

$$D(x, z) = \frac{1}{2} \int_\Omega (x - z)^2 \quad (2)$$

which is equal to zero when $z \notin \Omega_z$ since in that case there are no measurements to compare with.

Equation 1 is called the primal form of the problem- it contains only our prime variable of interest x. If R and D were smooth and convex we could proceed with commonplace continuum methods such as Levenberg-Marquadt [10]. But if R or D are not so tame our problem is harder, yet,

Authors are from the Mobile Robotics Group, Dept. Engineering Science, University of Oxford, UK. {ppinies,linapaz,pnewman}@robots.ox.ac.uk

as this paper illustrates, the solution yielded is superior. We will now describe the optimisation machinery [11] needed to proceed.

If we wished to encourage x to be piece-wise constant we might regularise the solution by penalising the Total Variation [12] and so set

$$R_{TV}(\nabla x) = \int_{\Omega} ||\nabla x||_1 \tag{3}$$

where Ω is the domain of x and ∇x its gradient. The intuition here is that the summation over the absolute infinitesimal changes of a function gives its total change or variation. We can discretise the regularisation and data terms obtaining

$$\begin{aligned} R_{TV}(K\mathbf{x}) &= ||K\mathbf{x}||_1 \qquad (4)\\ D(\mathbf{x},\mathbf{z}) &= \frac{1}{2}||\mathbf{x}-\mathbf{z}||_2^2 \qquad (5) \end{aligned}$$

where $\mathbf{x} = [\ldots x_i \ldots]^T$ are samples of signal x taken at regular intervals of its domain Ω, operator K is a discretised version of the gradient operator and the data term is zero for $z_i \notin \Omega_z$. Notice that here we are dealing with the non-differentiable L_1 norm and therefore we cannot make use of standard optimisation methods for smooth functions. In the next section we will summarise how a solution can be obtained by using a primal-dual energy minimization scheme which makes use of the *Legendre-Fenchel Transform* and the *proximal map* operator.

A. The Legendre-Fenchel Transform

For all convex functions $f(\mathbf{x})$ there exists a one to one transformation known as the Legendre-Fenchel Transform [13],

$$f^*(\mathbf{p}) = \sup_{\mathbf{x}} \{\langle \mathbf{x}, \mathbf{p}\rangle - f(\mathbf{x})\} \tag{6}$$

that maps $f(\mathbf{x})$ to another convex function $f^*(\mathbf{p})$ known as its dual conjugate. The variable $\mathbf{p}$ is called the dual variable.

The intuition behind the dual conjugate $f^*(\mathbf{p})$ and the dual variable $\mathbf{p}$ is as follows. Assume $f(\mathbf{x})$ is smooth, since $-f(\mathbf{x})$ is concave so it is Eq.(6) and the point $\mathbf{x}_{sol}$ that maximizes it is obtained by differentiating and setting it to zero:

$$\mathbf{p} - \nabla f(\mathbf{x}_{sol}) = \mathbf{0} \tag{7}$$

Thus the dual variable $\mathbf{p}$ can be interpreted as the slope of the original function at the solution. As for $f^*(\mathbf{p})$, substituting $\mathbf{p}$ for the slope we obtain

$$f^*(\mathbf{p}) = f^*(\nabla f(\mathbf{x}_{sol})) = \langle \mathbf{x}_{sol}, \nabla f(\mathbf{x}_{sol})\rangle \ - f(\mathbf{x}_{sol}) \tag{8}$$

which is just the distance we have to shift the linear function $\langle \mathbf{x}, \mathbf{p}\rangle$ at $\mathbf{x}_{sol}$ to make it tangent to $f(\mathbf{x}_{sol})$. Therefore all pairs $(\mathbf{p}, f^*(\mathbf{p}))$ represent the set of all affine planes that are below and tangent to the function $f(\mathbf{x})$. Finally, because the Legendre-Fenchel Transform of convex functions is bijective we can also express $f(\mathbf{x})$ itself in terms of its dual

$$f(\mathbf{x}) = \sup_{\mathbf{p}} \{\langle \mathbf{x}, \mathbf{p}\rangle \ - f^*(\mathbf{p})\} \tag{9}$$

We will need these relationships in the next sections.

B. Proximal map operator

If $f(\mathbf{x})$ is a convex function, a point $\mathbf{x}^*$ is a global minimizer of f if and only if

$$\partial f(\mathbf{x}^*) \ni \mathbf{0} \tag{10}$$

where $\partial f(\mathbf{x})$ is the subgradient [14] of f at a given point $\mathbf{x}$. If $f(\mathbf{x})$ is in addition smooth,the previous condition becomes the very well known $\nabla f(\mathbf{x}^*) = \mathbf{0}$ which can be easily solved via classical iterative gradient descent with step size τ^k:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \tau^k \nabla f(\mathbf{x}^k) \tag{11}$$

Defining an operator $grad_{\tau f} \equiv (I - \tau \nabla f)$ allows us to write

$$\mathbf{x}^{k+1} = grad_{\tau f}(\mathbf{x}^k) \tag{12}$$

A more general operator which can be applied to non-smooth convex functions is the *proximal map* [13]. Given a closed convex function $f(\mathbf{x})$ its proximal map is defined by:

$$\mathbf{x} = prox_{\tau f}(\mathbf{u}) = \arg\min_{\mathbf{x}} \left\{ \frac{\|\mathbf{x}-\mathbf{u}\|^2}{2\tau} + f(\mathbf{x}) \right\} \tag{13}$$

We can equivalently express the proximal operator of f in terms of its subgradients by solving Eq.(13). Since the proximal map involves a convex minimization problem it must satisfy Eq.(10)

$$\begin{aligned} \partial_{\mathbf{x}} \left(\frac{\|\mathbf{x}-\mathbf{u}\|_2^2}{2\tau} + f(\mathbf{x}) \right) &\ni \mathbf{0} \\ \frac{\mathbf{x}-\mathbf{u}}{\tau} + \partial f(\mathbf{x}) &\ni \mathbf{0} \\ \mathbf{x} &\in (I + \tau \partial f)^{-1}(\mathbf{u}) \qquad (14) \end{aligned}$$

Noticing that the first quadratic term in Eq.(13) is a strict convex function and that f is convex by assumption, then the proximal map always produces a unique solution $\mathbf{x}$ for any input $\mathbf{u}$

$$\mathbf{x} = (I + \tau \partial f)^{-1}(\mathbf{u}) = prox_{\tau f}(\mathbf{u}) \tag{15}$$

Note that although we have now defined the proximal map operator, we have yet to fill in any of the details about how to implement it for a particular f. In particular for our case, f will be one of the several norms of interest.

C. Primal Dual optimisation

Returning to our primal problem

$$\min_{\mathbf{x}} R_{TV}(K\mathbf{x}) + \lambda D(\mathbf{x}, \mathbf{z}) \tag{16}$$

we can express R_{TV} in terms of its dual conjugate $R^\star_{TV}$, according to Eq.(9), obtaining its primal-dual form

$$\min_{\mathbf{x}} \max_{\mathbf{p}} \langle K\mathbf{x}, \mathbf{p}\rangle \ + \lambda D(\mathbf{x}, \mathbf{z}) - R^\star_{TV}(\mathbf{p}) \tag{17}$$

which is a saddle point problem in which a concave function is maximized for $\mathbf{p}$:

$$\max_{\mathbf{p}} \langle K\mathbf{x}, \mathbf{p}\rangle \ - R^\star_{TV}(\mathbf{p}) \tag{18}$$

and a convex function is minimized for $\mathbf{x}$

$$\min_{\mathbf{x}} \langle K\mathbf{x}, \mathbf{p}\rangle \ + \lambda D(\mathbf{x}, \mathbf{z}) \tag{19}$$

From Eq.(10) the following hold at the solution $(\mathbf{x}_{sol}, \mathbf{p}_{sol})$:

Algorithm 1 $\mathbf{x}$ = TV($\mathbf{z}$, K)

```
1:  {Initialization of variables:}
2:  $\lambda, \tau, \sigma > 0, \theta \in [0,\ 1]$
3:  $\mathbf{x}^0 = \mathbf{z}, \mathbf{p}^0 = 0$
4:  $\bar{\mathbf{x}}^0 = \mathbf{x}^0$
5:  while $k \leq N$ do
6:      {Update Dual}
7:      $\mathbf{p}^{k+1} = prox_{\sigma R^*_{TV}}(\mathbf{p}^k + \sigma K \bar{\mathbf{x}}^k)$
8:      {Update Primal}
9:      $\mathbf{x}^{k+1} = prox_{\lambda\tau D}(\mathbf{x}^k - \tau K^T \mathbf{p}^{k+1})$
10:     {Relaxation}
11:     $\bar{\mathbf{x}}^{k+1} = \mathbf{x}^{k+1} + \theta(\mathbf{x}^{k+1} - \mathbf{x}^k)$
12: end while
```

Algorithm 2 [x,y] = TGV($\mathbf{z}$, K, K_y)

```
1:  {Initialization of variables:}
2:  $\lambda, \alpha_1, \alpha_2, \tau_x, \tau_y, \sigma_p, \sigma_q > 0, \theta \in [0,\ 1]$
3:  $\mathbf{x}^0 = \mathbf{z}, \mathbf{p}^0 = 0, \mathbf{y}^0 = 0, \mathbf{q}^0 = 0$
4:  $\bar{\mathbf{x}}^0 = \mathbf{x}^0, \bar{\mathbf{y}}^0 = \mathbf{y}^0,$
5:  while $k \leq N$ do
6:      {Update Dual}
7:      $\mathbf{p}^{k+1} = prox_{\sigma_p R^*_{TV}}(\mathbf{p}^k + \sigma_p \alpha_1 (K\bar{\mathbf{x}}^k - \bar{\mathbf{y}}^k))$
8:      $\mathbf{q}^{k+1} = prox_{\sigma_q R^*_{TV}}(\mathbf{q}^k + \sigma_q \alpha_2 K_y \bar{\mathbf{y}}^k)$
9:      {Update Primal}
10:     $\mathbf{x}^{k+1} = prox_{\lambda\tau_x D}(\mathbf{x}^k - \tau_x \alpha_1 K^T \mathbf{p}^{k+1})$
11:     $\mathbf{y}^{k+1} = \mathbf{y}^k + \tau_y(\alpha_1 \mathbf{p}^k - \alpha_2 K_y^T \mathbf{q}^k)$
12:     {Relaxation}
13:     $\bar{\mathbf{x}}^{k+1} = \mathbf{x}^{k+1} + \theta(\mathbf{x}^{k+1} - \mathbf{x}^k)$
14:     $\bar{\mathbf{y}}^{k+1} = \mathbf{y}^{k+1} + \theta(\mathbf{y}^{k+1} - \mathbf{y}^k)$
15: end while
```

Algorithm 3 $\mathbf{x}$ = logTV($\mathbf{z}$, K)

```
1: {Initialization of variables:}
2: $\beta > 0$
3: $\mathbf{x}^0 = \mathbf{z}$
4: while $k \leq N$ do
5:     {Build $W = diag([\cdots w_i \cdots])$}
6:     $w_i = \beta/(1 + \beta||K_i x_i||_2)$
7:     $\mathbf{x}^{k+1}$ = pd_TV($\mathbf{x}^k$, $WK$)
8: end while
```

Algorithm 4 [x,y] = logTGV($\mathbf{z}$, K, K_y)

```
1: {Initialization of variables:}
2: $\beta > 0$
3: $\mathbf{x}^0 = \mathbf{z}, \mathbf{y}^0 = 0$
4: while $k \leq N$ do
5:     {Build $W$ and $W_y$}
6:     $w_i = \beta/(1 + \beta||K_i x_i - y_i||_2)$
7:     $w_i^y = \beta/(1 + \beta||K_i^y y_i||_2)$
8:     [$\mathbf{x}^{k+1}, \mathbf{y}^{k+1}$] = pd_TGV($\mathbf{x}^k$, $WK$, $W_y K_y$)
9: end while
```

differentiating Eq.(18) with respect to $\mathbf{p}$ yields

$$\begin{aligned} 0 &\in K\mathbf{x}_{sol} - \partial R^\star_{TV}(\mathbf{p}_{sol}) \\ \mathbf{p}_{sol} &\in \mathbf{p}_{sol} + \sigma K \mathbf{x}_{sol} - \sigma \partial R^\star_{TV}(\mathbf{p}_{sol}) \\ (I + \sigma\partial R^\star_{TV})(\mathbf{p}_{sol}) &\ni \mathbf{p}_{sol} + \sigma K \mathbf{x}_{sol} \\ \mathbf{p}_{sol} &= (I + \sigma \partial R^\star_{TV})^{-1}(\mathbf{p}_{sol} + \sigma K \mathbf{x}_{sol}) \\ \mathbf{p}_{sol} &= prox_{\sigma R^\star_{TV}}(\mathbf{p}_{sol} + \sigma K \mathbf{x}_{sol}) \quad (20) \end{aligned}$$

analogously, differentiating Eq.(19) with respect to $\mathbf{x}$ yields

$$\mathbf{x}_{sol} = prox_{\lambda\tau D}(\mathbf{x}_{sol} - K^T \mathbf{p}_{sol}) \quad (21)$$

Equations 20 and 21 provide the iterative primal dual TV optimisation method shown in Algorithm 1. The optimisation is assured to converge if step sizes σ and τ are properly chosen. We follow the method proposed in [15] to select them. Line 11 in the algorithm is simply a relaxation step to speed up convergence. What remains to be explained though is how to implement the proximal map operator for the relevant norms. Solving Eq.(13) for $f(\mathbf{x}) = D(\mathbf{x}, \mathbf{z})$ using the L2 norm we obtain the analytical form

$$\mathbf{x} = prox_{\lambda\tau D}(\mathbf{u}) \iff x_i = \begin{cases} \frac{u_i + \lambda\tau z_i}{1+\lambda\tau} & \text{if} \quad z_i \in \Omega_z \\ u_i & \text{if} \quad z_i \notin \Omega_z \end{cases} \quad (22)$$

In the next section we will introduce a more powerful set of regularisers and in doing so explain how to implement $prox_{\sigma R^\star_{TV}}$.

III. On more regularisers, proximal maps and their properties

In this section we will describe the proximal maps and the properties of four regularisation norms. To ease the explanation and get a clearer visual interpretation of the results obtained with each of these regularisation terms we will make use of the 1D example shown in Figure 1. Figure 1(a) shows the original ground truth signal x_{GT} using a red continuous line and the noisy measured signal z contaminated with zero mean Gaussian noise with $\sigma = 2.0$ in dashed black. For each regulariser the result of the optimisation is shown in blue. The sum of squared errors (SSE) $\sum_i (x_i - x_{GTi})^2$ is shown at the bottom of each figure. The energy minimization and the iterations required for each norm to converge are shown in Figure 2.

Total Variation (TV) / Huber (TV$_\epsilon$)

Since the L1 norm present in the TV cost is a particular case of the Huber norm $||x||_\epsilon$ defined by

$$||x||_\epsilon = \begin{cases} \frac{||x||_2^2}{2\epsilon} & \text{if} \quad ||x||_2 < \epsilon \\ ||x||_1 - \frac{\epsilon}{2} & \text{if} \quad ||x||_2 \geq \epsilon \end{cases} \quad (23)$$

when $\epsilon = 0$, we shall define the proximal map for the more general case of TV$_\epsilon$ Huber norms.

We need to obtain the Fenchel dual conjugate of R_{TV_ϵ} using Eq.(6). The result is the function

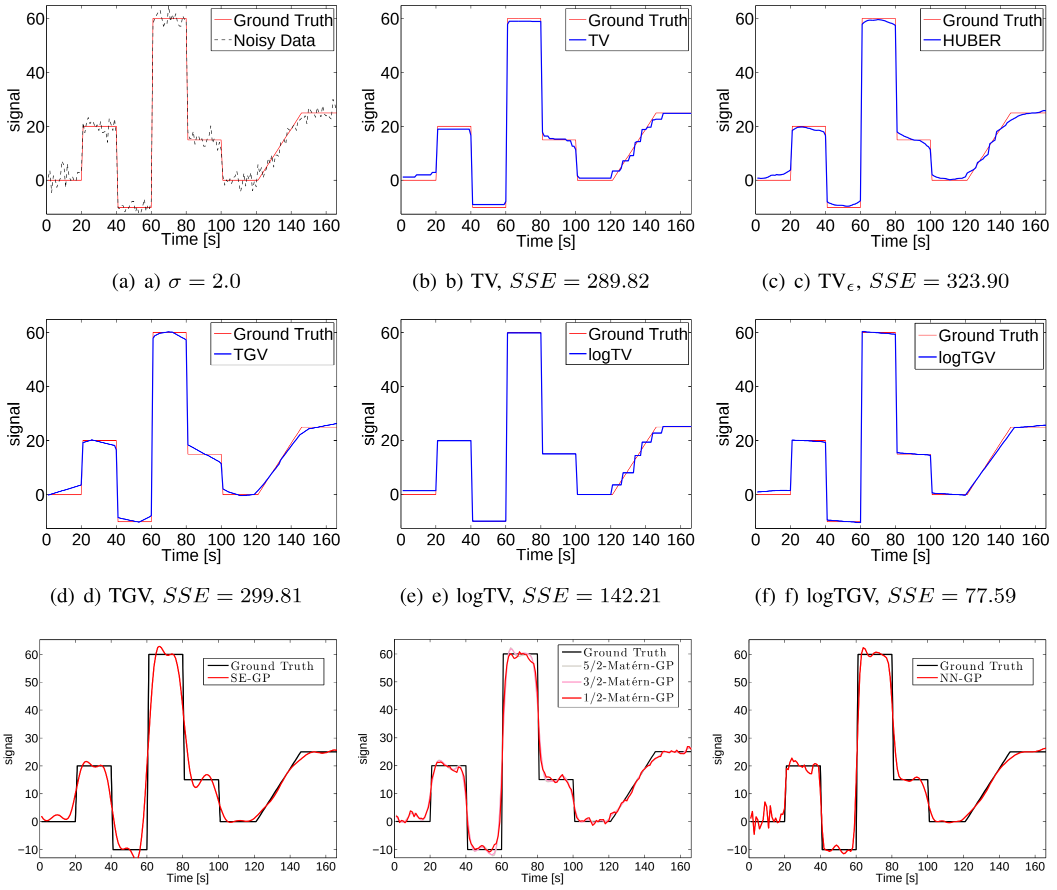
$$R^*_{TV_\epsilon}(\mathbf{p}) = \begin{cases} \frac{\epsilon}{2}||\mathbf{p}||_2^2, & ||\mathbf{p}||_\infty \leq 1 \\ \infty, & ||\mathbf{p}||_\infty > 1 \end{cases} \quad (24)$$

Notice that for $\epsilon = 0$ the dual conjugate becomes an indicator function constraining $\mathbf{p}$ to remain in the $[-1, 1]$ interval. The proximal map $prox_{\sigma R^*_{TV_\epsilon}}$ can be calculated for each element of $\mathbf{p}$ independently,

$$\mathbf{p}^{k+1} = prox_{\sigma R^*_{TV_\epsilon}}(\mathbf{u}) \iff p_i = \frac{\frac{u_i}{1+\sigma\epsilon}}{\max(1, \left|\frac{u_i}{1+\sigma\epsilon}\right|)} \quad (25)$$

If our domain is an image (2D) rather than a 1D signal then we index over (i,j) and not simply i.

TV-regularisation ($\epsilon = 0$) is known for its ability to preserve edges while smoothing homogeneous regions favouring piecewise constant solutions. This behaviour is responsible for the so-called stair casing effect shown in Figure 1(b) at the ramp section of the original function. A modest



(a) a) $\sigma = 2.0$

(b) b) TV, $SSE = 289.82$

(c) c) TV$_\epsilon$, $SSE = 323.90$

(d) d) TGV, $SSE = 299.81$

(e) e) logTV, $SSE = 142.21$

(f) f) logTGV, $SSE = 77.59$

(g) g) SE-GP, $SSE = 5162.86$

(h) h) 1/2-Matérn-GP $SSE = 2366.45$

(i) i) NN-GP, $SSE = 2616.17$

Fig. 1. This example shows the different effects induced by five convex and non-convex regularisation terms. Our goal is to recover the original form of a 1-dimensional signal, given noisy observations. a) True signal (red line) and corrupted signal (dashed black line) with added Gaussian noise. b) Total Variation keeps high variation changes while it is able to filter out the noise in the piecewise constant parts of the signal. However, it yields a stair-case effect for affine regions. c) Huber regulariser helps to prevent the stair-case effect by smoothing the signal. d) Total Generalized Variation, can naturally deal with affine regions of the signal. Unlike TV, notice that it also tends to fit linear functions to the piecewise constant parts. e) Given the non-convex nature of the logarithmic regulariser, noise effects are even more dramatically reduced in the constant regions of the signal but the stair-case effect gets amplified for the affine part. f) The joint action of the non-convex logarithmic norm and the Total Generalized Variation produces the best result due to the combination of their properties. Gaussian Process solution for three typical kernel configurations. g) Squared Exponential Kernel. h) 1/2, 3/2 and 5/2 Matérn Kernel. i) Neural Net Kernel.

improvement can be achieved in this region when $\epsilon > 0$ by combining quadratic regularisation for small changes of the function and total variation for bigger jumps. Figure 1(c) shows the result. Although the staircase effect gets slightly reduced, the quality of the recovered signal for the piecewise constant regions worsens.

Total Generalized Variation (TGV)

The total generalized variation semi-norm TGV_α^t [16] of order $t = 2$ is represented by

$$\begin{aligned} R_{TGV}(K\mathbf{x}) &= \min_{\mathbf{y}} \alpha_1 ||K\mathbf{x} - \mathbf{y}||_1 + \alpha_2 ||K_y \mathbf{y}||_1 \qquad (26) \\ &= \min_{\mathbf{y}} \alpha_1 R_{TV}(K\mathbf{x} - \mathbf{y}) + \alpha_2 R_{TV}(K_y \mathbf{y}) \end{aligned}$$

By introducing $t-1$ additional primal variables, TGV can intrinsically yield a balance between the first and higher order derivatives of the solution signal. This property allows us to generalise the piecewise constant behaviour of the TV norm and favour instead the reconstruction of piecewise polynomial functions of order $t-1$ (piecewise affine, piecewise quadratic, ...). Since the TGV regulariser is convex a global minimum can be found using a Primal Dual algorithm. Here we use a TGV regularisation of second order ($t = 2$) to properly approximate piecewise affine features such as planar surfaces found in man-made environments. The TGV_α^2 regularisation depends on the gradient of $\mathbf{x}$, on a new primal variable $\mathbf{y}$ that allows the creation of piecewise affine elements and on two constants α_1 and α_2 that control the piecewise smoothing.

Equation 26 gives us some intuition about why TGV_α^2 favours piecewise affine functions. Think of $\mathbf{y}$ as the slope of $\mathbf{x}$. If $\mathbf{x}$ is piecewise linear then $\mathbf{y}$ should be a piecewise constant signal which explains the TV $||K_y\mathbf{y}||_1$ penalty term for $\mathbf{y}$. Regarding the first term $||K\mathbf{x} - \mathbf{y}||_1$, if $\mathbf{y}$ properly estimates the slope of an affine region of $\mathbf{x}$ then the contribution to the energy cost will be zero because $K\mathbf{x} = \mathbf{y}$

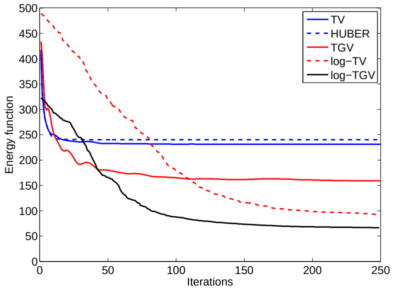


Fig. 2. Energy evolution for the 1D example. Each plot corresponds to a different regulariser term applied to the problem. Therefore, initial and minimum energy values do not have to coincide across models. In all cases, less than 250 cheap iterations are required to achieve a minimum. Thanks to their intrinsic convex properties, Total Variation (blue continuous line), Huber (dashed blue line) and the second order Total Generalized Variation (red continuous line) regularisers yield their minimum energy faster than the non-convex logTV (red dashed line) and logTGV (black line) norms.

and the only cost due to the penalty term will be the TV of $\mathbf{y}$.

Since R_{TGV} can be expressed in terms of R_{TV} we can swiftly derive the dual conjugate for each summand

$$\begin{aligned} R_{TGV}(K\mathbf{x}) &= \min_{\mathbf{y}}\{\max_{\mathbf{p}} \alpha_1\langle K\mathbf{x}-\mathbf{y},\mathbf{p}\rangle - R^{\star}_{TV}(\mathbf{p}) \\ &+ \max_{\mathbf{q}} \alpha_2\langle K_y\mathbf{y},\mathbf{q}\rangle - R^{\star}_{TV}(\mathbf{q})\} \qquad (27) \end{aligned}$$

where $\mathbf{p}$ is the dual variable for $\mathbf{x}$ and $\mathbf{q}$ is the dual variable for the new primal $\mathbf{y}$. Both lines in Eq.(27) have the same structure as Eq.(18) and so the proximal maps for the dual variables $\mathbf{p}$ and $\mathbf{q}$ are the same as the one calculated for the TV norm Eq.(25). Since Eq.(27) is differentiable with respect to the new primal variable, $\mathbf{y}$ can be updated using a gradient descent step. Algorithm 2 shows the Primal Dual optimization for the TGV.

In Figure 1(d) we can see the results obtained with the TGV norm for the 1D example. Notice that, as was expected, the ramp section is now much better recovered. However, piecewise constant regions get also approximated by affine functions whose slope depend on the mean behaviour of the noisy signal. For example, for time intervals $[0s, 20s]$ and $[140s, 160s]$ the mean value of $\mathbf{z}$ tends to grow and so does the solution $\mathbf{x}$. Again for intervals $[20s, 40s]$ and $[80s, 100s]$ the mean value of the noisy signal decreases and the slope of the solution becomes also negative.

Logarithmic TV (logTV)

The logarithmic TV norm is given by

$$R_{logTV}(K\mathbf{x}) = \log(1+\beta||K\mathbf{x}||_1) \qquad (28)$$

which is a non-differentiable and non-convex norm. The benefits of using a non-convex norm are justified from different perspectives in the literature of robust statistics [17] and natural image statistics [18]. In [1] the authors present a generalized iteratively re-weighted L1 (IRL1) algorithm to efficiently minimize a non-smooth and non-convex L_n pseudo-norm where $n \leq 1$. The idea is to rewrite the original function as the infimum over a family of L1 functions and then solve using an iterative re-weighted L1 algorithm. This means that, at each iteration j, instead of solving Eq.(28) we solve

$$\log(1+\beta||K\mathbf{x}^j||_1) \Longrightarrow ||W^j K\mathbf{x}^j||_1 \qquad (29)$$

where $W^j = diag([\cdots w_i^j \cdots])$ and $w_i^j = \beta/(1+\beta|K_i x_i^j|)$ and therefore we end up iteratively solving a primal dual TV problem. The structure of this process is shown in Algorithm 3 where an outer loop iterating over W^j surrounds an inner loop which is a standard TV solver.

The results obtained in the 1D example for the logTV are shown in Figure 1(e). Notice that this norm produces the best results for the piecewise constant section while the staircase effect is amplified. The reason for this behaviour is the peaky shape of the logarithmic function near the origin and flat tails far from it. The logTV norm is reduced more by anulling small bumps in the function (recall K is the derivative operator) where the gradient of the logarithm is steep than big jumps that appear in its flat tails (see all $20s$ wide piecewise constant intervals before $120s$).

Logarithmic TGV (logTGV)

We are now able to describe the final norm of interest. The logarithmic TGV is given by the following expression

$$\begin{aligned} R_{logTGV}(K\mathbf{x}) &= \alpha_1\log(1+\beta||K\mathbf{x}-\mathbf{y}||_1) \qquad (30) \\ &+ \alpha_2\log(1+\beta||K_y\mathbf{y}||_1) \qquad (31) \end{aligned}$$

and is solved in much the same was the logTV, we solve at each iteration j a L1 approximation given by

$$\alpha_1||W^j K\mathbf{x}^j - \mathbf{y}^j||_1) + \alpha_2||W_y^j K_y\mathbf{y}^j||_1) \qquad (32)$$

i.e., we iteratively solve a weighted TGV problem.

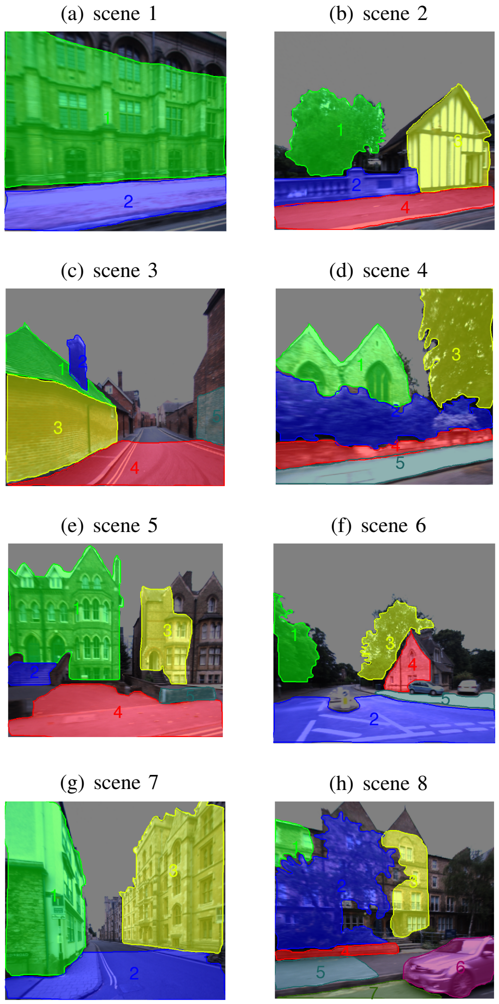
Note how the logTGV norm produces the best results for the 1D example in Figure 1(f). It is able to combine the best properties of the TGV for the affine regions while maintaining sudden discontinuities thanks to the log properties. We will now put these techniques to work in a robotics context - dense reconstruction from sparse laser data.

IV. Dense map generation from sparse scenes

While a sparse map provides a compact representation for autonomous navigation, higher level robot tasks can require denser maps to improve scene understanding. This section explains how we exploit the recently discussed norms to generate 3D dense representations of outdoor scenes from sparse laser data. The input to our pipeline consists of a point cloud obtained from push broom laser data and a corresponding set of images gathered by a camera. Both sensors are mounted on a car that traverses a city environment. For a given image, a sparse depth map $\mathbf{z}$ is obtained by assigning to a sparse set of pixels the depth of a 3D laser point that falls in the field of view of the camera. Our task is to assign a depth for every pixel starting from the sparse seeding. Due to the heterogeneous structures that are present in the scene (buildings, vegetation, roads, walls,...), we do not apply a single energy model for the whole scene. Instead, for each distinctive region of the image, we search for a regulariser and its corresponding parameters that best captures the underlying 3D structure. Regions are acquired via an interactive multilabel segmentation of the images, as it is a common practice in many computer vision applications [9]. Figure 3 shows the segmented RGB image (left) and the sparse projected map (right) for one of the gathered scenes.

TABLE I

PRECISION GAIN W.R.T STANDARD TV REGULARISATION


(a) scene 1 (b) scene 2 (c) scene 3 (d) scene 4 (e) scene 5 (f) scene 6 (g) scene 7 (h) scene 8

Scene 1							
S_l	TV	logTV	TGV	logTGV	Min range	Max range	Gain
1	0.0530	0.0230	0.0523	**0.0117**	7.2299	13.2393	77.92 %
2	0.1122	0.0496	0.0686	**0.0233**	2.6110	10.4195	79.23 %
Scene 2							
S_l	TV	logTV	TGV	logTGV	Min range	Max range	Gain
1	0.1754	0.1238	0.1931	**0.0867**	7.3390	24.9605	50.57 %
2	0.0305	0.0205	0.0216	**0.0075**	4.5311	6.8861	75.41 %
3	0.0198	0.0126	0.0228	**0.0108**	6.1705	10.4076	45.45 %
4	0.1268	0.1003	0.0474	**0.0113**	2.9398	7.8697	91.09 %
Scene 3							
S_l	TV	logTV	TGV	logTGV	Min range	Max range	Gain
1	0.0525	0.0365	0.0491	**0.0212**	2.7486	10.6596	59.62 %
2	0.0081	0.0066	0.0068	**0.0060**	4.9034	5.5492	25.93 %
3	0.0161	0.0126	0.0152	**0.0106**	2.8364	10.3617	34.16 %
4	0.0195	0.0208	0.0162	**0.0149**	2.8427	10.1468	23.59 %
5	0.0014	**0.0011**	0.0057	0.0031	7.2601	9.0595	21.43 %
Scene 4							
S_l	TV	logTV	TGV	logTGV	Min range	Max range	Gain
1	0.0444	0.0129	0.0595	**0.0079**	3.8445	24.5059	82.21 %
2	0.0810	0.0491	0.0724	**0.0409**	3.2909	15.9772	49.51 %
3	0.1910	0.1475	0.1856	**0.1064**	6.7509	14.5637	44.29 %
4	0.0236	0.0148	0.0170	**0.0096**	3.5569	7.8554	59.32 %
5	0.1382	0.0686	0.0223	**0.0089**	2.9013	6.9222	93.56 %
Scene 5							
S_l	TV	logTV	TGV	logTGV	Min range	Max range	Gain
1	0.0707	0.0287	0.0502	**0.0135**	9.0210	16.8020	80.91 %
2	0.0570	0.0311	0.0741	**0.0162**	8.6941	12.6879	71.58 %
3	0.0141	0.0094	0.0091	**0.0064**	14.8289	17.8515	54.61 %
4	0.1367	0.0401	0.0488	**0.0076**	2.8503	10.4670	94.44 %
5	0.0174	0.0152	0.0152	**0.0112**	5.4886	9.5690	35.63 %
Scene 6							
S_l	TV	logTV	TGV	logTGV	Min range	Max range	Gain
1	0.2367	0.0951	0.2982	**0.0591**	12.1138	34.4584	75.03 %
2	0.1874	0.0598	0.0822	**0.0150**	3.0169	14.9666	92.00 %
3	0.5061	0.3340	0.3360	**0.1890**	22.5005	38.4495	62.66 %
4	0.0341	0.0155	0.0285	**0.0117**	15.0562	20.6384	65.69 %
5	1.0200	0.9103	0.2730	**0.0345**	6.7719	18.5484	96.62 %
Scene 7							
S_l	TV	logTV	TGV	logTGV	Min range	Max range	Gain
1	0.0166	0.0093	0.0094	**0.0081**	4.3923	20.0294	51.20 %
2	0.0347	0.0146	0.0153	**0.0094**	2.7525	13.5629	72.91 %
3	0.3076	0.0528	0.1427	**0.0252**	7.3418	42.4991	91.81 %
Scene 8							
S_l	TV	logTV	TGV	logTGV	Min range	Max range	Gain
1	0.0270	0.0136	0.2225	**0.0121**	6.1231	13.6918	55.19 %
2	0.1804	0.1224	0.1635	**0.0743**	4.4674	13.8613	58.81 %
3	0.0446	0.0286	0.0461	**0.0119**	7.7731	18.8951	73.32 %
4	0.0273	0.0125	0.0140	**0.0093**	6.4383	8.3789	65.93 %
5	0.2272	0.2030	0.0891	**0.0126**	3.4327	7.3482	94.45 %
6	0.0513	0.0190	0.0216	**0.0150**	2.7488	11.6562	70.76 %
7	0.1640	0.1411	0.1557	**0.0154**	2.9164	4.2481	90.61 %

Left, set of selected outdoor scenes. Right, Evaluation of costs per segment/scene. First column, index of segmented region. Second-Fifth columns, median of the computed errors. We use Bayesian optimisation to learn the best parameters per segmented region which minimize the energy model according to the four different regularisation terms (TV, logTV, TGV, and the proposed logTGV). Sixth-Seventh columns, minimum and maximum range values obtained from the laser for the corresponding segment. Eighth column, gain obtained for the Best regularizer (lowest median error) that is registered (green) with respect to the TV median error $(TV - Best)/TV$. All units are in meters.

A. Learning Energy model parameters using Bayesian optimisation

One of the most important aspects that we encounter during the minimization of any of the presented energy models is their parameter dependence. Fortunately, some of the parameters required, like the proximal map steps σ, τ in the primal dual algorithm, only depend on the inner structure of the discrete gradient operator K. It has been shown in [15] that a good general choice for the steps can be calculated using diagonal preconditioning. The advantages of the preconditioning are three-fold: First, it reduces the number of iterations required to converge to the optimal solution. Second, as K is a sparse matrix, the calculation of the preconditioners only requires fast sparse computations and third, steps are calculated only once and remain constant for all iterations.

Unfortunately, there are few guidelines regarding the values that could be used for the regularisation parameters, i.e. the trade-off λ between the regulariser and the data term, the relation α_1 and α_2 between the primal variables for the TGV model or the value for the β parameter in the non-convex logarithmic priors. To estimate these parameters, we have followed a Bayesian Optimisation (BO) approach, since it provides an elegant Bayesian strategy to optimize unknown black-box objectives by using a Gaussian Process (GP) to learn the underlying function. After each function evaluation, the GP is updated, forms a new posterior distribution over the objective function and determines what the next query point should be. This automatic approach allows us to optimize the performance of each of the regularisation models with

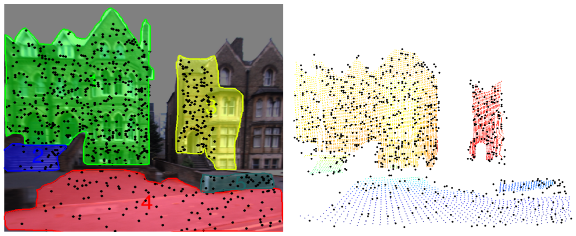


Fig. 3. Example of a large scale outdoor scene as input data for the 3D densification problem. Left, multilabel segmentation S_l of the RGB image. Right, the resulting sparse projected map (colored points). We extract 10 % of the measured sparse points per segmented region (black dots) as test data $\mathbf{z}_l$ to learn the energy parameters $\mathcal{P}_{R_*}$ for a given regulariser through the GP-EI-MCMC Bayesian Optimisation.

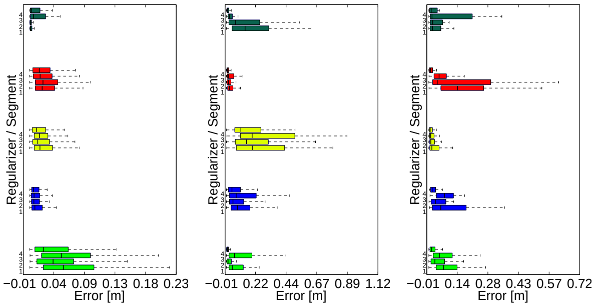
respect to their parameters while avoiding any dark magic/art that relies on expert experience, rules of thumb, or brute-force search. We use the GP-EI-MCMC method proposed in [19] which uses Markov Chain Monte Carlo (MCMC) to estimate and maximize the Expected Improvement (EI) over the current best parameter solution.

Our black-box cost function is defined as follows. For each segment S_l in a sparse depth map $\mathbf{z}$, 10% of the points $\mathbf{z}_l$ are extracted as test data and the remaining points $S_l \setminus \mathbf{z}_l$ are used as input for the densification. For a given regulariser R_* and its parameters $\mathcal{P}_{R_*}$ we run the primal dual optimisation until convergence and we calculate the errors between the extracted test samples $\mathbf{z}_l$ and their corresponding interpolated points $\mathbf{x}_l$. The cost measure for $\mathcal{P}_{R_*}$ is then given by the median of the errors

$$cost(\mathcal{P}_{R_*}) = median(\mathbf{z}_l - \mathbf{x}_l) \tag{33}$$

Figure 3 shows the sampled set $\mathbf{z}_l$ (black dots) per segment S_l used during the BO evaluation of one of the sparse depth maps.

V. Results



(a) scene 3 (b) scene 4 (c) scene 5

Fig. 4. Analysis of absolute errors on 5 of the scenes. Box plots are obtained for each model (1-TV, 2-TGV, 3-logTV, 4-logTGV) applied per segment for scenes (a)-(c). Colors match segments according to table I. The tops and bottoms of each boxplot are the 25th and 75th percentiles of the samples, respectively. The line in the middle of each box is the error median. In most of the cases the median values fall below tens of millimeters. Affine planes and further objects promote more accurate results for TGV and logTGV regularisation. In contrast, stairs and trees marginally benefit from non convex norms.

We carefully selected eight outdoor scenes and ran multilabel interactive segmentation with the number of labels ranging into the interval $n_l = 2 \ldots 8$. To generate a 3D dense map, we carried out densification for each segment with four of the energy models described previously (since TV and Huber produce very similar outcomes): i.e we perform energy minimization with TV, TGV convex priors and logTV, logTGV non-convex priors.

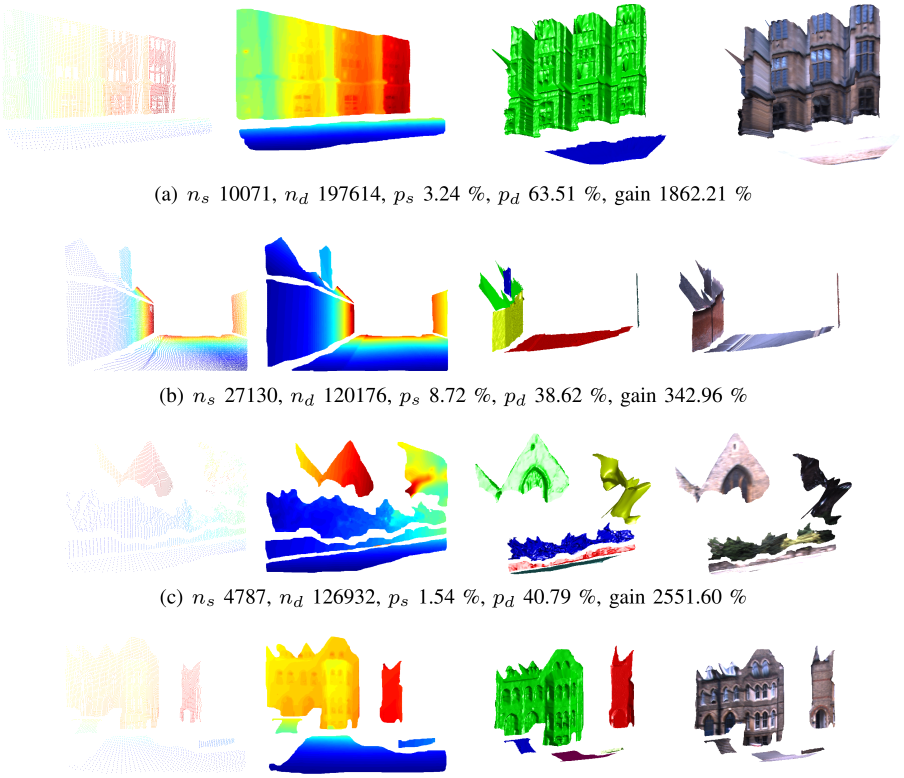
To find the best parameter values in terms of accuracy that minimizes each energy, we run 50 iterations of GP-EI-MCMC per energy model on each region, which gives rise to 9600 of total iterations ($\sum n_l \times 4 \times 50$) for the whole dataset. Notice that the dimension of the parameter search space will differ from one model to other: for TV, the GP-EI-MCMC optimizes on $\mathbb{R}$ (λ); TGV induces a search on the $\mathbb{R}^3$ domain (λ, α_1, α_2); logTV a search on $\mathbb{R}^2$ (λ,β); and logTGV induces a search on $\mathbb{R}^4$ (λ, α_1, α_2, β).

Following the definition in Eq. 33, table I summarizes the cost values for the best parameters attained once the Bayesian optimisation is executed. The worst (red) and best (green) costs among the energy models are highlighted in the table showing that, in practically all the cases, the logTGV regularisation prior achieves the lowest values (for the two cases where it does not the cost is in fact very close to the best one). In contrast, the TV prior usually provides the highest cost. The results reported in the table demonstrate the adaptive behaviour of logTGV for heterogeneous outdoor scenes. For instance, consider segment 1 in both scenes 2 and 6 for which TGV alone yields the worst cost. Looking at their corresponding RGB images, these results are understandable since they do not correspond to affine structures but vegetation. However, the combination with the nonconvex logarithmic function, logTGV, allows the system to capture vegetation irregularity. This observation holds also for segments with a high level of detail in their structure as shown in scene 10. In general, purely flat segments are well suited to the use of TGV or logTGV (see the cases of segments representing roads or sidewalks). We have also computed the percentage gain of the lowest prior cost (basically logTGV) with respect to the standard TV approach. Our results show a vast improvement (more than 50% of gain on average) when introducing the logTGV norm.

We expand the analysis above to study the accuracy of the dense interpolation. For compactness of the results, Figure 4 depicts the boxplots of the error distributions. In general, for each segment and each regularised energy model, the mass (and the median) of the distribution collapses towards its 25th percentile ranging between tens of millimeters to approx. 0.5m for very large depths. Note in particular that for each region in which logTGV achieved the best cost (see table I), the accuracy is significantly improved in contrast to the other regularisation options. Finally, figure 5 shows the 3D dense interpolation for five (dictated by space constraints) of the ten datasets.

VI. Conclusions and Summary

In this paper we have made the case that the TV norm, despite its widespread use in dense reconstruction, has far inferior performance when compared to the logTGV norm. As we suspected, too much TV is bad. We have carefully and for the first time in one place, described how to implement this non-convex norm and provided clear and concise pseudo-code with the aim of making this powerful technique accessible to the robotics community's attention. We have



(a) n_s 10071, n_d 197614, p_s 3.24 %, p_d 63.51 %, gain 1862.21 %

(b) n_s 27130, n_d 120176, p_s 8.72 %, p_d 38.62 %, gain 342.96 %

(c) n_s 4787, n_d 126932, p_s 1.54 %, p_d 40.79 %, gain 2551.60 %

(d) n_s 6745, n_d 138294, p_s 2.17 %, p_d 44.44 %, gain 1950.32 %

Fig. 5. Dense reconstruction of five outdoor scenes: a) scene 1 , b) scene 2, c) scene 3, d) scene 4, e) scene 5. First column, Sparse depth map obtained after the projection of the laser scan on the corresponding image frame. Second column, dense depth map obtained after optimisation with the best regulariser learned per region according to table I. Third column, ray tracing of the 3D dense reconstruction per region. Fourth column, the textured 3D reconstruction. For each case, we calculate the percentage of points in the sparse (p_d) and dense (p_d) maps w.r.t the total number of pixels in the image. Also, we show the gain of densification $(n_d - n_s)/n_s$ (expressed in %), where n_d is the number of pixels in the dense map and n_s the initial number of pixels in the sparse map.

applied logTGV normalisation to a problem well known to the robotics community - outdoor dense reconstruction from sparse laser data - and presented detailed results. We show that the logTGV norm, is on average twice as good at capturing scene structure as the common-place TV norm. Finally we addressed the little spoken of problem of parameter selection by invoking Bayesian Optimisation which learns the correlations between interacting parameters on a per-segment basis.

References

[1] P. Ochs, A. Dosovitskiy, T. Brox, and T. Pock, "An iterated L1 algorithm for non-smooth non-convex optimization in computer vision," in *IEEE CVPR 2013*, Los Alamitos, CA, USA, 2010, pp. 1759–1766.

[2] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *In DAGM 2010*, Darmstadt, Germany, September 2010, pp. 11–20.

[3] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *ICCV 2011*, Washington, DC, USA, pp. 2320–2327.

[4] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," in *NIPS 2005*. Cambridge, MA: MIT Press.

[5] C. Plagemann, K. Kersting, and W. Burgard, "Nonstationary Gaussian Process Regression Using Point Estimates of Local Smoothness," in *ECML PKDD - Part II 2008*. Berlin, Heidelberg: Springer-Verlag, pp. 204–219.

[6] R. F. N. E. Vasudevan, S. and Durrant-Whyte, "Gaussian process modeling of large-scale terrain," *Int. Journal of Field Robotics*, vol. 26, no. 10, p. 812–840., Sep. 2009.

[7] M. P. Gerardo-Castro, T. Peynot, and F. Ramos, "Laser-radar data fusion with Gaussian Process Implicit Surfaces," in *The 9th Int. Conf. on Field and Service Robotics*, Brisbane, Australia, 2013.

[8] S. Kim and J. Kim, "Occupancy Mapping and Surface Reconstruction Using Local Gaussian Processes With Kinect Sensors," *IEEE T. Cybernetics*, vol. 43, no. 5, pp. 1335–1346, 2013.

[9] C. Nieuwenhuis and D. Cremers, "Spatially varying color distributions for interactive multi-label segmentation," *IEEE PAMI*, vol. 35, no. 5, pp. 1234–1247, 2013.

[10] J. Nocedal and S. Wright, *Numerical Optimization*. New York: Springer-Verlag, 1999.

[11] A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, May 2011.

[12] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," in *Proc. of the 11th annual Int. Conf. of the Center for Nonlinear Studies on Experimental mathematics : computational issues in nonlinear science*. Elsevier North-Holland, Inc., 1992, pp. 259–268.

[13] R. T. Rockafellar, *Convex Analysis*. Princeton, New Jersey: Princeton University Press, 1970.

[14] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005.

[15] T. Pock and A. Chambolle, "Diagonal preconditioning for first order primal-dual algorithms in convex optimization," in *IEEE Int. Conf. on Computer Vision (ICCV)*, 2011, pp. 1762–1769.

[16] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM J. Img. Sci.*, vol. 3, no. 3, pp. 492–526, Sep. 2010.

[17] M. J. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 57–91, 1996.

[18] J. Huang and D. Mumford, "Statistics of natural images and models," in *IEEE CVPR 1999*, vol. 1, pp. –547 Vol. 1.

[19] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Neural Information Processing Systems*, 2012.