

## Article

# Fast Sparse Coding for Range Data Denoising with Sparse Ridges Constraint

Zhi Gao <sup>1,\*</sup> , Mingjie Lao <sup>1</sup>, Yongsheng Sang <sup>2</sup>, Fei Wen <sup>3</sup>, Bharath Ramesh <sup>1</sup>   
and Ruifang Zhai <sup>4</sup>

<sup>1</sup> Temasek Laboratories, National University of Singapore, 117411 Singapore; tsllaom@nus.edu.sg (M.L.); tsrame@nus.edu.sg (B.R.)

<sup>2</sup> College of Computer Science, Sichuan University, Chengdu 610065, China; sangys@scu.edu.cn

<sup>3</sup> School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China; wenfei@whu.edu.cn

<sup>4</sup> College of Informatics, Huazhong Agricultural University, Wuhan 430070, China; rfzhai@mail.hzau.edu.cn

\* Correspondence: gaozhinus@gmail.com; Tel.: +65-977-091-85

Received: 16 March 2018; Accepted: 5 May 2018; Published: 6 May 2018



**Abstract:** Light detection and ranging (LiDAR) sensors have been widely deployed on intelligent systems such as unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs) to perform localization, obstacle detection, and navigation tasks. Thus, research into range data processing with competitive performance in terms of both accuracy and efficiency has attracted increasing attention. Sparse coding has revolutionized signal processing and led to state-of-the-art performance in a variety of applications. However, dictionary learning, which plays the central role in sparse coding techniques, is computationally demanding, resulting in its limited applicability in real-time systems. In this study, we propose sparse coding algorithms with a fixed pre-learned ridge dictionary to realize range data denoising via leveraging the regularity of laser range measurements in man-made environments. Experiments on both synthesized data and real data demonstrate that our method obtains accuracy comparable to that of sophisticated sparse coding methods, but with much higher computational efficiency.

**Keywords:** LiDAR; range data denoising; sparse coding; ridge constraint

## 1. Introduction

In addition to their increasing proliferation and the growing importance of their roles in numerous applications (including 3D reconstruction and landscape surveying), light detection and ranging (LiDAR) sensors have recently been widely deployed on intelligent systems such as unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs) to perform localization, obstacle detection, and navigation tasks. Consequently, research into range data processing with competitive performance in terms of both accuracy and efficiency has attracted increasing attention. Despite the simplicity and directness in obtaining 3D information, such range data often suffer from noise due to the reflectance property of an object's surfaces, or due to electrical and mechanical disturbances. Therefore, efficient denoising for range data remains a critical problem.

Some range data denoising methods directly originate from their counterparts for images by regarding the depth value of range data as intensity, and readers can refer to [1] for detailed discussions. However, such methods typically cause artifacts of vertex drifting, which reduce the geometric regularity [2,3]. Techniques designed specifically for range data (e.g., the 3D points cloud) can be roughly classified into point-based [4,5] and mesh-based [6,7] methods. However, estimating normal vectors from noisy data for mesh denoising appears to be a chicken-and-egg problem and remains

an open topic [8,9]. In [10,11], range data and color images were fused for denoising, and promising results were reported. However, the prerequisites of these methods only hold when expensive sensors are applied or extra efforts are devoted to registering range data and images.

Sparse coding (SC) has revolutionized signal processing and led to state-of-the-art performance in a variety of applications, including image or video denoising, inpainting [12,13], restoration [14], and synthesis [15], face recognition [16], and anomaly detection [17]. These successes are mainly due to the fact that signal (images or image patches) have naturally sparse representations with respect to appropriate bases [18]. In recent years, such SC techniques have been applied to range data [19,20] and obtained competitive results. However, dictionary learning, which plays the central role in SC techniques, is computationally demanding, resulting in its limited applicability in real-time systems [21,22].

Based on the work of [20], in which reflectance information was synergized with depth data to facilitate the enforcement of adaptive sparsity constraint (therein, the depth and reflectance information that are complementary to each other work together to estimate the informative level of each patch, followed by adaptive dictionary learning which assigns more atoms to represent the patches with more information), we propose SC algorithms without dictionary learning to realize range data denoising via leveraging the regularity of laser range measurements in man-made environments. Specifically, a second-order differential transformation is applied to extract ridges that can be reconstructed and enhanced using a pre-learned ridge dictionary with sparsity constraint. Instead of performing refinement on the coefficients and dictionary alternately and iteratively [12,14,19,20,22], our method can achieve a one-step closed-form solution, resulting in much improved efficiency. Experiments on both synthesized data and real data were conducted to verify the effectiveness of our method, together with performance comparison against other state-of-the-art methods.

## 2. Our Method

### 2.1. Notation and Preliminaries

Matrices and vectors are shown in bold capital and bold lower-cased fonts, respectively. Sets are denoted with calligraphic fonts (e.g.,  $\mathcal{S}$ ), and their cardinality is denoted as  $|\mathcal{S}|$ . For a vector  $\mathbf{x} \in \mathbb{R}^m$ ,  $x[i]$  is the  $i$ -th entry,  $\mathbf{x}_{\mathcal{S}}$  is the subvector of  $\mathbf{x}$  corresponding to the entries with indices in  $\mathcal{S}$ . We define  $\ell_2$ -norm,  $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1,\dots,m} x[i]^2}$ ;  $\ell_1$ -norm,  $\|\mathbf{x}\|_1 = \sum_{i=1,\dots,m} |x[i]|$ ;  $\ell_0$ -norm,  $\|\mathbf{x}\|_0 = |\text{supp}(\mathbf{x})|$  (i.e., the number of nonzero elements).  $\text{sign}(\mathbf{x})$  is a sign vector with entries:  $\text{sign}(x[i]) = 1$  if  $x[i] > 0$ ,  $\text{sign}(x[i]) = -1$  if  $x[i] < 0$ , and zero otherwise. Based on the sign operator, we define the shrinkage operator:

$$\mathcal{T}_{\alpha}(\mathbf{x})[i] = (|x[i]| - \alpha)_+ \text{sign}(x[i]) = \max(|x[i]| - \alpha, 0) \text{sign}(x[i]).$$

For a matrix  $\mathbf{X} \in \mathbb{R}^{m \times n}$  and an index set  $\mathcal{S}$ ,  $\mathbf{X}_{\mathcal{S}}$  is the submatrix containing only the rows of  $\mathbf{X}$  with indices in  $\mathcal{S}$ .  $\text{vec}(\mathbf{X})$  denotes the column-wise vectorization of matrix  $\mathbf{X}$ . Letters  $d$  and  $r$  in superscript refer to the components related to the depth and reflectance information, respectively. Letters  $v$  and  $h$  in subscript refer to the components related to the vertical and horizontal directions, respectively.

**Ridge detection for range measurements.** We first take a scan from a 2D laser scanner as an example to introduce our operator of ridge detection and then extend the approach to 3D depth profiles. In Figure 1, we plot the depth profile  $\mathbf{z} \in \mathbb{R}^n$  of a typical indoor environment, which is obtained using a standard planar range finder measured with (known) discrete angles. Clearly, the profile is sufficiently regular and contains a few corners. Considering three consecutive points at coordinates  $(x_{i-1}, z_{i-1})$ ,  $(x_i, z_i)$ , and  $(x_{i+1}, z_{i+1})$ , there is a corner at  $i$  if  $\frac{z_{i+1} - z_i}{x_{i+1} - x_i} \neq \frac{z_i - z_{i-1}}{x_i - x_{i-1}}$ . In the following, we assume that  $x_i - x_{i-1} = 1$  for all  $i$  (this assumption comes without loss of generality since we can define it at arbitrary resolution). Therefore, the corner detection for the 2D profile is to find those indices  $i$

such that  $z_{i-1} - 2z_i + z_{i+1} \neq 0$ . To make the notation more compact, we introduce the second-order difference operator:

$$D_{2nd} = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix} \in \mathbb{R}^{(n-2) \times n}. \quad (1)$$

Thus, a simple operation  $D_{2nd} \cdot z$  can detect corners. For a 3D depth profile  $Z \in \mathbb{R}^{r \times c}$ , we estimate the ridges of vertical and horizontal directions  $R_v$  and  $R_h$ , respectively:

$$\begin{aligned} R_v &= D_{v2nd} \cdot Z \in \mathbb{R}^{(r-2) \times c}, \\ R_h &= Z \cdot D_{h2nd}^T \in \mathbb{R}^{r \times (c-2)}, \end{aligned} \quad (2)$$

where the matrices  $D_{v2nd}$  and  $D_{h2nd}$  are the same as  $D_{2nd}$  in Equation (1), but with suitable dimensions. To combine the two equations of (2), we reformulate it as below:

$$\begin{bmatrix} \text{vec}(R_h) \\ \text{vec}(R_v) \end{bmatrix} = \begin{bmatrix} I_c \otimes D_{v2nd} \\ D_{h2nd} \otimes I_r \end{bmatrix} \cdot \text{vec}(Z) = D_{vh} \cdot \text{vec}(Z). \quad (3)$$

Here,  $\otimes$  is the Kronecker product, and  $D_{vh} \in \mathbb{R}^{2(r \cdot c - r - c) \times (r \cdot c)}$ .

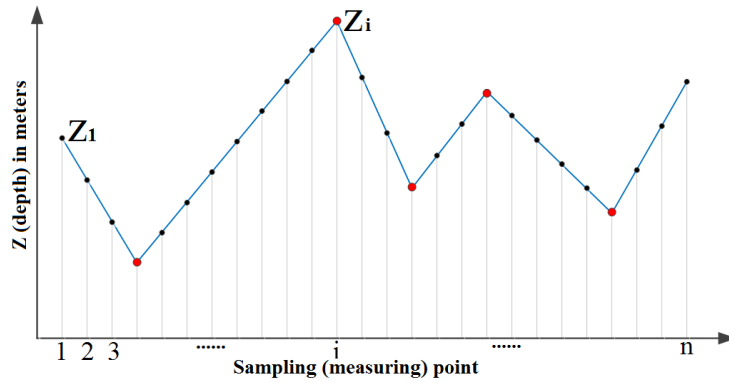


Figure 1. Illustration of the depth profile of a 2D laser scan.

## 2.2. Adaptive SC and Its $\ell_0, \ell_1$ Solutions

For each patch  $r_i$  (in the vectorized form) in the depth ridge map, we formulate the following adaptive SC model:

$$\begin{aligned} \min_{x_{vi}, x_{hi}} & \|r_{vi} - D_L \cdot x_{vi}\|_2 + \|r_{hi} - D_L \cdot x_{hi}\|_2, \\ \text{s.t. } & \forall 1 \leq i \leq N, \quad \|x_{vi}\|_0 \leq k_{vi}, \quad \|x_{hi}\|_0 \leq k_{hi}, \end{aligned} \quad (4)$$

where  $r_{vi}$  and  $r_{hi}$  represent the patches from vertical and horizontal ridge maps, respectively.  $D_L$  is the fixed pre-learned dictionary, as shown in Figure 2, which is obtained by applying the basic SC algorithm [19] on 40 pre-selected ridge maps of man-made scenarios (here, we display our pre-learned dictionary with the optimal setting of parameters as obtained in Section 3.1.1, namely the number of dictionary atoms is 1024 and the size of each atom is  $8 \times 8$ ). In [23], different dictionaries (including off-the-shelf ones such as discrete cosine transform (DCT) basis, wavelets basis, and dictionaries obtained via learning—either pre-learned or learned from the data itself) were tested for image denoising. Although the dictionary learned from the data itself reported the best denoising accuracy, it was also the most computationally demanding. In addition to our pre-learned dictionary, we also

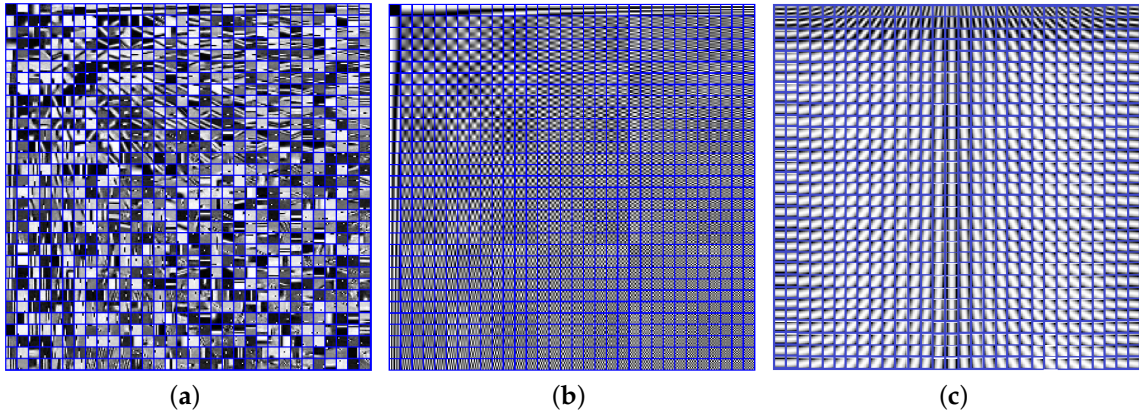
displayed the DCT dictionary, Gabor wavelets dictionary in Figure 2. The parameter tweaking of such dictionaries and their denoising performance are discussed in Section 3.1.1.  $x_{vi}$  and  $x_{hi}$  correspond to the sparse coefficient vectors.  $k_{vi}$  and  $k_{hi}$  are the given sparsity controlling parameters. Instead of setting these parameters as constants for all patches, we set them adaptively according to the informative level of each patch (see Section 3.1.1 for details) in a manner similar to that of [20]. Given  $k_{vi}$  and  $k_{hi}$ , the popular orthogonal-matching-pursuit (OMP) algorithm is applied to solve such  $\ell_0$  optimization problem. Moreover, the batch-OMP (BOMP) technique is exploited to further improve the efficiency. As the problem of (4) is non-convex and NP-hard, we reformulate its relaxation as below:

$$\begin{aligned} \min_{x_{vi}, x_{hi}} & ||r_{vi} - D_L \cdot x_{vi}||_2 + \lambda_{vi} ||x_{vi}||_1 \\ & + ||r_{hi} - D_L \cdot x_{hi}||_2 + \lambda_{hi} ||x_{hi}||_1. \end{aligned} \quad (5)$$

Here,  $\lambda_{vi}$  and  $\lambda_{hi}$ , which balance the representation fidelity term and the sparsity penalty term, are again estimated adaptively (see Section 3.1.1 for details). By stacking all the patches, we obtain Equation (6):

$$\begin{aligned} \min_{X_v, X_h} & ||[r_{v1}, \dots, r_{vN}] - D_L \cdot X_v||_2 + \sum_{i=1}^N \lambda_{vi} ||x_{vi}||_1 \\ & + ||[r_{h1}, \dots, r_{hN}] - D_L \cdot X_h||_2 + \sum_{i=1}^N \lambda_{hi} ||x_{hi}||_1, \end{aligned} \quad (6)$$

in which the neighboring patches are extracted with half-overlapping. To solve such  $\ell_1$  optimization problem, the least-absolute-shrinkage-and-selection-operator (LASSO) algorithm was proposed, which has proven to find the global optimizer.



**Figure 2.** Our pre-learned dictionary of ridge maps (a), discrete cosine transform (DCT) dictionary (or say basis) (b), and Gabor wavelet dictionary (or say basis) (c).

Clearly, Equations (4)–(6) differ from previous methods [12,14,19,20,22] in one aspect, which is that  $D_L$  is fixed. Thus,  $X_v$  and  $X_h$  can be obtained with a one-step closed-form solution instead of performing refinement on the coefficients and dictionary alternately and iteratively, thus resulting in much improved efficiency. With  $X_v$  and  $X_h$  in hand, we reconstruct each patch  $\hat{r}_{vi}$  and  $\hat{r}_{hi}$ . When all patches are processed, the average value is used for the locations occupied by multiple patches, and then we obtain the refined ridge maps  $\hat{R}_v$  and  $\hat{R}_h$ . Next, we can recover  $\hat{Z}$  by performing the inverse operation of Equation (3). However, due to the dependency of the rows of  $D_{vh}$ , the system is under-determined. Therefore, we need to incorporate the boundary conditions to recover  $\hat{Z}$  from its refined second-order difference, as shown in Equation (7):

$$\begin{bmatrix} z_N \\ \text{vec}(\mathbf{R}_h) \\ \text{vec}(\mathbf{R}_v) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_N & \mathbf{0}_{N \times (r \cdot c - N)} \\ & \mathbf{D}_{vh} \end{bmatrix} \cdot \text{vec}(\mathbf{Z}) = \begin{bmatrix} \mathbf{A} \\ \mathbf{D}_{vh} \end{bmatrix} \cdot \text{vec}(\mathbf{Z}), \quad (7)$$

where the available  $N$  boundary points are incorporated, and six boundary points are used in our work. A more rigorous theoretical proof of the minimum necessary boundary points is out of the scope of this work. Performing the inverse operation on (7), we can recover  $\hat{\mathbf{Z}}$  as Equation (8):

$$\text{vec}(\hat{\mathbf{Z}}) = \left[ \begin{bmatrix} \mathbf{A}^T & \mathbf{D}_{vh}^T \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{D}_{vh} \end{bmatrix} \right]^{-1} \begin{bmatrix} \mathbf{A}^T & \mathbf{D}_{vh}^T \end{bmatrix} \begin{bmatrix} z_N \\ \text{vec}(\mathbf{R}_h) \\ \text{vec}(\mathbf{R}_v) \end{bmatrix}. \quad (8)$$

Here, all the matrices  $\mathbf{A}$ ,  $\mathbf{D}_{vh}$ , their transposes, and their inverses can be pre-computed to significantly reduce the overall processing time. We now summarize all previous operations as Algorithm 1. In fact, our algorithm includes two versions of implementation: *our-BOMP* and *our-LASSO*.

In Algorithm 2, the convergence criterion is well-defined as  $\|x\|_0 \leq k$ . In Algorithm 3, the convergence criterion is that the relative update difference of  $x_k$  is less than  $10^{-3}$ .

---

**Algorithm 1** Our Sparse Coding for Depth Data Denoising

---

**Input:** Depth map  $\mathbf{Z}$ , dictionary  $\mathbf{D}_c$ , parameters  $k_1, k_2, \lambda_v, \lambda_h$ ;

**Output:** Restored  $\hat{\mathbf{Z}}$  of Equation (8);

- 1: Estimate the ridge maps  $\mathbf{R}_V$  and  $\mathbf{R}_H$  as Equation (2);
  - 2: Extract and vectorize ridge map patches  $r_{vi}, r_{hi}, 1 \leq i \leq N$ ;
  - 3: //Line 5 & 7 apply different algorithms to estimate  $x_{vi}, x_{hi}$ .
  - 4: //Line 5 belongs to the version of *our-BOMP*.
  - 5: Apply the Batch-OMP algorithm to solve Equation (4);
  - 6: //Line 7 belongs to the version of *our-LASSO*.
  - 7: Apply the LASSO algorithm to solve Equation (5);
  - 8: Obtain the refined ridge maps  $\hat{\mathbf{R}}_v$  and  $\hat{\mathbf{R}}_h$  via reconstruction;
  - 9: Recover  $\hat{\mathbf{Z}}$  via applying Equation (8).
- 

---

**Algorithm 2** Batch-orthogonal-matching-pursuit [24] to solve:  $\min_x \|\mathbf{y} - \mathbf{D} \cdot \mathbf{x}\|_2$  s.t.  $\|\mathbf{x}\|_0 \leq k$

---

**Input:**  $\mathbf{y}, \mathbf{D}, k$ ;

**Output:**  $\mathbf{x}$ ;

- 1: Initialization:  $\mathbf{x} = \mathbf{0}, \mathbf{r} = \mathbf{y}, \mathcal{S} = \{\emptyset\}$ ;
- 2: **while** stopping criterion not met **do**
- 3:  $k = \text{argmax}_k |d_k^T \cdot \mathbf{r}|$ ;
- 4:  $\mathcal{L} = \mathcal{S}$ ;
- 5:  $\mathcal{S} = \{k\} + \mathcal{L}$ ;
- 6:  $\mathbf{G}_{\mathcal{S}} = \mathbf{D}_{\mathcal{S}}^T \mathbf{D}_{\mathcal{S}}, \mathbf{G}_{\mathcal{L}} = \mathbf{D}_{\mathcal{L}}^T \mathbf{D}_{\mathcal{L}}$ ;
- 7: //Line 8 obtains  $\mathbf{G}_{\mathcal{S}}^{-1}$  using progressive Cholesky update.
- 8:  $\mathbf{x}_{\mathcal{S}} = \mathbf{G}_{\mathcal{S}}^{-1} \mathbf{D}_{\mathcal{S}}^T \mathbf{y}$ ;

the key updating function to estimate  $\mathbf{G}_{\mathcal{S}}^{-1}$  is:

$$\mathbf{G}_{\mathcal{S}}^{-1} = \begin{bmatrix} \mathbf{G}_{\mathcal{L}}^{-1} + \frac{1}{a} \mathbf{G}_{\mathcal{L}}^{-1} \mathbf{D}_{\mathcal{L}}^T d_k d_k^T \mathbf{D}_{\mathcal{L}} \mathbf{G}_{\mathcal{L}}^{-1} & -\frac{1}{a} \mathbf{G}_{\mathcal{L}}^{-1} \mathbf{D}_{\mathcal{L}}^T d_k \\ -\frac{1}{a} d_k^T \mathbf{D}_{\mathcal{L}} \mathbf{G}_{\mathcal{L}}^{-1} & \frac{1}{a} \end{bmatrix}$$

here,  $a = d_k^T d_k - d_k^T \mathbf{D}_{\mathcal{L}} \mathbf{G}_{\mathcal{L}}^{-1} \mathbf{D}_{\mathcal{L}}^T d_k$

- 9:  $\mathbf{r} = \mathbf{y} - \mathbf{D}_{\mathcal{S}} \mathbf{x}_{\mathcal{S}}$ ;

- 10: **end while**
-

---

**Algorithm 3** Fast iterative shrinkage-thresholding algorithm (FISTA) [25] to solve LASSO problem:  
 $\min_x ||y - D \cdot x||_2 + \lambda ||x||_1$

---

**Input:**  $y, D, \lambda$ ;

**Output:**  $x$ ;

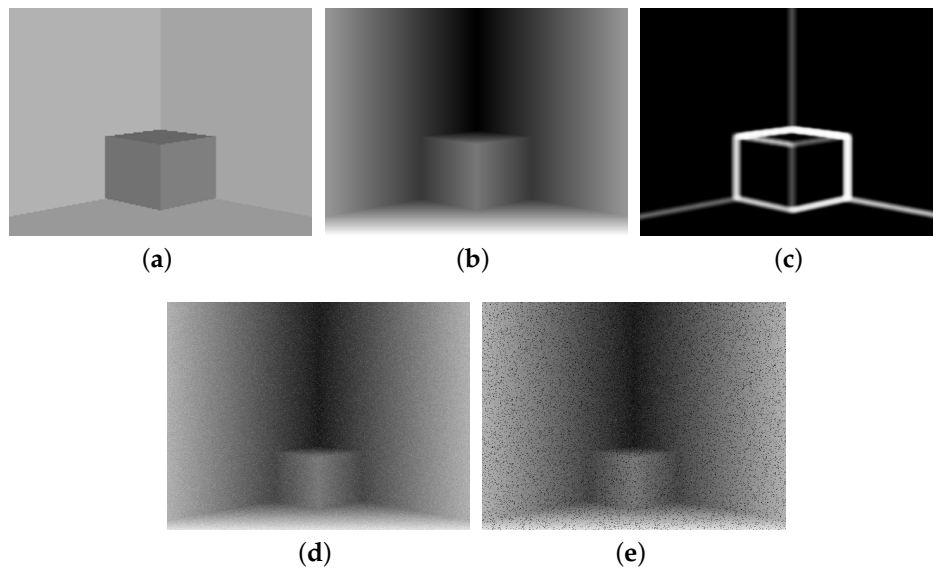
- 1: Initialization:  $x_0 = \mathbf{0}, z_1 = x_0, L = \max(\text{eig}(D^T D)), t_0 = 1, k = 0$ ;
  - 2: **while** not converged **do**
  - 3:  $k = k + 1$ ;
  - 4:  $\text{grad}(z_k) = D^T D z_k - D^T y$ ;
  - 5:  $x_k = \mathcal{T}_{\frac{\lambda}{L}}(z_k - \frac{1}{L} \text{grad}(z_k))$ ;
  - 6: //see Section 2.1 for the definition of  $\mathcal{T}_\alpha(x)$ .
  - 7:  $t_k = (1 + \sqrt{1 + 4 \cdot t_{k-1}^2})/2$ ;
  - 8:  $z_{k+1} = x_k + \frac{t_{k-1}-1}{t_k}(x_k - x_{k-1})$ ;
  - 9: **end while**
- 

### 3. Experiments and Analysis

We performed experiments on both synthesized data and actual data recorded using a laser scanner. In the synthesized data experiments where the ground truth was available, the setting of important parameters was investigated, followed by denoising evaluation. In the real data experiments, the results are also demonstrated for visual assessment, in addition to the performance comparison. Similar to [14,20], the peak signal-to-noise ratio (PSNR) value is employed to evaluate the performance. Here, we clarify that our work is run on an ASUS Pro Notebook with a 2.4 GHz Intel Quad-Core i7-4500U processor and 12 GB RAM executing MATLAB code in parallel.

#### 3.1. Experiments on Synthesized Data

Similar to [20,26], we designed a scene of a square box having sides of one meter in a room, as shown in Figure 3, where all information is known as the ground truth.



**Figure 3.** Synthesized data (of size  $800 \times 800$  pixels) designed for experiments. (a) Reflectance image. (b) Depth map. (c) Result of informative-level estimation. (d) Gaussian noise-corrupted depth data. (e) Gaussian noise and outliers-corrupted depth data.

### 3.1.1. Important Parameters

To achieve plausible performance in terms of both accuracy and efficiency, the following parameters play the central role: the sparsity controlling parameters  $k_{vi}$ ,  $k_{hi}$ ,  $\lambda_{vi}$ ,  $\lambda_{hi}$ , dictionary dimension  $d$  (which indicates that the patch size is  $\sqrt{d} \times \sqrt{d}$  pixels), and its atom number  $n_D$  ( $r = \frac{n_D}{d}$  is the redundancy of the dictionary).

Applying the same parameter tweaking strategy as that in [20], we recommend the sparsity controlling parameters as shown in Table 1. For each patch  $P_j$  ( $j = v$  or  $h$ , from vertical or horizontal ridge maps, respectively), the parameters  $\alpha_{vi}, \beta_{vi}$  (or  $\alpha_{hi}, \beta_{hi}$ ) are related to its informative-level measurement ( $InM$ ), which is estimated as below:

$$\begin{aligned} D_{en}(P_v^d) &= \sum_{i=1}^{\sqrt{d}} \sum_{j=1}^{\sqrt{d}} |P_v^d[i, j]|, \\ R_{en}(P^r) &= \sum_{i=1}^{\sqrt{d}} \sum_{j=1}^{\sqrt{d}} GradMag(P^r)[i, j], \quad GradMag(P^r)[i, j] = \sqrt{\left(\frac{\partial P^r[i, j]}{\partial x}\right)^2 + \left(\frac{\partial P^r[i, j]}{\partial y}\right)^2}, \\ InM(P_v^d) &= D_{en}(P_v^d) \cdot \exp(\sigma(R_{en}(P^r)) - 0.5). \end{aligned} \quad (9)$$

Here,  $GradMag(P^r)$  is the gradient magnitude map of patch  $P^r$ , and  $\sigma(\cdot)$  is the sigmoid function  $\sigma(\cdot) = \frac{1}{1+\exp(-\cdot)}$ . For the input from  $(-\infty, +\infty)$ , the output of  $\sigma(\cdot)$  is in the range  $(0, 1)$ . In our work, as the value of  $R_{en}$  cannot be negative, the output of  $\sigma(R_{en})$  is in the range  $[0.5, 1)$ . The output of  $\exp(\sigma(R_{en}(P^r)) - 0.5)$  is in the range  $[1, e^{0.5})$ . With  $InM(P_v^d)$ ,  $InM(P_h^d)$  for all patches in hand, we determine the sparsity controlling parameters via computing  $\alpha_{vi}, \beta_{vi}, \alpha_{hi}$ , and  $\beta_{hi}$  as below:

$$\begin{aligned} InM_{min} &= \min\{InM(P_{vi}^d), InM(P_{hi}^d)\}, \quad i = 1, \dots, N, \\ InM_{max} &= \max\{InM(P_{vi}^d), InM(P_{hi}^d)\}, \quad i = 1, \dots, N, \\ \alpha_{ji} &= \frac{InM_{max} - InM(P_{ji}^d)}{InM_{max} - InM_{min}}, \quad j = v \text{ or } h, \\ \beta_{ji} &= \frac{InM(P_{ji}^d) - InM_{min}}{InM_{max} - InM_{min}}, \quad j = v \text{ or } h. \end{aligned} \quad (10)$$

In Table 1, the positions of  $\alpha_{ji}$  and  $\beta_{ji}$  are exchanged in *Our-BOMP* and *Our-LASSO* because  $k_{ji}$  and  $\lambda_{ji}$  enforce the sparsity constraint in different manners to adaptively determine the number of dictionary atoms permitted for the patches with different informative levels. For example, to assign more atoms to represent the patch with more information,  $k_{ji}$  should increase to push up the sparsity limit, whereas  $\lambda_{ji}$  should decrease to penalize the sparsity constraint term to a lower degree.

In addition to  $k_{ji}, \lambda_{ji}$ , we tested different combinations of patch size and redundancy to find the most reasonable settings of  $d$  and  $r$ . Specifically,  $\sqrt{d}$  can be 4, 8, 16, or 24, and  $r$  can be 4, 8, 16, or 24. Thus, a total of 16 combinations were tested. As shown in Figure 4a, dictionaries with redundancy 16 consistently achieved better denoising results than dictionaries with redundancy 4, 8, and 24, when their patch sizes were the same. Figure 4b shows the computation time. Clearly, the time increased dramatically with increasing patch size and redundancy. Taking both the denoising performance and computation time into account, we conclude that the dictionary setting of  $\sqrt{d} = 8$  and  $r = 16$  (namely 1024 atoms) can yield decent outputs, while allowing fast computation. Such parameters are used throughout this paper, except where indicated. In Table 2, we further report the denoising result using different fixed dictionaries: our pre-learned dictionary, DCT dictionary, and Gabor wavelets dictionary. Clearly, the results using our pre-learned dictionary were better those using off-the-shelf ones.

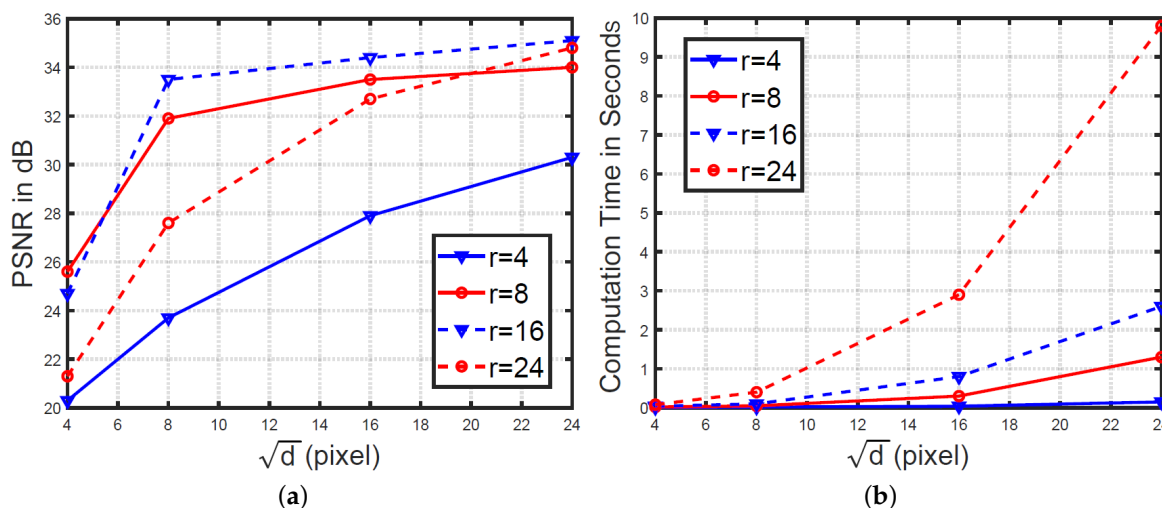
**Table 1.** Setting of sparsity controlling parameters. BOMP: batch orthogonal-matching-pursuit; LASSO: least-absolute-shrinkage-and-selection-operator.

| Parameters       | Our-BOMP   | Our-LASSO  |
|------------------|--|--|
|                  | $k_{vi}, k_{hi}$   | $\lambda_{vi}, \lambda_{hi}$   |
| Max value        | $k_{max} = 6$  | $\lambda_{max} = 2.2$  |
| Min value        | $k_{min} = 1$  | $\lambda_{min} = 0.2$  |
| Adaptive value * | $k_i^v = \alpha_i^v k_{min} + \beta_i^v k_{max}$<br>$k_i^h = \alpha_i^h k_{min} + \beta_i^h k_{max}$ | $\lambda_i^v = \beta_i^v \lambda_{min} + \alpha_i^v \lambda_{max}$<br>$\lambda_i^h = \beta_i^h \lambda_{min} + \alpha_i^h \lambda_{max}$ |

\*  $\alpha_i^v, \beta_i^v, \alpha_i^h, \beta_i^h$  are related to the informative estimation of each patch.

**Table 2.** Denoising results (peak signal-to-noise ratio (PSNR) value) using different dictionaries (unit: dB).

| Dictionary                 | Dictionary with Different Dimensions |                        |                         |
|----------------------------|--------------------------------------|------------------------|-------------------------|
|                            | $r = 10, \sqrt{d} = 8$               | $r = 16, \sqrt{d} = 8$ | $r = 16, \sqrt{d} = 16$ |
| DCT dictionary             | 32.26                                | 32.37                  | 32.17                   |
| Gabor Wavelet dictionary   | 32.46                                | 32.58                  | 32.92                   |
| Our pre-learned dictionary | 32.86                                | <b>33.75</b>           | <b>33.81</b>            |

**Figure 4.** (a) PSNR and (b) computation time with dictionaries of different combinations of patch size and redundancy.

### 3.1.2. Denoising Results

The denoising results of our work on the synthesized data (see Figure 3d in which 100% dense Gaussian noises with standard deviation of 0.02 m and gray-scale value 5 for range and reflectance data respectively were added) are reported in Table 3, together with the comparison against available competitive methods including *Trilateral filter* [26], *Basic SC* [19], and *Adaptive SC* [20]. Moreover, in addition to the Gaussian noise, sparse outliers (5% of the total positions, which were randomly distributed) were added to the scene, as shown in Figure 3e, and the denoising results are also reported in Table 3. In addition to the PSNR criteria, the root mean square (RMS, with units of millimeters) error indicating the difference between the recovered range data and the ground truth is also included, since it can give readers a more direct indication of how well the contaminated range information was restored by these methods. Moreover, the computational time of each method is also reported. As shown in Table 3, our methods outperformed *Trilateral filter*, *Basic SC*, and obtained comparable

results to *Adaptive SC* in terms of denoising accuracy (the best results were obtained by *Adaptive SC* with PSNR = 33.83 and RMS = 1.98), while achieving significantly higher efficiency than *Adaptive SC*. Moreover, *Our-BOMP* even worked slightly more efficiently than *Our-LASSO*, achieving the processing speed of approximately 13 fps for the range image with resolution  $800 \times 800$ .

**Table 3.** Denoising results on the synthesized data. RMS: root mean square; SC: sparse coding.

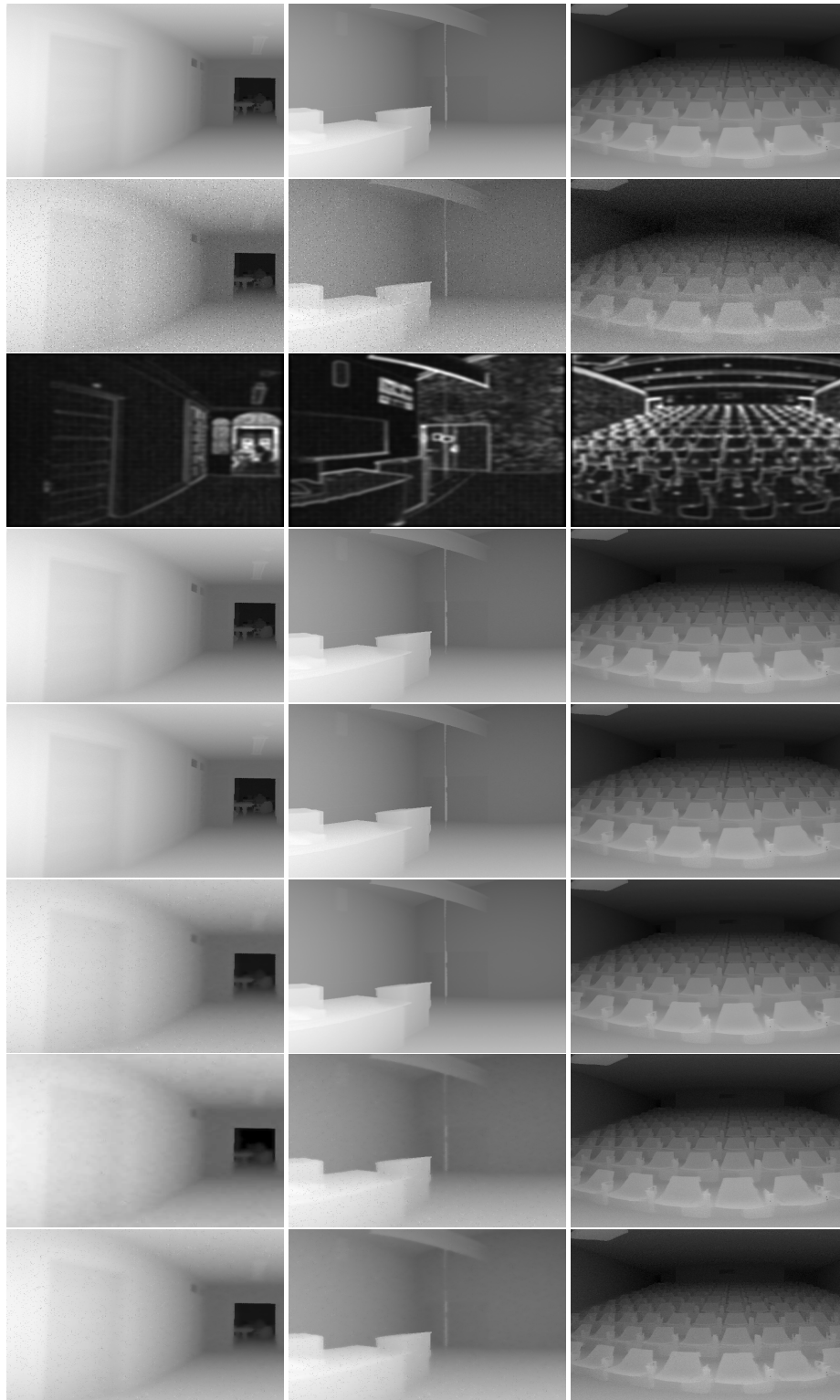
| Method            | Only Gaussian Noise |          |            | Gaussian Noise and Outliers |          |            |
|-------------------|---------------------|----------|------------|-----------------------------|----------|------------|
|                   | PSNR (dB)           | RMS (mm) | Time * (s) | PSNR (dB)                   | RMS (mm) | Time * (s) |
| Our-BOMP          | 33.72               | 2.06     | 0.08       | 33.28                       | 2.14     | 0.09       |
| Our-LASSO         | 33.78               | 2.04     | 0.12       | 33.35                       | 2.12     | 0.15       |
| Adaptive SC       | 33.83               | 1.98     | 227.38     | 33.71                       | 2.04     | 239.62     |
| Basic SC          | 33.17               | 2.17     | 153.62     | 33.02                       | 2.56     | 172.38     |
| Trilateral filter | 32.37               | 2.49     | 0.04       | 32.12                       | 2.67     | 0.06       |

\* Time indicates the computation time for each method.

### 3.2. Denoising Data from Laser Scanner

This set of experiments used the Brown range image database [27] captured with a Riegl LMS-Z210 laser scanner, its field of view (FoV) is  $80^\circ$  vertically and  $259^\circ$  horizontally. We focused only on the 15 indoor scenes (e.g., classrooms and theatres). To deal with (and benefit from) such a large horizontal FoV, similar to the multiple local projection method in [19], we partitioned the scene into several parts with  $80^\circ$  FoV vertically and horizontally. Note that in the Brown range image database, the depth of each measuring point is saved sequentially, left to right and up to down. Thus, the image location only indicates the order of saving and it was necessary to perform local projection to obtain reasonable images. The consecutive parts were half-overlapping along the horizontal direction, then each part was projected to an image plane orientated to the center. Finally, the average of all the restored parts was computed to restore the whole scene. In the following, the denoising results on such real data are reported and discussed. Similar to Section 3.1, Gaussian noise was added to the depth and reflectance data, and the results on three representative scenes are shown in Figure 5.

As can be observed in Figure 5, the results of *Trilateral filter* were good in the planar regions, but contained a great deal of noise in the edge regions. Our method, *Basic SC*, and *Adaptive SC* are all based on SC, and obtained consistent results in all regions. Moreover, the results of our method in the third column were slightly worse than those of columns 1 and 2, because the curved structure of the scene in the third column posed a greater challenge to our sparse ridge assumption. In Table 4, the average denoising results on all 15 indoor scenes of the dataset are reported. Similar to the results on the synthesized data, our method outperformed *Trilateral filter* and *Basic SC* consistently, and obtained comparable results to those of *Adaptive SC* in terms of denoising accuracy, while achieving significantly higher efficiency than *Adaptive SC*. Compared with the results in Table 3, the results of Table 4 deteriorated as the structure of real scenes was more complex than the synthetic scene. Similar to the previous section, in addition to the Gaussian noise, sparse outliers were added to the scene, and the denoising results are also reported in Table 4.



**Figure 5.** Results of denoising experiments on real data (see online version for details). The original and noisy corrupted range images are shown in the first and second rows, respectively. The third row depicts our intermediate results of informative level estimation. Rows 4 to 8 depict the results of *Our-BOMP*, *Our-LASSO*, *Adaptive SC*, *Basic SC*, and *Trilateral filter*, respectively.

**Table 4.** Denoising results on real data.

| Method            | Only Gaussian Noise |          |            | Gaussian Noise and Outliers |          |            |
|-------------------|---------------------|----------|------------|-----------------------------|----------|------------|
|                   | PSNR (dB)           | RMS (mm) | Time * (s) | PSNR (dB)                   | RMS (mm) | Time * (s) |
| Our-BOMP          | 35.37               | 2.31     | 0.86       | 35.17                       | 2.43     | 0.89       |
| Our-LASSO         | 35.43               | 2.27     | 1.15       | 35.21                       | 2.37     | 1.28       |
| Adaptive SC       | 35.72               | 2.03     | 918.71     | 35.68                       | 2.08     | 972.33     |
| Basic SC          | 34.28               | 2.83     | 538.37     | 34.16                       | 2.97     | 572.81     |
| Trilateral filter | 32.13               | 3.27     | 0.16       | 32.02                       | 3.75     | 0.18       |

\* Time indicates the computation time for each method.

#### 4. Conclusions

In this study, based on our previous work [20], we propose SC-based algorithms in which a pre-learned ridge dictionary is applied to realize range data denoising by leveraging the regularity of laser range measurements in man-made environments. Experiments on both synthesized data and real data demonstrate that our method obtained accuracies comparable to those of sophisticated SC methods with much higher efficiency, achieving approximately 13 fps (for resolution of  $800 \times 800$ ) with the resource of a lightweight laptop computer. Therefore, our method can be applied for real-time systems in man-made environments. Furthermore, we are trying to implement our method with GPU acceleration, intending to further increase the efficiency and also save the precious payload for unmanned systems, especially UAVs.

**Author Contributions:** Zhi Gao designed the algorithm and wrote the paper; Mingjie Lao designed the codes and performed the experiments; Yongsheng Sang also helped in coding and experimental tuning; Fei Wen focused on the search of related work and experimental comparison; Bharath Ramesh provided significant comments and suggestions; Ruifang Zhai helped with revision.

**Funding:** This research was supported in part by the Science Foundation of Ministry of Education (MOE) of China and China Mobile Communications Corporation under Grant MCM20160307.

**Conflicts of Interest:** The authors declare no conflict of interest.

#### References

1. Buades, A.; Coll, B.; Morel, J.M. A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.* **2005**, *4*, 490–530. [\[CrossRef\]](#)
2. Lysaker, M.; Lundervold, A.; Tai, X.C. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Trans. Image Proc.* **2003**, *12*, 1579–1590. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Wu, Y.; Tracey, B.; Natarajan, P.; Noonan, J.P. Probabilistic non-local means. *IEEE Signal Proc. Lett.* **2013**, *20*, 763–766. [\[CrossRef\]](#)
4. Smigiel, E.; Alby, E.; Grussenmeyer, P. TLS data denoising by range image processing. *Photogramm. Rec.* **2011**, *26*, 171–189. [\[CrossRef\]](#)
5. Sun, B.; Fang, H.; Huang, D. Lidar signal denoising using least-squares support vector machine. *IEEE Signal Proc. Lett.* **2005**, *12*, 101–104.
6. Fleishman, S.; Drori, I.; Cohen-Or, D. Bilateral mesh denoising. *ACM Trans. Graph.* **2003**, *22*, 950–953. [\[CrossRef\]](#)
7. Marco, M.; Luca, P.; Augusto, S.; Stefano, T. Fast PDE approach to surface reconstruction from large cloud of points. *Comput. Vis. Image Underst.* **2008**, *112*, 274–285. [\[CrossRef\]](#)
8. Belyaev, Y.; Seidel, H. Mesh smoothing by adaptive and anisotropic Gaussian filter applied to mesh normals. In *Vision, Modeling, and Visualization*; IOS Press: Amsterdam, The Netherlands, 2002; pp. 203–210.
9. Tasdizen, T.; Whitaker, R.; Burchard, P.; Osher, S. Geometric surface smoothing via anisotropic diffusion of normals. In *Proceedings of the Conference on Visualization'02*, Boston, MA, USA, 27 October–1 November 2002; pp. 125–132.

10. Crabb, R.; Tracey, C.; Puranik, A.; Davis, J. Real-time foreground segmentation via range and color imaging. In Proceedings of the Computer Vision and Pattern Recognition Workshops, Anchorage, AK, USA, 23–28 June 2008; pp. 1–5.
11. Huhle, B.; Schairer, T.; Jenke, P.; Straber, W. Fusion of range and color images for denoising and resolution enhancement with a non-local filter. *Comput. Vis. Image Underst.* **2010**, *114*, 1336–1345. [[CrossRef](#)]
12. Elad, M.; Aharon, M. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Proc.* **2006**, *15*, 3736–3745. [[CrossRef](#)]
13. Ji, H.; Huang, S.; Shen, Z.; Xu, Y. Robust video restoration by joint sparse and low rank matrix approximation. *SIAM J. Imaging Sci.* **2011**, *4*, 1122–1142. [[CrossRef](#)]
14. Mairal, J.; Elad, M.; Sapiro, G. Sparse representation for color image restoration. *IEEE Trans. Image Proc.* **2008**, *17*, 53–695. [[CrossRef](#)]
15. Peyré, G. Sparse modeling of textures. *J. Math. Imaging Vis.* **2009**, *34*, 17–31. [[CrossRef](#)]
16. Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Ma, Y. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 210–227. [[CrossRef](#)] [[PubMed](#)]
17. Adler, A.; Elad, M.; Hel-Or, Y.; Rivlin, E. Sparse coding with anomaly detection. *J. Signal Proc. Syst.* **2015**, *79*, 179–188. [[CrossRef](#)]
18. Wright, J.; Ma, Y.; Mairal, J.; Sapiro, G.; Huang, T.S.; Yan, S. Sparse representation for computer vision and pattern recognition. *Proc. IEEE* **2010**, *98*, 1031–1044. [[CrossRef](#)]
19. Mahmoudi, M.; Sapiro, G. Sparse representations for range data restoration. *IEEE Trans. Image Proc.* **2012**, *21*, 2909–2915. [[CrossRef](#)] [[PubMed](#)]
20. Gao, Z.; Li, Q.; Zhai, R.; Lin, F. Laser Range Data Denoising via Adaptive and Robust Dictionary Learning. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1750–1754.
21. Elad, M.; Figueiredo, M.A.; Ma, Y. On the role of sparse and redundant representations in image processing. *Proc. IEEE* **2010**, *98*, 972–982. [[CrossRef](#)]
22. Gao, Z.; Li, Q.; Zhai, R.; Shan, M.; Lin, F. Adaptive and Robust Sparse Coding for Laser Range Data Denoising and Inpainting. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 2165–2175. [[CrossRef](#)]
23. Elad, M. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*; Springer: New York, NY, USA, 2010.
24. Rubinstein, R.; Zibulevsky, M.; Elad, M. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *CS Technion* **2008**, *40*, 1–15.
25. Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [[CrossRef](#)]
26. Oishi, S.; Kurazume, R.; Iwashita, Y.; Hasegawa, T. Denoising of range images using a trilateral filter and belief propagation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September 2011; pp. 2020–2027.
27. Lee, A.B.; Huang, J. Brown Range Image Database. 2000. Available online: [www.dam.brown.edu/ptg/brid/index.html](http://www.dam.brown.edu/ptg/brid/index.html) (accessed on 26 April 2017).

