Joint Inpainting of Depth and Reflectance with Visibility Estimation

Marco Bevilacqua, Jean-François Aujol, Mathieu Brédif, Aurélie Bugeau

To cite this version:

Marco Bevilacqua, Jean-François Aujol, Mathieu Brédif, Aurélie Bugeau. Joint Inpainting of Depth and Reflectance with Visibility Estimation. 2016. hal-01348304

HAL Id: hal-01348304

https://hal.science/hal-01348304v1

Preprint submitted on 22 Jul 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Joint Inpainting of Depth and Reflectance with Visibility Estimation☆

Marco Bevilacqua[a,b,c,d], Jean-François Aujol[a], Mathieu Brédif[c], Aurélie Bugeau[b]

[a] *Université de Bordeaux, IMB, CNRS UMR 5251, 33400 Talence, France.*
[b] *Université de Bordeaux, LaBRI, CNRS UMR 5800, 33400 Talence, France.*
[c] *Université Paris-Est, IGN, SRIG, MATIS, 73 Avenue de Paris, 94160 Saint-Mandé, France.*
[d] *Bordeaux INP, IMS, CNRS UMR UMR 5218, 33400 Talence, France.*

Abstract

This paper presents a novel strategy to generate, from 3-D lidar measures, dense depth and reflectance images coherent with given color images. It also estimates for each pixel of the input images a visibility attribute. 3-D lidar measures carry multiple information, e.g. relative distances to the sensor (from which we can compute depths) and reflectances. When projecting a lidar point cloud onto a reference image plane, we generally obtain sparse images, due to undersampling. Moreover, lidar and image sensor positions typically differ during acquisition; therefore points belonging to objects that are hidden from the image view point might appear in the lidar images. The proposed algorithm estimates the complete depth and reflectance images, while concurrently excluding those hidden points. It consists in solving a joint (depth and reflectance) variational image inpainting problem, with an extra variable to concurrently estimate handling the selection of visible points. As regularizers, two coupled total variation terms are included to match, two by two, the depth, reflectance, and color image gradients. We compare our algorithm with other image-guided depth upsampling methods, and show that, when dealing with real data, it produces better inpainted images, by solving the visibility issue.

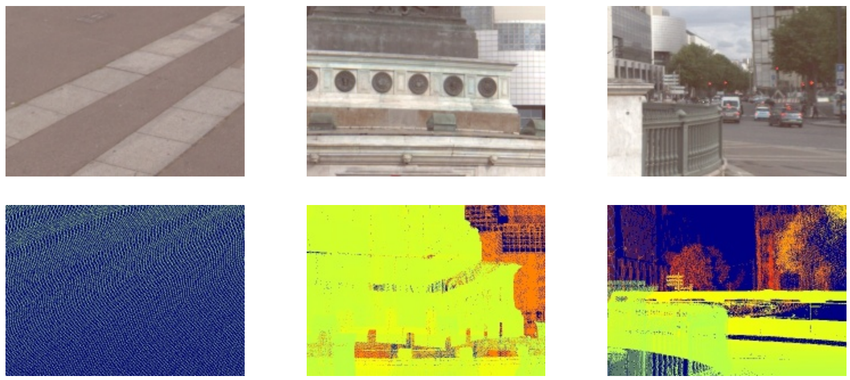
☆This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the Investments for the future Programme IdEx Bordeaux (ANR-10-IDEX-03-02). J.-F. Aujol also acknowledges the support of the Institut Universitaire de France.

Keywords: Inpainting, Total Variation, Depth Maps, Lidar, Reflectance, Point Cloud, Visibility

1. Introduction

Image-based 3D reconstruction of static and dynamic scenes (Herbort and Wöhler, 2011; Seitz et al., 2006; Stoykova et al., 2007) is one of the main challenges in computer vision nowadays. In the recent years many efforts have been made to elaborate configurations and approaches, possibly requiring the employment of multiple sensors, with the final goal of generating plausible and detailed 3D models of scenes. To this end, typical optical cameras are often combined with non-visual sensors. The intermediate outputs of these hybrid systems, prior to the final scene rendering, are in general depth or depth+color images (RGB-D). Among the non-visual sensors, we can find Time-of-Flight (ToF) cameras (Kolb et al., 2010), which acquire low-resolution co-registered depth and color images at a cheap cost, and the famous Kinect (Zhang, 2012), capable to extract depth information by exploiting structural light. Another possibility is represented by lidar devices, which are used in a variety of applications and provide as output point clouds with measures of distance and reflectivity of the sensed surfaces.

This work lies in the context described and is particularly driven by the exploitation of data acquired by Mobile Mapping Systems (MMS), such as (Paparoditis et al., 2012). MMS systems are vehicles equipped with high-resolution cameras and at least one lidar sensor: their contained dimensions allow them to be driven through regular streets and acquire data of urban scenes. The data acquired is a set of calibrated and geolocated images, together with coherent lidar point clouds. The interest towards them comes from the possibility of having available, at a relatively small processing cost, the combination of depth and color information, without having to perform explicit (error-prone) reconstructions. Having a good depth estimate at each pixel, for example, would enable the possibility to perform depth-image-based rendering algorithms, e.g. (Chen et al., 2005; Schmeing and Jiang, 2011; Zinger et al., 2010). Similarly, the availability of depth information allows the insertion of virtual elements into the image, such as pedestrians or vehicles generated by a traffic simulation (Brédif, 2013). While MMS data sets do not include directly depth images aligned with the available color images, it is easy, by exploiting the known geometry, to project the lidar point clouds onto each image. This operation produces initial depth images, which present three main issues (see Figure 1, where three parts of an input depth image are shown, together with the corresponding image parts).



(a) Undersampling (b) Visibility (c) Occlusions

Figure 1: Examples of parts from a resulting input depth image (bottom row), with the corresponding parts from the reference color image (top row), showing the three issues mentioned: undersampling, appearance of hidden points, and presence of occlusions.

1. *Undersampling*: since lidar and image acquisitions are deeply different in terms of geometry and characteristics, the resulting depth images turn to be irregular. No points are present in the sky and on reflective surfaces. Moreover, the point density, which depends on the variable distances between the camera image plane and the positions of the lidar sensor, is generally significantly smaller than the pixel resolution. We can therefore talk about sparse input depth images (see for example Figure 1a, showing the low density of lidar points from the ground).

2. *Visibility* (hidden parts appear): since points that are not visible from the image view point (hidden points) can be occasionally "seen" by the moving lidar sensor, erroneous values referring to such points can appear in the input depth image. This occurs even when a Z-buffer approach (Greene et al., 1993) is used, i.e. only the closest depth values for each pixel are kept (in case multiple values end up in the same pixel location). E.g., Figure 1b shows that depth values from the building behind appear as foreground points.

3. *Occlusions* (visible parts disappear): for the same reason as above, i.e. the different acquisition timing and geometry between image and lidar

sensors, surfaces normally visible from the image view point do not get a corresponding depth. This can happen when the lidar sensor suffers occlusions at a given instant or because of the scene dynamics. E.g., in Figure 1c, a moving bus that is not present at the moment of the image shot happens to appear in the depth image.

While there is variety of methods in the literature that deal with the first issue, i.e. that aim at upscaling an irregular input depth image possibly with the guidance of a corresponding color image, little work has been performed to address the last two issues. In this paper, while inpainting the input depth image, we also intend to tackle the visibility problem. Moreover, we treat at the same time an additional input: a sparse reflectance image derived in the same way as the input depth image (i.e., by naively projecting the lidar point cloud, considering the reflectance information carried out by each point). We will show that the simultaneous use of a reflectance image, which is inpainted jointly with the depth, improves the quality of the produced depth image itself. To jointly inpaint depth and reflectance and concurrently evaluate the visibility of each point (i.e. establish if a single point is reliable or, since non-visible, must be discarded), we formulate an optimization problem with three variables to estimate: depth, reflectance and a visibility attribute per pixel. The inpainting process is also guided by the available color image, by means of a two-fold coupled total variation (TV) regularizer.

The remainder of the paper is organized as follows. In Section 2, we present our approach and mention the related works, in particular on the image-guided depth inpainting problem. In sections 3 and 4 we describe the model used and the primal-dual optimization algorithm that arises, respectively. Finally, in Section 5 we bring experimental evidence that proves the effectiveness of the proposed approach.

2. Problem addressed and related work

Figure 2 depicts the scheme of the proposed approach. Given an MMS data set consisting of a lidar point cloud and a set of camera images, we choose among the latter a reference color image (w), and we obtain input depth (u_S) and reflectance (r_s) images by re-projecting the lidar points according to the image geometry. The two lidar-originated images are sparse images with irregular sampling and need to be inpainted. We propose to do that jointly and simultaneously estimate the visibility of the input points, within a variational optimization framework. The output of the algorithm are then three: the inpainted depth and reflectance (u and r, respectively), and a binary image expressing the visibility at each point (v).

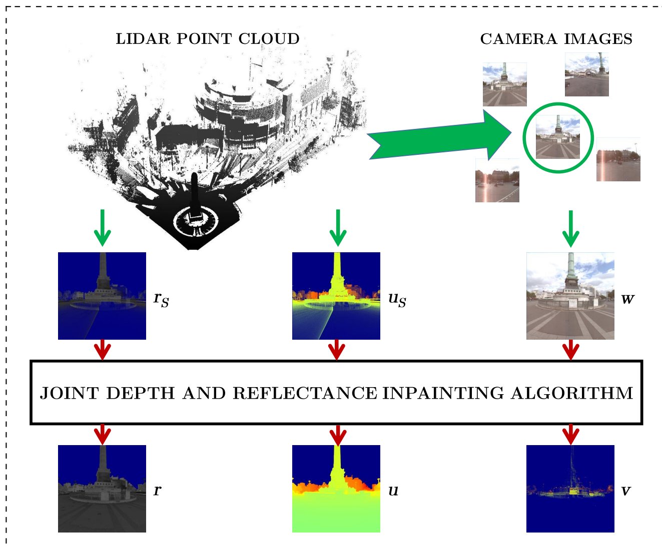


Figure 2: General scheme of the proposed approach. The final outputs of the algorithm are the inpainted reflectance and depth images, r and d respectively, and a binary visibility image v. To represent v, we show the original depth values that finally get $v \simeq 0$.

In the literature there is a variety of methods that aim at upscaling or inpainting an original sparse depth image. Most of them are presented in the context of ToF cameras; thus, a high quality color image is acquired at the same time and can be exploited. We refer to this problem as image-guided depth inpanting. The typical assumption, when exploiting the available image, is that image edges are related to depth edges. Following this principle, many approaches have been proposed, such as methods using different versions of multilateral filtering (Chan et al., 2008; Garcia et al., 2010; Yang et al., 2013), methods based on Markov Random Fields (Diebel and Thrun, 2005), and methods using Non-Local Means (Huhle et al., 2010; Park et al., 2011). Another family relates to recent methods that make use of optimization (Ferstl et al., 2013; Harrison and Newman, 2010; Liu and Gong, 2013; Schwarz et al., 2012). Among these, in (Harrison and Newman, 2010), a method to assign image pixel with a range value, using both image appearance and sparse laser data, is proposed. The problem is posed as an optimization of a cost function encapsulating a spatially varying smoothness cost and measurement compatibility. Another optimization-based depth upsampling method is presented in Ferstl et al. (2013): an Anisotropic Total Generalized Variation (ATGV) term is proposed to regularize the solution while exploiting the color image information.

While presenting good results on "non-problematic images", in none of the mentioned methods the visibility issue is tackled, i.e. there is no estimation of input depth measures to possibly remove, but all input depth measures are assumed to be valid and equally contribute to the inpainting process. We instead intend to estimate visibility, to be able to cope with realistic depth images. To this end, we build on our previous work on lidar-based depth inpainting (Bevilacqua et al., 2016). W.r.t. the latter, the model is significantly modified to include a reflectance image as well into a new optimization framework. We will show that depth and reflectance mutually benefit of each other in the inpainting process, thus leading to better output results for both. In the next section we present the novel model.

3. Model

Let $\Omega \subseteq \mathbb{R}^2$ be the "full" image support, and $\Omega_S \subseteq \Omega$ the sparse image support where the input images are defined (i.e., there is at least one lidar point ending up there after projection). Given an input depth image $u_S : \Omega_S \to \mathbb{R}$, an input reflectance image $r_S : \Omega_S \to \mathbb{R}$, and the luminance component of their corresponding color image $w : \Omega \to \mathbb{R}$ (defined in the complete domain), the goal is to fully inpaint the depth and reflectance input images to obtain $u : \Omega \to \mathbb{R}$ and $r : \Omega \to \mathbb{R}$, and concurrently estimate a visibility attribute $v : \Omega_S \to \mathbb{R}$. For each input point, v indicates whether it is visible from the image view point and should thus be taken into account in the inpainting process. Figure 3 reports an example of three possible input images - depth (u_S), reflectance (r_S) and camera images - and their respective gradient images.

We model our joint inpainting problem as an optimization problem with three variables, u, r, and v, to be estimated. Lower and upper bounds for the values of u and r are considered in the expression. The visibility attribute v takes values in $[0, 1]$, where $v = 0$ stands for "hidden" and $v = 1$ means that the point is visible from the considered image view point. The model

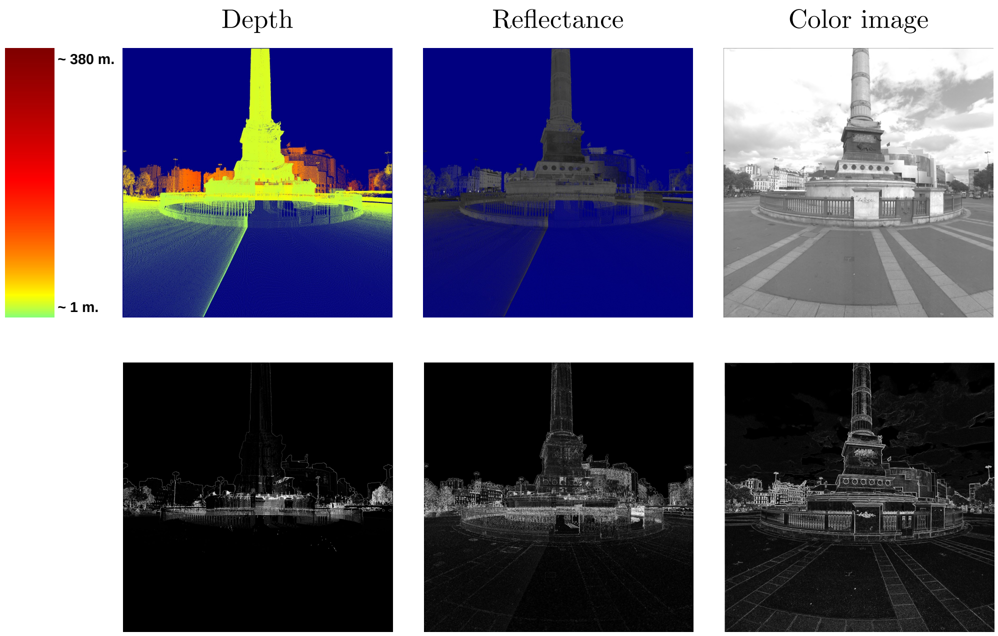


Figure 3: Example of input depth, reflectance and color images (top row), and their respective gradient images (bottom row). Besides the input depth image, the color map used to encode depth values is reported. Gradients of depth and reflectance are computed on the interpolated versions of the input sparse images, initially obtained by nearest neighbor interpolation.

considered consists of four terms:

$$\min_{\substack{u\in[u_{\mathrm{m}},u_{\mathrm{M}}]\\ r\in[r_{\mathrm{m}},r_{\mathrm{M}}]\\ v\in[0,1]}} F(u,v|u_S) + G(r,v|r_S) + H(v|u_S,r_S) + R(u,r|w)\ . \tag{1}$$

$F(u,v|u_S)$ and $G(r,v|r_S)$ are two data-fidelity terms, for depth and reflectance respectively. In both of them the visibility attribute v intervenes. $H(v|u_S,r_S)$ is a term depending exclusively on v, which represents the total cost of classifying input pixels as non-visible. Finally, $R(u,r|w)$ is a regularization term that penalizes the total variation of u and r, by also taking into account the color image w. In the next sections we will detail all the terms composing (1).

3.1. Visibility-weighted data-fidelity terms

The data-fitting terms in (1) are meant to enforce fidelity with the original values of depth and reflectance, u_S and r_S respectively. Deviations from the original values are more penalized if the point are considered "trustful"; conversely, for erroneous original measures (e.g., referring to hidden points) larger deviations are allowed. Therefore we use the visibility attribute v to weight the data terms. For the reflectance data-fidelity term $G(r, v|r_S)$ we have the following expression:

$$G(r, v|r_S) = \eta_2 \int_{\Omega_S} v|r - r_S| \, \mathrm{d}x_1 \, \mathrm{d}x_2 \ , \tag{2}$$

where η_2 is a coefficient weighting the term within the model, and $\mathrm{d}x_1$ and $\mathrm{d}x_2$ express the differential lengths in the two image directions. Note that in (2) an ℓ_1-norm error is used. The ℓ_1 norm is considered in substitution of the classical ℓ_2 measure of the error for its effectiveness in implicitly removing impulse noise with strong outliers (Nikolova, 2004) and its better contrast preservation (Chan and Esedoglu, 2005). As said, weighting by v relaxes the dependence on the input data for those points classified as hidden.

The depth data-fidelity term, weighted by the coefficient η_1, is further divided into two terms, as follows:

$$\begin{aligned} F(u, v|u_S) &= \eta_1 \left(\int_{\Omega_S} \max(0, u - u_S) \, \mathrm{d}x_1 \, \mathrm{d}x_2 + \int_{\Omega_S} v(\max(0, u_S - u)) \, \mathrm{d}x_1 \, \mathrm{d}x_2 \right) \\ &= F_1(u|u_S) + F_2(u, v|u_S) \ . \end{aligned} \tag{3}$$

The basic idea behind this separation is to treat differently over- and under-estimated depths. Points for which the estimated depth is greater than the original value ($u > u_S$) most likely correspond to correct input measures, where the over-estimation would be due to the surrounding presence of larger erroneous depths. The expression $\max(0, u - u_S)$ is meant to select this kind of points (over-estimated depths). As they are considered reliable, an unweighted data-fitting term, $F_1(u|u_S)$, is imposed. It is easy to see that for these points the visibility attribute v tends to converge to 1, i.e. they are the best candidates for being classified as visible points. Conversely, the hidden points to remove are sought among depth values which undergo under-estimation ($u < u_S$). These points are taken into account in the second term $F_2(u, v|u_S)$, where the ℓ_1 error is weighted by the visibility attribute. Ideally, a fraction of them, the most "problematic" ones, will be classified as hidden

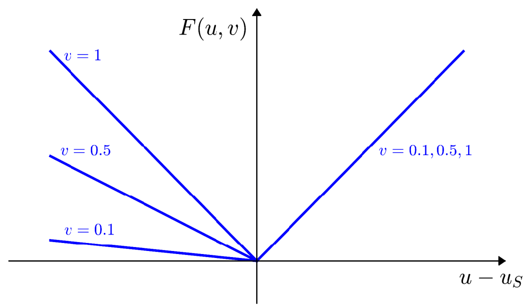


Figure 4: Depth data-fidelity cost $F(u, v|u_S)$ as a function of $u - u_S$ for different values of v ($\eta_1 = 1$ for simplicity). For over-estimated depths ($u - u_S > 0$) the cost is independent of v, whereas for $u - u_S < 0$ we have different lines as v varies.

($v = 0$) and thus not considered in the data fitting cost. Figure 4 shows graphically the depth data-fidelity cost as a function of $u - u_S$. Depending on the value of the visibility attribute v, the ℓ_1-type error $|u - u_S|$ is relaxed for negative depth deviations ($u < u_S$).

3.2. Removal cost

The second term of the model (1) is meant to penalize the total number of hidden points.

$$H(v|u_S, r_S) = \int_{\Omega_S} \alpha(u_S, r_S)(1 - v)\, \mathrm{d}x_1\, \mathrm{d}x_2 \ . \tag{4}$$

The cost of a single pixel exclusion is proportional to $1 - v$, i.e. we have the highest cost for an input pixel when it is totally excluded in the data-fitting cost ($v = 0$). We individually weight each removal cost, in order to give different importance to each decision visible/hidden. Individual weighting is given by a coefficient dependent on the original depth and reflectance values, $\alpha(u_S, r_S)$. We generally choose $\alpha = k_1 u_S + k_2 r_S$. The linear dependence of α on the depth and the reflectance "balances" the three terms of (1) depending on v, such that k_1 and k_2 appear to be constants. We will discuss their choice later in this paper (Section 5.1).

3.3. Coupled Total Variation

Depth upsampling/inpainting methods that exploit corresponding camera images often relate image edges to depth edges. This has been shown to improve the quality of the reconstructed depth images.

To couple two images in a total variation framework, we adopt the *coupled* total variation (coupled TV) of (Pierre et al., 2015):

$$\mathrm{TV}_\lambda(a,b) = \int_\Omega \sqrt{(\partial_{x_1}a)^2 + (\partial_{x_2}a)^2 + \lambda^2(\partial_{x_1}b)^2 + \lambda^2(\partial_{x_2}b)^2}\, \mathrm{d}x_1\, \mathrm{d}x_2 \ . \quad (5)$$

where λ is a coupling parameter. When $\lambda \neq 0$ the minimization of TV_λ encourages the gradient "jumps" to occur at the same locations in a and b. The coupled TV is then a way to align the edges of an image with those of a given one.

In our problem we have three types of images: a color image w, a depth image u, and a reflectance image r. Figure 3 reports in the bottom row an example of gradient magnitudes related to three images. The gradients of the input depth and reflectance images have been computed after initial interpolation of the latter. As we can clearly see from the image, the color image gradient particularly matches the reflectance one, while being rather dissimilar to the depth gradient. In turn, the reflectance gradient share some patterns, yet less prominently, with the depth one. See, e.g., the area at the base of the column, where multiple layers mix and produce a similar effect in the two gradient images. We therefore propose to match the three gradients two by two: depth with reflectance, and the same reflectance with the fixed color image. By using the previous definition of coupled TV (5), we express the regularization term as follows:

$$R(u,r|w) = \mathrm{TV}_{\lambda_1}(u,r) + \mathrm{TV}_{\lambda_2}(r,w) \ . \quad (6)$$

After detailing all the terms, our model (1) can therefore be rewritten as follows, the four terms being still distinct:

$$\begin{aligned}\min_{\substack{u\in[u_\mathrm{m},u_\mathrm{M}]\\ r\in[r_\mathrm{m},r_\mathrm{M}]\\ v\in[0,1]}} \quad & \underbrace{\eta_1\left(\int_{\Omega_S} \max(0,u-u_S) + \int_{\Omega_S} v(\max(0,u_S-u))\right)}_{F\text{: Data-fidelity for Depth}} + \underbrace{\eta_2\int_{\Omega_S} v|r-r_S|}_{G\text{: Data-fidelity for Reflectance}} \\ & + \underbrace{\int_{\Omega_S} \alpha(u_S,r_S)\,(1-v)}_{H\text{: Removal cost}} + \underbrace{\mathrm{TV}_{\lambda_1}(u,r) + \mathrm{TV}_{\lambda_2}(r,w)}_{R\text{: TV regularization}}\end{aligned} \quad (7)$$

In the next section we detail a primal-dual approach to solve (7).

4. Algorithm

The optimization problem (7) turns out to be convex, but not smooth, due to ℓ_1-type data-fidelity terms, $F(u,v|u_S)$ and $G(r,v|r_S)$, and the total variation regularization term $R(u, r|w)$. Recently, in (Chambolle and Pock, 2011) a primal-dual first-order algorithm has been proposed to solve such problems. In Section 4.1 we provide the necessary definitions for the algorithm, which is subsequently described in Section 4.2.

4.1. Discrete setting and definitions

Images, considered in Section 3 as continuous functions in $\mathbb{R}^2$, are here converted into real finite-dimensional vectors. Let M and N be the image dimensions in this discrete setting, and (i, j) the indices denoting all possible discrete locations in the Cartesian grid of size $M \times N$ ($1 \leq i \leq M$, $1 \leq j \leq N$). We then have u, u_S, r, r_S, v, w, and $\alpha \in X = \mathbb{R}^{MN}$, where X is a finite dimensional vector space equipped with a standard scalar product:

$$\langle u, v \rangle_X = \sum_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} u_{i,j} v_{i,j} \ , \quad u, v \in X \ . \tag{8}$$

The gradient of an image $u \in X$, ∇u, is a vector in the vector space X^2 with two components per pixel:

$$(\nabla u)_{i,j} = ((\nabla_H u)_{i,j}, \ (\nabla_V u)_{i,j}) \ . \tag{9}$$

We compute the gradient components via standard finite differences with Neumann boundary conditions, i.e.:

$$\begin{aligned} (\nabla_H u)_{i,j} &= \begin{cases} \frac{u_{i+1,j} - u_{i,j}}{2} & i < M \\ 0 & i = M \end{cases} \\ (\nabla_V u)_{i,j} &= \begin{cases} \frac{u_{i,j+1} - u_{i,j}}{2} & j < N \\ 0 & j = N \end{cases} \end{aligned} \ . \tag{10}$$

From the definition of gradient, it follows the expression of discrete coupled total variation, which matches the continuous one (5):

$$\mathrm{TV}_\lambda(a, b) = \sum_{\substack{1 \leq i \leq M \\ 1 \leq j \leq N}} \sqrt{(\nabla_H a_{i,j})^2 + (\nabla_V a_{i,j})^2 + \lambda^2 \ (\nabla_H b_{i,j})^2 + \lambda^2 \ (\nabla_V b_{i,j})^2} \ . \tag{11}$$

As first suggested by (Chan et al., 1999), a total variation optimization problem can be recast into a primal-dual form that makes its solution easier, by rewriting the gradient norm by means of a vector-valued dual variable. To this end, in our case we first define a "coupled gradient" operator

$\mathcal{K}_{\lambda b} : X \to Y$ ($Y = X^4$), which, applied to an image $a \in X$, expand its gradient to include the one of a reference image b according to a coupling parameter λ. I.e., we have the following element-wise definition:

$$(\mathcal{K}_{\lambda b} a)_{i,j} = ((\nabla_H a)_{i,j},\ (\nabla_V a)_{i,j},\ \lambda(\nabla_H b)_{i,j},\ \lambda(\nabla_V b)_{i,j})\ . \tag{12}$$

Thanks to the definition above, we can express alternatively the coupled total variation (11), by introducing the dual variable $p \in Y$:

$$\mathrm{TV}_\lambda (a, b) = \max_{p \in Y} \langle \mathcal{K}_{\lambda b} a, p \rangle_Y - \delta_P(p)\ , \tag{13}$$

where the scalar product in Y is defined as

$$\begin{gathered}\langle p, q \rangle_Y = \sum_{\substack{1 \le i \le M \\ 1 \le j \le N}} p^1_{i,j} q^1_{i,j} + p^2_{i,j} q^2_{i,j} + p^3_{i,j} q^3_{i,j} + p^4_{i,j} q^4_{i,j} \\ p = (p^1, p^2, p^3, p^4), \quad q = (q^1, q^2, q^3, q^4) \in Y\end{gathered}\ ,$$

δ_P denotes the indicator function of the set P

$$\delta_P(p) = \begin{cases} 0 & \text{if } p \in P \\ +\infty & \text{if } p \notin P \end{cases}\ , \tag{14}$$

and the feasibility set P for the dual variable p, is defined as

$$P = \left\{ p \in Y \mid \ \|p_{i,j}\|_2 \le 1,\ \forall i, j \right\}\ , \tag{15}$$

i.e. $\|p\|_\infty \le 1$.

We can now finally express the regularization term of our model $R(u, r|w)$ (6) as the maximization over two dual variables. We then have:

$$R(u, r|w) = \max_{p \in Y} \max_{q \in Y} \mathcal{K}_{\lambda_1 r} u + \mathcal{K}_{\lambda_2 w} r - \delta_P(p) - \delta_Q(q)\ . \tag{16}$$

This will let us formulate a discrete version of our joint inpainting problem (7), which falls into the primal-dual optimization framework. As for the other terms in (7), rewritten in discrete notation, we have:

$$
\begin{aligned}
F_1(u|u_S) &= \eta_1 \sum_{\substack{1\le i\le M\\1\le j\le N}} \Phi_{i,j} \max(0, u_{i,j} - u_{S\,i,j}) \\
F_2(u,v|u_S) &= \eta_1 \sum_{\substack{1\le i\le M\\1\le j\le N}} \Phi_{i,j} v_{i,j} \max(0, u_{S\,i,j} - u_{i,j}) \\
G(r,v|r_S) &= \eta_2 \sum_{\substack{1\le i\le M\\1\le j\le N}} \Phi_{i,j} v_{i,j} |r_{i,j} - r_{S\,i,j}| \\
H(v|u_S, r_S) &= \sum_{\substack{1\le i\le M\\1\le j\le N}} \Phi_{i,j} \alpha_{i,j} (1 - v_{i,j})
\end{aligned}
\quad , \tag{17}
$$

where Φ is a binary mask indicating the initial known pixels, i.e. belonging to the sparse image support Ω_S.

4.2. A primal-dual algorithm

Thanks to the previous definitions, we can express our model (7) in the form of the following saddle-point problem, which is an extension (including two extra variables) of the one presented in (Pierre et al., 2015):

$$
\begin{aligned}
\min_{u\in X} \min_{r\in X} \min_{v\in X} \max_{p\in Y} \max_{q\in Y} \{ & \langle K_1 u | p \rangle + \langle K_2 r | q \rangle - D_1^*(p) - D_2^*(q) \\
& + A(u) + B(r) + a(u,v) + b(r,v) + C(v) \} \ .
\end{aligned} \tag{18}
$$

It is a primal-dual problem with three primal variables (u, r, and v) and two dual variables (p and q) that evolve independently. Each dual variable is particularly linked to the gradient of a primal variable, i.e. p to u, and q to r. D_1^*, D_2^*, A, B, and C are convex functions; a and b are convex w.r.t. each of its respective variables. Globally, the functional is not convex w.r.t. the triplet (u, r, v). By relating (7) and (18), we have the following equivalences:

- $K_1 u = \mathcal{K}_{\lambda_1 r} u$;
- $K_2 r = \mathcal{K}_{\lambda_2 w} r$;
- $D_1^*(p) = \delta_P(p)$;
- $D_2^*(q) = \delta_Q(q)$;
- $A(u) = F_1(u|u_S) + \delta_{[u_\mathrm{m}, u_\mathrm{M}]}(u)$;
- $B(r) = \delta_{[r_\mathrm{m}, r_\mathrm{M}]}(r)$;
- $a(u,v) = F_2(u,v|u_S)$;
- $b(r,v) = G(r,v|r_S)$;

- $C(v) = H(v|u_S, r_S) + \delta_{[0,1]}(v)$.

An algorithm to solve (18) can be derived within the primal-dual optimization framework of (Chambolle and Pock, 2011). It consists in a unique loop, where all variables are alternatively updated via proximal operators (see Algorithm 1). The algorithm takes as inputs the initial estimates of the complete depth and reflectance images (u_0 and r_0, respectively), and the reference intensity image w. It also requires three parameters inherent to the algorithm: σ and τ, which are related to each other by the relation $16\tau\sigma \leq 1$ (Chambolle and Pock, 2011), and ρ, which is a parameter regulating the update speed of v.

Algorithm 1 Primal-dual based algorithm for depth and reflectance joint inpainting.

Inputs:
  $u_0, r_0, w, \sigma, \rho, \tau$
Initialize:
  $u^0, \bar{u}^0 \leftarrow u_0,\ r^0, \bar{r}^0 \leftarrow r_0,\ v^0_{i,j} \leftarrow 0.5,$
  $p^0 \leftarrow (\nabla u_0, \lambda_1 \nabla r_0),\ q^0 \leftarrow (\nabla r_0, \lambda_2 \nabla w)$
for $n = 0, 1, \ldots$ **do**
  $p^{n+1} \leftarrow \mathrm{prox}_{\sigma D_1^*}(p^n + \sigma K_1 \bar{u}^n)$
  $q^{n+1} \leftarrow \mathrm{prox}_{\sigma D_2^*}(q^n + \sigma K_2 \bar{r}^n)$
  $v^{n+1} \leftarrow \mathrm{prox}_{\rho a(\bar{u}^n, \cdot) + \rho b(\bar{r}^n, \cdot) + \rho C}(v^n)$
  $u^{n+1} \leftarrow \mathrm{prox}_{\tau A + \tau a(\cdot, v^{n+1})}(u^n - \tau K_1^* p^{n+1})$
  $r^{n+1} \leftarrow \mathrm{prox}_{\tau B + \tau b(\cdot, v^{n+1})}(r^n - \tau K_2^* q^{n+1})$
  $\bar{u}^{n+1} \leftarrow 2u^{n+1} - u^n$
  $\bar{r}^{n+1} \leftarrow 2r^{n+1} - r^n$
end for

Algorithm 1 involves the computation of the adjoints to the linear operators K_1 and K_2 (the coupled gradient operators). It is known that the adjoint of the gradient operator is the negative divergence operator ($\nabla^* = -\operatorname{div}$). In our case, the adjoint to the coupled gradient operator $K_1 : X \to Y$ is a linear operator $K_1^* : Y \to X$ consisting in the negative divergence computed only on the two first components of a four-component dual variable $p \in Y$. These components are in fact the ones related to the primal variable to which the coupled gradient operator has been applied. We then have the following definition for $K_1^* p$ (the same definition stands for $K_2^* p$):

$$K_1^* p = -(\nabla_H(p^1) + \nabla_V(p^2)) \ . \tag{19}$$

Closed-form expressions for the update rules in Algorithm 1 can be easily computed by applying the definition of proximal operator (see Appendix A). The resulting expressions are reported here below, where $\mathcal{P}$ denotes the projection operation over a given real interval, i.e. values are clipped if exceeding the interval limits. While the proximal operators for the update of the dual variables p and q come straight from the definitions of the feasibility sets P and Q, we report details about the derivation of the other proximal operators in Appendix A.

$$p = \operatorname{prox}_{\sigma D_1^*}(\tilde{p}) \iff p_{i,j} = \frac{\tilde{p}_{i,j}}{\max(1, |\tilde{p}_{i,j}|)} \tag{20}$$

$$q = \operatorname{prox}_{\sigma D_2^*}(\tilde{q}) \iff q_{i,j} = \frac{\tilde{q}_{i,j}}{\max(1, |\tilde{q}_{i,j}|)} \tag{21}$$

$$\begin{aligned}&\operatorname{prox}_{\rho a(\bar{u},\cdot)+\rho b(\bar{r},\cdot)+\rho C}(\tilde{v}) = \\ &\qquad\begin{cases} \mathcal{P}_{[0,1]}\left(\tilde{v}\right) & \text{if} \quad \Phi_{i,j} = 0 \\ \mathcal{P}_{[0,1]}\left(\tilde{v} + \rho\alpha - \rho\eta_2 |\bar{r} - r_S|\right) & \text{if} \quad \Phi_{i,j} = 1, \ \bar{u}_{i,j} \geq u_{S\,i,j} \\ \mathcal{P}_{[0,1]}\left(\tilde{v} + \rho\alpha - \rho\eta_1(u_S - \bar{u}) - \rho\eta_2 |\bar{r} - r_S|\right) & \text{if} \quad \Phi_{i,j} = 1, \ \bar{u}_{i,j} < u_{S\,i,j} \end{cases}\end{aligned} \tag{22}$$

$$\operatorname{prox}_{\tau A + \tau a(\cdot,v)}(\tilde{u}) = \begin{cases} \mathcal{P}_{[u_m,u_M]}\left(\tilde{u}\right) & \text{if} \quad \Phi_{i,j} = 0 \\ \mathcal{P}_{[u_m,u_M]}\left(\tilde{u} - \tau\eta_1\right) & \text{if} \quad \Phi_{i,j} = 1, \ \tilde{u}_{i,j} > u_{S\,i,j} + \tau\eta_1 \\ \mathcal{P}_{[u_m,u_M]}\left(\tilde{u} + v\tau\eta_1\right) & \text{if} \quad \Phi_{i,j} = 1, \ \tilde{u}_{i,j} < u_{S\,i,j} - v\tau\eta_1 \\ \mathcal{P}_{[u_m,u_M]}\left(u_S\right) & \text{otherwise} \end{cases} \tag{23}$$

$$\operatorname{prox}_{\tau B + \tau b(\cdot,v)}(\tilde{r}) = \begin{cases} \mathcal{P}_{[r_m,r_M]}\left(\tilde{r}\right) & \text{if} \quad \Phi_{i,j} = 0 \\ \mathcal{P}_{[r_m,r_M]}\left(\tilde{r} - v\tau\eta_2\right) & \text{if} \quad \Phi_{i,j} = 1, \ \tilde{r}_{i,j} > r_{S\,i,j} + v\tau\eta_2 \\ \mathcal{P}_{[r_m,r_M]}\left(\tilde{r} + v\tau\eta_2\right) & \text{if} \quad \Phi_{i,j} = 1, \ \tilde{r}_{i,j} < r_{S\,i,j} - v\tau\eta_2 \\ \mathcal{P}_{[r_m,r_M]}\left(r_S\right) & \text{otherwise} \end{cases} \tag{24}$$

The operations indicated in the proximal operators are pixel-wise, although the pixel coordinates have not been made explicit for clearer reading.

5. Experimental results

The algorithm presented in Section 4 is evaluated with a realistic data set acquired in an urban scenario, composed of lidar measures and camera-originated images. We also assess the quality of the visibility estimation task, which is a crucial characteristic of our algorithm. Before showing results and comparisons, in Section 5.1 we motivate some critical choices in terms of model and algorithmic parameters.

5.1. Parameters of the algorithm and model choices

Our finally resulting joint inpainting model (7) consists of four terms: two data-fidelity terms, $F(u, v|u_S)$ and $G(r, v|r_S)$, a "removal" cost depending solely on the variable v, $H(v|u_S, r_S)$, and the two-fold regularization term $R(u, r|w)$. As discussed in Section 3.1, for the data-fidelity terms we opt for an ℓ_1 measure of the error, in order to promote more contrasted solutions (Chan and Esedoglu, 2005). The visibility attribute v weights the data matching cost of each single pixel (data matching is more and more relaxed, as v tends to zero, i.e. when that particular point is considered to be excluded). However, over-estimated depths ($u > u_S$) are not weighted by v but are fully penalized. These values relate to pixels where either there is noise on a visible point that is slightly corrected ($u - u_S$ is small), or the value u_S represents an outlier (e.g. it is due to a mobile object). At present, we do not have a way to handle the latter case.

In $H(v|u_S, r_S)$ (4), each point removal cost is the product between $(1 - v)$ (the level of "invisibility" of the point) and a coefficient α depending on the local input depth and reflectance: $\alpha = k_1 u_S + k_2 r_S$. This choice has been made in order to balance all terms in (7) where v appears. Let us now observe the "complete" update rule for v (last case of (22), i.e. for points with under-estimated depth). According to it, we have that at each iteration v is incremented/decremented by a quantity $\Delta v = \rho\,(\alpha - \eta_1 \Delta u - \eta_2 \Delta r)$. Let us suppose that the fluctuations on depth are significantly larger than the fluctuations on reflectance (the appearance of a hidden point can cause a big "jump" in depth, while the reflectance values might still be similar. For the sake of simplicity we can then adjust the value of α only on the basis of the depth input value. The proposed simplified expression for α is then:

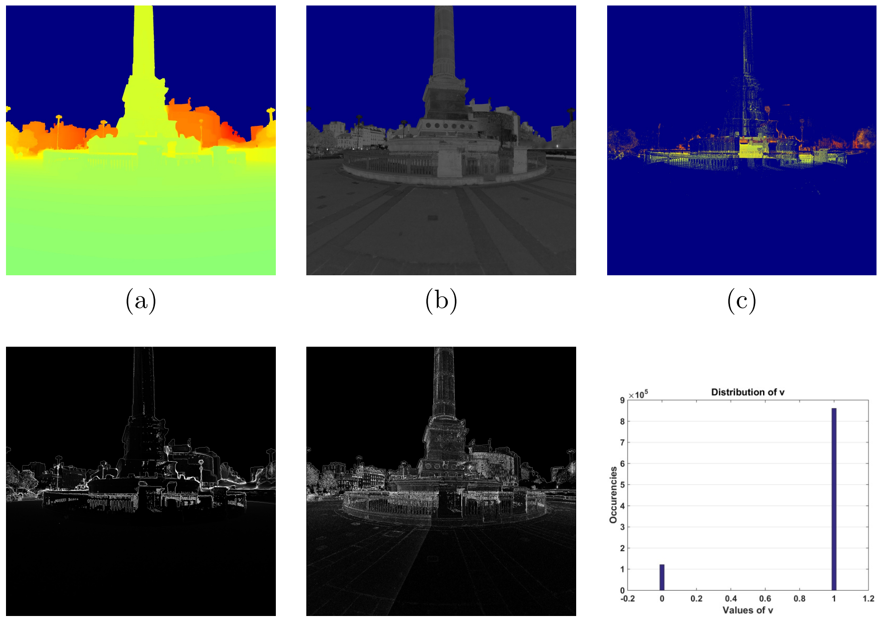
$$\alpha = k u_S\,. \tag{25}$$

With the assumptions made we therefore have $\Delta v \propto (k u_S - \eta_1 \Delta u)$. The attribute v for a certain pixel increases (it gets a higher confidence as a visible point) if $\frac{\Delta u}{u_S} < \frac{k}{\eta_1}$, i.e. if the relative depth deviation is below a certain threshold. k is an a-dimensional parameter that contributes determining this threshold. Conversely, v decreases for relative depth deviations exceeding the threshold. As for the update of v for points with over-estimated depths (second case of (22)), if we hypothesize that α, adjusted on depth, is large enough w.r.t. the reflectance deviation, we have that v progressively tends to one (unless large absolute reflectance deviations occur).

As for the regularization term $R(u, r|w)$, we proposed in Section 3.3 to combine two distinct coupled total variation terms: $\mathrm{TV}_{\lambda_1}(u, r)$ (depth is individually coupled with reflectance) and $\mathrm{TV}_{\lambda_2}(r, w)$ (reflectance is individually coupled with the color image). By having two separate coupled TV terms, each one encoded by a dual variable that evolves independently from the other one, the reflectance gradient is constantly brought back to the reference gradient of the color image. At the same time the "correct" gradient information is transferred to the depth via the second term. Figure 5 shows an example of results obtained with the algorithm for the same test case as Figure 3.

For the example test of Figure 5, as well as for all the results reported hereinafter, the following parameters, found with multiple tests, have been used to characterize the model (7): $\eta_1 = 1.7$, $\eta_2 = 50$, $k = 0.05$ (the coefficient determining α according to (25)), $\lambda_1 = 0.5$, $\lambda_2 = 1$. These values have been found empirically by letting them vary one by one and observing the obtained visual results. The two data terms $F(u, v|u_S)$ and $G(r, v|r_S)$ are attributed different weights. The larger coefficient assigned to the reflectance data term ($\eta_2 > \eta_1$) means that a greater data fidelity is imposed on reflectance. Depth values have instead a greater "freedom" in deviating from their original values. The two coupling parameters λ_1 and λ_2 being in the same order of magnitude, it shows that the two coupling terms have a similar importance. As for the parameters, inherent to the primal-dual optimization scheme (Algorithm 1), the following values have been set after testing: $\rho = 10$, $\tau = 0.004$, $\sigma = 14$.

If we observe the input sparse depth image of Figure 3, we see that the major problems come from the fact that depth values referring to the building behind the column appear mixed with foreground depths. With our algorithm we are able to resolve these conflicts, as we can see in the inpainted depth image (Figure 5a). Part of the input points have in fact been removed, i.e. classified as non-visible ($v = 0$). Figure 5c reports the locations of such points in the original depth image. From the histogram of



(a) (b) (c)

(d) (e) (f)

Figure 5: Output of the proposed algorithm for the image *Column1*: (a) Inpainted depth, (b) Inpainted reflectance, (c) Removed points ($v = 0$), (d) Final depth gradient, (e) Final reflectance gradient, (f) Final histogram of v.

the values of v (Figure 5f) it is evident that the algorithm produces a bi-partition of the points according to their visibility attribute. Figure 5 shows also the inpainted reflectance and the final depth and reflectance gradients. By comparing the latter to the original gradients (Figure 3), we can observe that they end up incorporating elements of the color image gradient, while removing erroneous edges. In the next section we will present more results obtained with our algorithm, also in comparison to other inpainting methods.

5.2. Results with urban data

We consider a data set acquired by a MMS system (Paparoditis et al., 2012) at *Place de la Bastille*, Paris, consisting of one lidar point cloud in the order of one billion of points and hundreds of optical image simultaneously acquired by 5 cameras mounted on the vehicle. Given a reference optical image, we project onto it the available lidar points to form the initial depth and reflectance incomplete images. Note that not all the points are effectively visible from the image view point. The incomplete depth and reflectance images, along with the reference color image chosen, represent the input of the algorithm (u_S, r_S, and w respectively).

Figures 6–9 present results for four images (cropped w.r.t. the full size) of the data set: *Column1*, *Column2*, *Buildings1*, *Buildings2*. For each reference image, the input sparse depth and reflectance images, obtained via projection, are shown, as well as the inpainted depth and reflectance images, obtained with four different methods. For the output depth images of Figure 8 and 9 we added some shading by modulating the color intensity of each pixel based on the zenith angle of the normal vector, to emphasize high-frequency changes. Moreover, for the inpainted depths, an alternative view of the resulting 3-D point cloud is proposed, where the coordinates of the points are retrieved thanks to the computed depths and color texture is applied to enrich the points. A color box is overlaid to the first of these 3-D views to highlight areas where the comparison between the different methods is particularly significant.

Our algorithm, presented in Section 4, gives as output the two inpainted images u and r. As for the produced depth image, our algorithm is visually compared with nearest neighbor (NN) interpolation, the anisotropic total generalized variation (*ATGV*) method of Ferstl et al. (2013), and our previous depth inpainting method (Bevilacqua et al., 2016), which does not rely on reflectance information. We refer to the latter as Depth Inpainting with Visibility Estimation (*DIVE*). The optimization problem of DIVE is the following:

$$\min_{\substack{u\in[u_{\mathrm{m}},u_{\mathrm{M}}]\\ v\in[0,1]}} \eta \int_{\Omega_S} (\max(0,u-y))^2 \,\mathrm{d}x_1\,\mathrm{d}x_2 + \eta \int_{\Omega_S} v(\max(0,y-u))^2 \,\mathrm{d}x_1\,\mathrm{d}x_2 + \int_{\Omega_S} (ku_S)^2(1-v)\,\mathrm{d}x_1\,\mathrm{d}x_2 + \mathrm{TV}_\lambda(u,w) \ . \quad (26)$$

The DIVE problem can be related to our proposed model (7), if we consider in the latter $\eta_1 = \eta$, $\eta_2 = 0$, $\lambda_1 = \lambda$, and we suppress the coupled TV term related to the reflectance (depth is instead coupled directly with the color image). Moreover, in (26) we have a ℓ_2-norm data fidelity term; as a consequence of that, the coefficient of the removal cost term follows a quadratic law (we have $\alpha = (ku_S)^2$, instead of $\alpha = ku_S$, as in (7)).

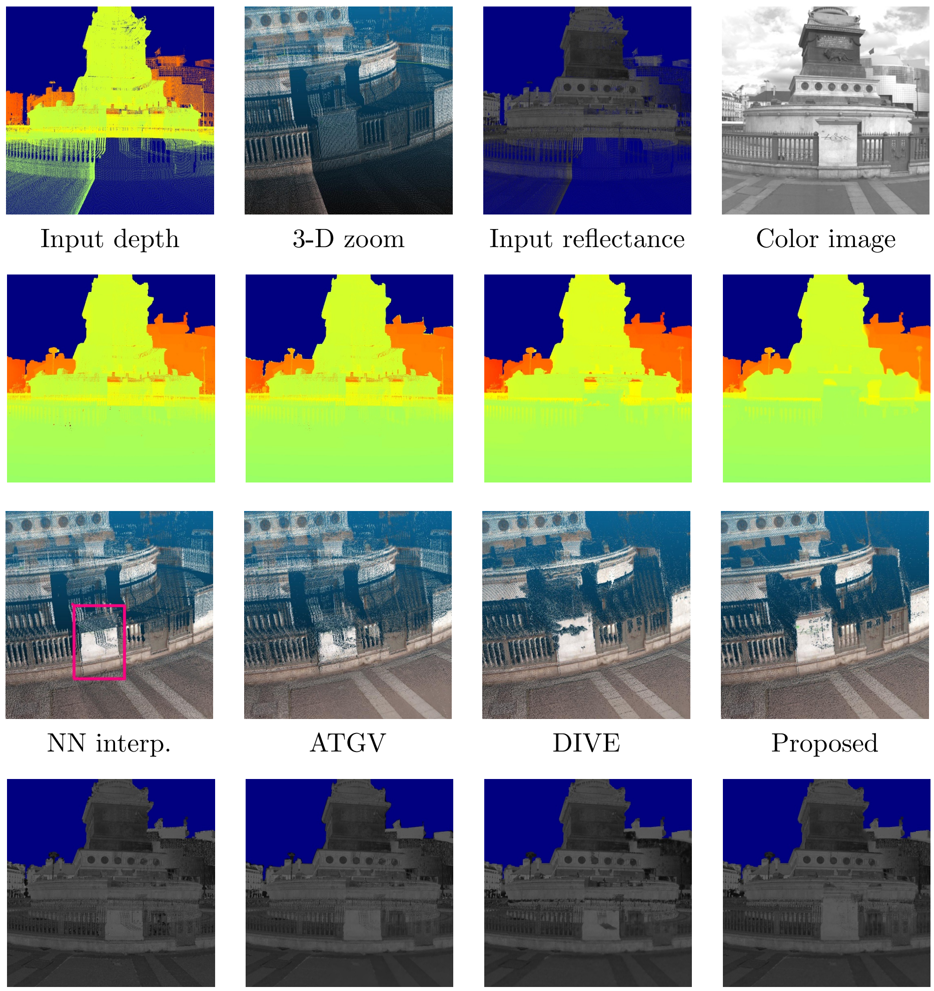
As for the produced reflectance image, our algorithm is compared with nearest neighbor (NN) interpolation, the ATGV method of Ferstl et al. (2013) applied to reflectance, and a reduced version of our model (7) limited to reflectance. We refer to this method as Reflectance Inpainting with Visibility Estimation (*RIVE*). The RIVE method is derived from the solution of the following optimization problem:

$$\min_{\substack{r\in[r_{\mathrm{m}},r_{\mathrm{M}}]\\ v\in[0,1]}} \eta \int_{\Omega_S} v|r-r_S|\,\mathrm{d}x_1\,\mathrm{d}x_2 + \int_{\Omega_S} (kr_S)(1-v)\,\mathrm{d}x_1\,\mathrm{d}x_2 + \mathrm{TV}_\lambda\,(r,w) \ . \quad (27)$$

Also in this case we can derive the considered problem (RIVE) as a simplified version of our proposed model (7), where $\eta_1 = 0$, $\eta_2 = \eta$, $\lambda_2 = \lambda$, and the coupled TV term related to depth is suppressed. Moreover, the coefficient of the removal cost, while still following a linear law, here depends on the input reflectance r_S.

The four examples reported show the better performance of our algorithm in generating complete depth and reflectance images from real lidar measures. Results with the image *Column1*, reported in Figure 6, particularly prove the effectiveness of our algorithm in detecting and removing hidden points appearing in the front, thus producing inpainted images correct from the image view point. These points, in yellow/orange according to the color code used for depth, appear mixed to visible points belonging to the column and the fence. By looking at the depth images generated (row (b)), our algorithm is the only one which is able to remove the misleading points and correctly reconstruct the foreground depth plane. This is even more visible by observing the main marble pole highlighted in the 3-D views (row (c)). While other methods are not able to reconstruct the pole, since "distracted" by the interfering background depths, the reconstruction is better performed in our case. Results on the reflectance image confirm the trend. By observing again the main marble pole, we clearly see that the reflectance is better inpainted. This is possible thanks to the joint use of depth information, which helps detecting hidden points by leveraging depth over- and under-estimations, and the coupling with the color image gradient, which helps correctly restoring the edges. Similar considerations can be made for the image *Column2* (visual results are reported in Figure 7). Here the box overlaid on the 3-D views indicates an area where points, non-visible from the reference image view point, should be removed. The removal of these points, as well as the inpainting of depth and reflectance, is performed more efficiently by our method.

Figures 8 and 9 show results w.r.t. two other images taken peripherally to the scene. For the image *Buildings1*, we can observe that with our algorithm the inpainted depth and reflectance images looks more satisfactory, the pole



(a) Input depth | 3-D zoom | Input reflectance | Color image

(b)

(c) NN interp. | ATGV | DIVE | Proposed

(d) NN interp. | ATGV | RIVE | Proposed

Figure 6: Visual results for the image *Column1*. Row (a) shows the related input images: depth (with a 3-D zoom), reflectance, and reference color image. Rows (b) and (c) report the results obtained in terms of inpainted depth images (with related 3-D zoomed-in view) with the algorithms indicated below. Row (d) shows the inpainted reflectance images obtained with different methods, our proposed method always reported as last.

on the left being completely unveiled as a foreground element. The box overlaid on the 3-D views highlights a part of the scene where the depth values of two trees interfere. Our proposed algorithm (as well as the DIVE method (Bevilacqua et al., 2016)) makes a correct distinction between the two depth layers. Figure 9, reporting results related to the image *Buildings2*,

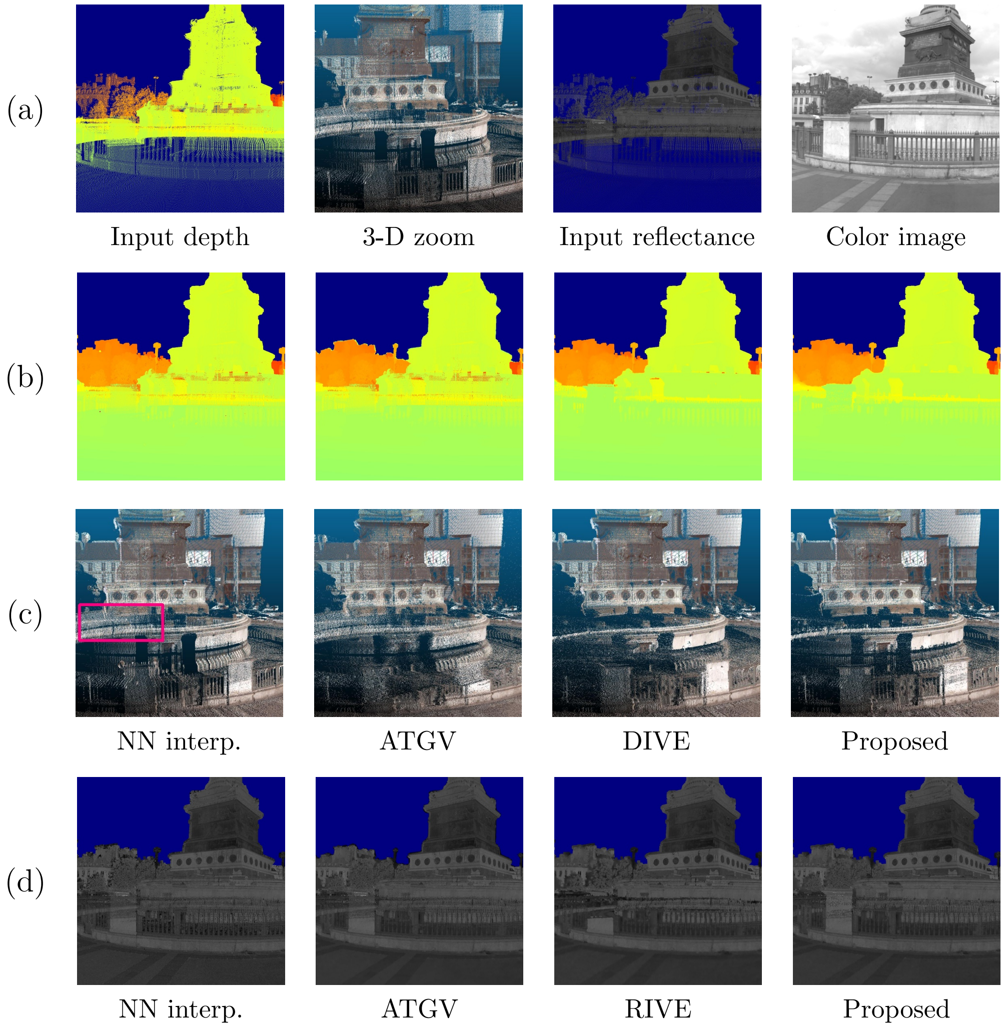


Figure 7: Visual results for the image *Column2*. Row (a) shows the related input images: depth (with a 3-D zoom), reflectance, and reference color image. Rows (b) and (c) report the results obtained in terms of inpainted depth images (with related 3-D zoomed-in view) with the algorithms indicated below. Row (d) shows the inpainted reflectance images obtained with different methods, our proposed method always reported as last.

presents the problem of wrong lidar measures appearing in the front. Our method turns out to be the most effective in clearing out these points, as also shown in the area highlighted by the box.

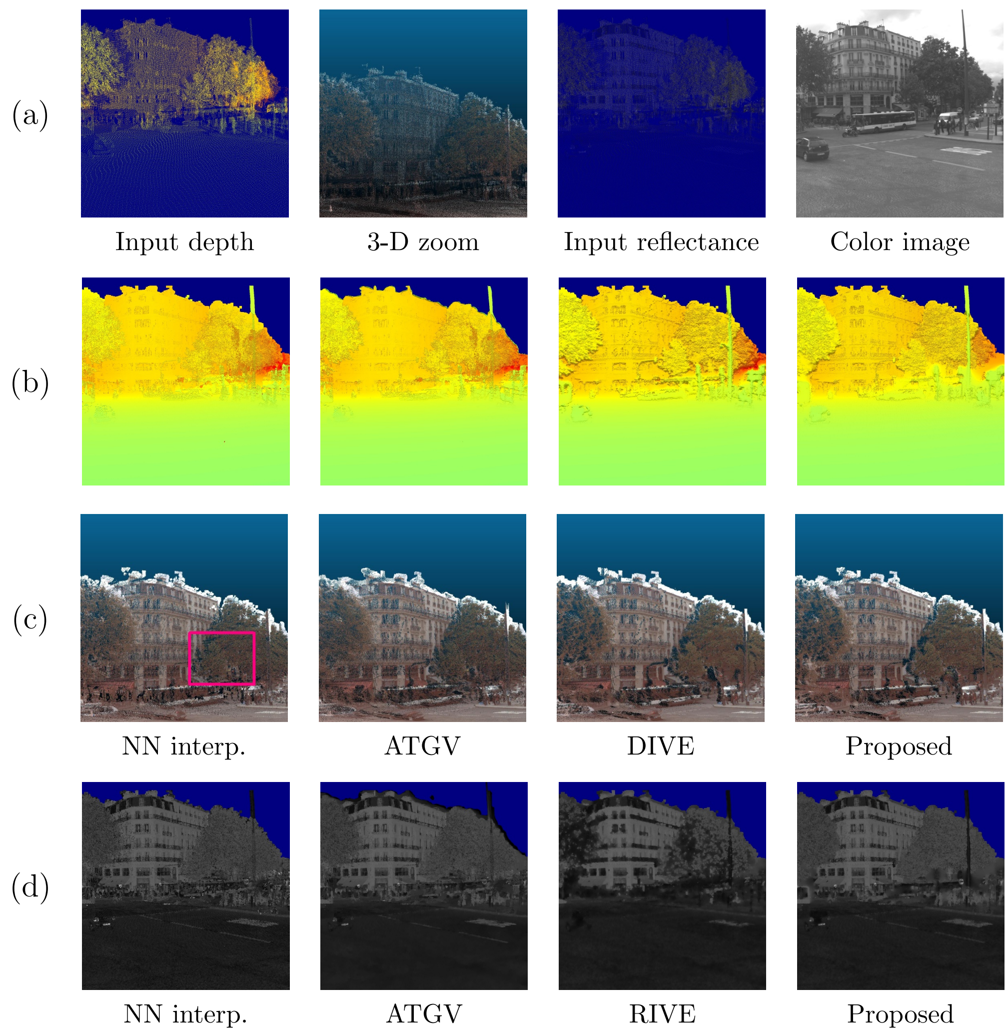


Figure 8: Visual results for the image *Buildings1*. Row (a) shows the related input images: depth (with a 3-D zoom), reflectance, and reference color image. Rows (b) and (c) report the results obtained in terms of inpainted depth images (with related 3-D zoomed-in view) with the algorithms indicated below. Row (d) shows the inpainted reflectance images obtained with different methods, our proposed method always reported as last.

5.3. Performance on visibility estimation

While in the previous section we evaluated the performance of the algorithm in terms of produced inpainted images u and r, we now want to assess the quality of the third output of the algorithm, i.e. v, the visibility attribute

As visibility is estimated while performing the depth and reflectance es-

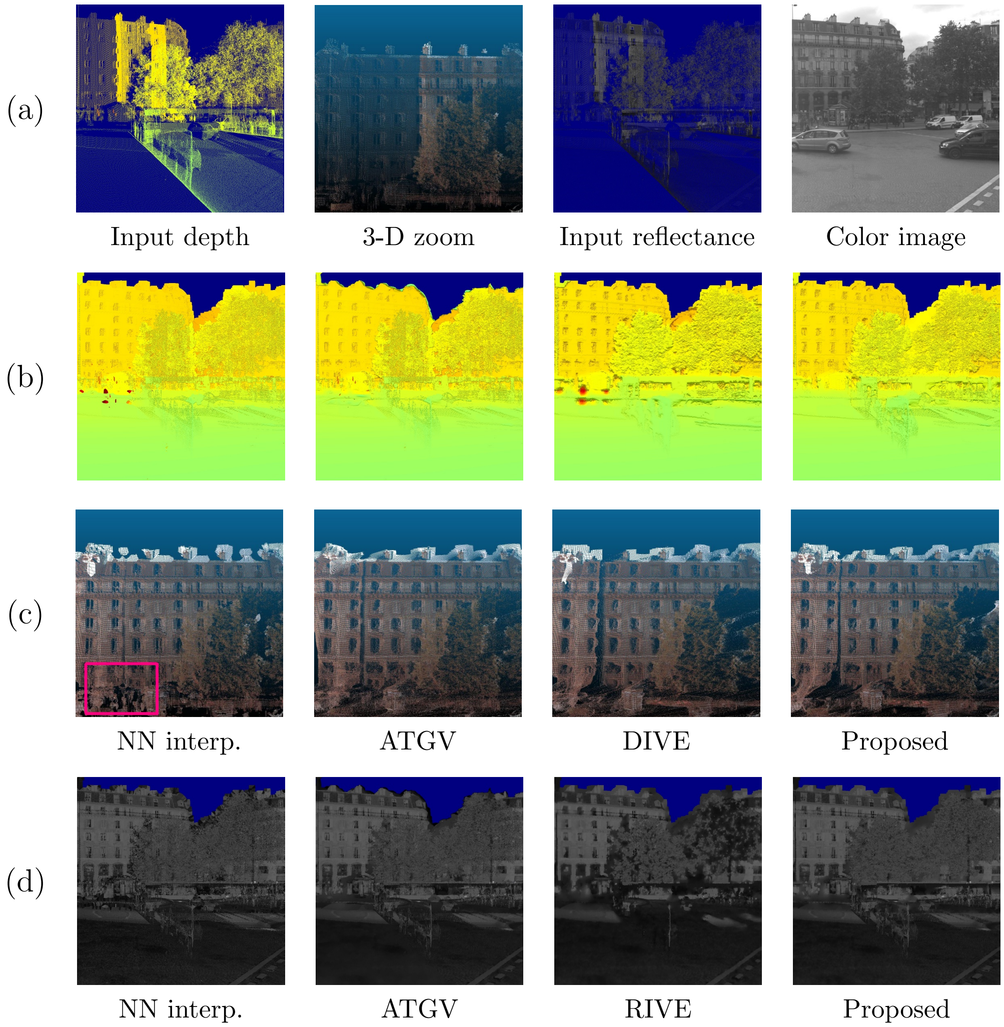
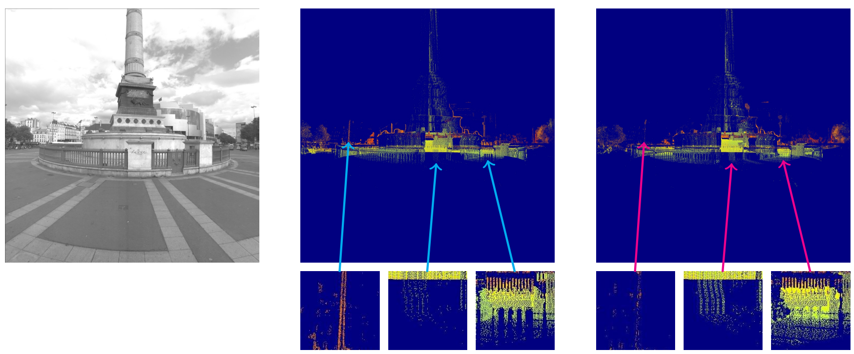


Figure 9: Visual results for the image *Buildings2*. Row (a) shows the related input images: depth (with a 3-D zoom), reflectance, and reference color image. Rows (b) and (c) report the results obtained in terms of inpainted depth images (with related 3-D zoomed-in view) with the algorithms indicated below. Row (d) shows the inpainted reflectance images obtained with different methods, our proposed method always reported as last.

timation, we can say that our algorithm fuses two problems: hidden point removal (HPR) and inpainting. Typically HPR is, instead, possibly performed as a preliminary operation. For HPR "stand-alone" the state of the art is represented by variations of (Katz et al., 2007) that relate the visible point set to the convex hull of a viewpoint-dependent transformation of it, discarding points based on a concavity threshold as seen from the view point. While this approach is effective, there is in general no globally satisfactory concavity threshold that would both correctly detect hidden surfaces and keep background points close to foreground silhouettes. To compare the two strategies for estimating visibility (the dedicated operation of (Katz et al., 2007) and our "soft" estimation), we show an example in Figure 10, related to the image *Column1*. In our case, we consider hidden points those depth values that are assigned $v = 0$ at the end of the algorithm. As for (Katz et al., 2007), a concavity parameter equal to 4 has been chosen after tuning.



(a) Intensity image (b) (Katz et al., 2007) (c) Proposed method

Figure 10: Detected hidden points in the case of the image *Column1*, by the state-of-the-art method of (Katz et al., 2007) and our method. The three patches below each image represent zoomed-in areas of the images themselves at same locations.

The images obtained shows that the "quality" of the visibility estimation process is comparable, if not higher with our method. If we observe closely the zoomed-in areas in Figure 10, in fact, we can see that the HPR method wrongly selects points around the silhouettes (see first patch), while sometimes missing the detection of actual hidden points (see last two patches).

6. Conclusion

In this paper we presented a novel strategy to jointly inpaint depth and reflectance images with the guidance of a co-registered color image, and by simultaneously estimating a visibility attribute for each pixel. The problem studied and the proposed approach are particularly suited for data sets acquired by Mobile Mapping Systems (MMS): vehicles that can easily image urban scenes by means of optical cameras and lidar sensors. By projecting the 3D lidar points onto a chosen reference image, we obtain depth and reflectance images, which suffer of practical issues due to the big diversity of the lidar an optical sensor acquisitions. By estimating visibility, we aim at solving one of these issues, i.e. the appearance (in depth and reflectance) of parts of objects non-visible from the image view point, but captured by the lidar sensor. Those points are meant to be detected by our algorithm and thus discarded in the inpainting process. The proposed approach consists in a variational optimization problem, where three variables (depth, reflectance, and visibility) are simultaneously estimated. As a regularization term, a two-fold coupled total variation (TV) term is proposed, where the gradients of depth, reflectance and color image are matched two by two, by leveraging the inherent correlation between them. The proposed algorithm is compared, in terms of inpainted images, to other inpainting algorithms, which do not take into account the simultaneous detection of possibly erroneous measures. The clear superiority of the proposed method w.r.t. the latter proves that the visibility estimation is a necessary step. Another comparison is made with a simplified version of the algorithm, which accounts for visibility but considers alternatively either depth or reflectance. The worse performance of the simplified algorithm indicates that the joint exploitation of depth and reflectance is a key aspect for the success of the algorithm. The mutual benefit comes from the fact that depth is particularly important for the visibility estimation task; in turn, reflectance is crucial in restoring the correct edges, via coupling with the color image. Future work will continue in the direction of solving practical issues with lidar-based images to inpaint. Notably, another problem is related to disocclusions: the detection of mobile objects is in this case necessary to prevent occlusions in the produced depth and reflectance images.

Appendix A. Derivation of the proximal operators in Algorithm 1

In this section we detail the derivation of the closed-form expressions of the proximal operators for the update of three primal variables (v, u, and r) in Algorithm 1, as listed in Section 4.2. Let $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ be a closed proper convex function. The proximal operator or mapping $\text{prox}_{\lambda f} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ of f (Parikh and Boyd, 2013) is defined by:

$$\text{prox}_f(v) = \underset{x \in \mathbb{R}^n}{\arg\min}\, f(x) + \frac{1}{2} \|x - v\|_2^2 \ . \tag{A.1}$$

Broadly speaking, the proximal operator of a function is a mathematical tool that allows to make an approximation to a certain value, while making a compromise between the accuracy of the approximation and a cost given by the function itself.

Given the general definition (A.1), we can derive the expressions for the proximal operators of the functions considered in our algorithm. We have that the operations involved are independent for each coordinate of the processed images. Therefore, the expressions reported below are to be intended per coordinate, although the spatial indices indicating a particular pixel location are not specified for brevity.

- $\boxed{\mathrm{prox}_1 = \mathrm{prox}_{\rho a(\bar{u},\cdot)+\rho b(\bar{r},\cdot)+\rho C}(\tilde{v})}$

➪

$$\begin{aligned}\mathrm{prox}_1 = \arg\min_v \frac{1}{2}(v-\tilde{v})^2 + \rho\eta_1\Phi v \max(0, u_S - \bar{u}) \\ + \rho\eta_2\Phi v|\bar{r} - r_S| + \rho\alpha\Phi(1-v) + \delta_{[0,1]}(v)\end{aligned} \tag{A.2}$$

✦ If $\Phi_{i,j} = 0$ (point out of the sparse domain), we trivially have:

$$\begin{aligned}\mathrm{prox}_1 &= \arg\min_v \frac{1}{2}(v-\tilde{v})^2 + \delta_{[0,1]}(v) \\ &= \mathcal{P}_{[0,1]}(\tilde{v})\ .\end{aligned} \tag{A.3}$$

✦ If $\Phi_{i,j} = 1$ and $\bar{u}_{i,j} \geq u_{S\,i,j}$, we have:

$$\begin{aligned}\mathrm{prox}_1 &= \arg\min_v \frac{1}{2}(v-\tilde{v})^2 + \rho\eta_2 v|\bar{r}-r_S| + \rho\alpha(1-v) + \delta_{[0,1]}(v) \\ &= \arg\min_v \frac{1}{2}v^2 - v\tilde{v} + \rho\eta_2 v|\bar{r}-r_S| - \rho\alpha v + K + \delta_{[0,1]}(v) \\ &= \arg\min_v \frac{1}{2}v^2 - v\left(\tilde{v} + \rho\alpha - \rho\eta_2|\bar{r}-r_S|\right) + K + \delta_{[0,1]}(v) \\ &= \arg\min_v \frac{1}{2}\left[v - \left(\tilde{v} + \rho\alpha - \rho\eta_2|\bar{r}-r_S|\right)\right]^2 + K' + \delta_{[0,1]}(v) \\ &= \mathcal{P}_{[0,1]}\left(\tilde{v} + \rho\alpha - \rho\eta_2|\bar{r}-r_S|\right)\ .\end{aligned} \tag{A.4}$$

✦ If $\Phi_{i,j} = 1$ and $\bar{u}_{i,j} < u_{S\,i,j}$, we have:

$$\begin{aligned}
\mathrm{prox}_1 &= \arg\min_v \frac{1}{2}(v-\tilde{v})^2 + \rho\eta_1 v(u_S - \bar{u}) + \rho\eta_2 v|\bar{r} - r_S| \\
&\quad +\rho\alpha(1-v) + \delta_{[0,1]}(v) \\
&= \arg\min_v \frac{1}{2}v^2 - v\tilde{v} + \rho\eta_1 v(u_S - \bar{u}) + \rho\eta_2 v|\bar{r} - r_S| - \rho\alpha v \\
&\quad +K + \delta_{[0,1]}(v) \\
&= \arg\min_v \frac{1}{2}v^2 - v\left(\tilde{v} + \rho\alpha - \rho\eta_1(u_S - \bar{u}) - \rho\eta_2|\bar{r} - r_S|\right) \\
&\quad +K + \delta_{[0,1]}(v) \\
&= \arg\min_v \frac{1}{2}\left[v - \left(\tilde{v} + \rho\alpha - \rho\eta_1(u_S - \bar{u}) - \rho\eta_2|\bar{r} - r_S|\right)\right]^2 \\
&\quad +K' + \delta_{[0,1]}(v) \\
&= \mathcal{P}_{[0,1]}\left(\tilde{v} + \rho\alpha - \rho\eta_1(u_S - \bar{u}) - \rho\eta_2|\bar{r} - r_S|\right) \ .
\end{aligned} \tag{A.5}$$

✦ Summing up, we have:

$$\mathrm{prox}_1 = \begin{cases} \mathcal{P}_{[0,1]}\left(\tilde{v}\right) & \text{if} \quad \Phi_{i,j} = 0 \\ \mathcal{P}_{[0,1]}\left(\tilde{v} + \rho\alpha - \rho\eta_2|\bar{r} - r_S|\right) & \text{if} \quad \Phi_{i,j} = 1,\ \bar{u}_{i,j} \leq u_{S\,i,j} \\ \mathcal{P}_{[0,1]}\left(\tilde{v} + \rho\alpha - \rho\eta_1(u_S - \bar{u}) - \rho\eta_2|\bar{r} - r_S|\right) & \text{if} \quad \Phi_{i,j} = 1,\ \bar{u}_{i,j} > u_{S\,i,j} \end{cases} . \tag{A.6}$$

- $\boxed{\mathrm{prox}_2 = \mathrm{prox}_{\tau A + \tau a(\cdot,v)}(\tilde{u})}$

➪

$$\mathrm{prox}_2 = \arg\min_u \frac{1}{2}(u-\tilde{u})^2 + \tau\eta_1\Phi\max(0, u - u_S) + \tau\eta_1\Phi v\max(0, u_S - u) + \delta_{[u_m,u_M]}(u) \tag{A.7}$$

✦ If $\Phi_{i,j} = 0$, we trivially have:

$$\begin{aligned}
\mathrm{prox}_2 &= \arg\min_u \frac{1}{2}(u-\tilde{u})^2 + \delta_{[u_m,u_M]}(u) \\
&= \mathcal{P}_{[u_m,u_M]}\left(\tilde{u}\right) \ .
\end{aligned} \tag{A.8}$$

✦ If $\Phi_{i,j} = 1$ and $u_{i,j} > u_{S\ i,j}$, we have:

$$\begin{aligned} \mathrm{prox}_2 &= \arg\min_u \frac{1}{2}(u-\tilde{u})^2 + \tau\eta_1(u-u_S) + \delta_{[u_m,u_M]}(u) \\ &= \arg\min_u \frac{1}{2}u^2 - u\tilde{u} + \tau\eta_1 u + K + \delta_{[u_m,u_M]}(u) \\ &= \arg\min_u \frac{1}{2}u^2 - u(\tilde{u}-\tau\eta_1) + K + \delta_{[u_m,u_M]}(u) \\ &= \arg\min_u \frac{1}{2}\left[u-(\tilde{u}-\tau\eta_1)\right]^2 + K' + \delta_{[u_m,u_M]}(u) \\ &= \mathcal{P}_{[u_m,u_M]}\left(\tilde{u}-\tau\eta_1\right)\ . \end{aligned} \tag{A.9}$$

By substituting the optimal value found for u in the splitting condition, we have:

$$u_{i,j} > u_{S\ i,j} \ \Rightarrow\ \tilde{u}_{i,j} > u_{S\ i,j} + \tau\eta_1\ .$$

✦ If $\Phi_{i,j} = 1$ and $u_{i,j} < u_{S\ i,j}$, we have:

$$\begin{aligned} \mathrm{prox}_2 &= \arg\min_u \frac{1}{2}(u-\tilde{u})^2 + \tau\eta_1 v(u_S-u) + \delta_{[u_m,u_M]}(u) \\ &= \arg\min_u \frac{1}{2}u^2 - u\tilde{u} - \tau\eta_1 v u + K + \delta_{[u_m,u_M]}(u) \\ &= \arg\min_u \frac{1}{2}u^2 - u(\tilde{u}+v\tau\eta_1) + K + \delta_{[u_m,u_M]}(u) \\ &= \arg\min_u \frac{1}{2}\left[u-(\tilde{u}+v\tau\eta_1)\right]^2 + K' + \delta_{[u_m,u_M]}(u) \\ &= \mathcal{P}_{[u_m,u_M]}\left(\tilde{u}+v\tau\eta_1\right)\ . \end{aligned} \tag{A.10}$$

By substituting the optimal value found for u in the splitting condition, we have:

$$u_{i,j} < u_{S\ i,j} \ \Rightarrow\ \tilde{u}_{i,j} < u_{S\ i,j} - v\tau\eta_1\ .$$

✦ The remaining case is: $\Phi_{i,j} = 1$ and $u_{i,j} = u_{S\ i,j}$. This directly implies the solution for the proximal operator:

$$\mathrm{prox}_2 = \mathcal{P}_{[u_m,u_M]}\left(u_S\right)\ . \tag{A.11}$$

From the previous cases, we can derive the related validity condition on the calculation point $\tilde{u}_{i,j}$, i.e.:

$$-v\tau\eta_1 < \tilde{u}_{i,j} - u_{S\ i,j} < \tau\eta_1\ .$$

✦ Summing up, we have:

$$\text{prox}_2 = \begin{cases} \mathcal{P}_{[u_m,u_M]}\left(\tilde{u}\right) & \text{if} \quad \Phi_{i,j}=0 \\ \mathcal{P}_{[u_m,u_M]}\left(\tilde{u}-\tau\eta_1\right) & \text{if} \quad \Phi_{i,j}=1,\ \tilde{u}_{i,j} > u_{S\ i,j}+\tau\eta_1 \\ \mathcal{P}_{[u_m,u_M]}\left(\tilde{u}+\upsilon\tau\eta_1\right) & \text{if} \quad \Phi_{i,j}=1,\ \tilde{u}_{i,j} < u_{S\ i,j}-\upsilon\tau\eta_1 \\ \mathcal{P}_{[u_m,u_M]}\left(u_S\right) & \text{otherwise} \end{cases} \tag{A.12}$$

- $\boxed{\text{prox}_3 = \text{prox}_{\tau B+\tau b(\cdot,v)}(\tilde{r})}$

➪

$$\text{prox}_3 = \arg\min_r \frac{1}{2}(r-\tilde{r})^2 + \tau\eta_2\Phi v|r-r_S| + \delta_{[r_m,r_M]}(r) \tag{A.13}$$

✦ If $\Phi_{i,j}=0$, we trivially have:

$$\begin{aligned} \text{prox}_3 \quad &= \arg\min_r \frac{1}{2}(r-\tilde{r})^2 + \delta_{[r_m,r_M]}(r) \\ &= \mathcal{P}_{[r_m,r_M]}\left(\tilde{r}\right)\ . \end{aligned} \tag{A.14}$$

✦ If $\Phi_{i,j}=1$ and $r_{i,j} > r_{S\ i,j}$, we have:

$$\begin{aligned} \text{prox}_3 \quad &= \quad \arg\min_r \frac{1}{2}(r-\tilde{r})^2 + \tau\eta_2 v(r-r_S) + \delta_{[r_m,r_M]}(r) \\ &= \quad \arg\min_r \frac{1}{2}r^2 - r\tilde{r} + \upsilon\tau\eta_2 r + K + \delta_{[r_m,r_M]}(r) \\ &= \quad \arg\min_r \frac{1}{2}r^2 - r(\tilde{r}-\upsilon\tau\eta_2) + K + \delta_{[r_m,r_M]}(r) \\ &= \quad \arg\min_r \frac{1}{2}\left[r-(\tilde{r}-\upsilon\tau\eta_2)\right]^2 + K' + \delta_{[r_m,r_M]}(r) \\ &= \quad \mathcal{P}_{[r_m,r_M]}\left(\tilde{r}-\upsilon\tau\eta_2\right)\ . \end{aligned} \tag{A.15}$$

By substituting the optimal value found for r in the splitting condition, we have:

$$r_{i,j} > r_{S\ i,j}\ \Rightarrow\ \tilde{r}_{i,j} > r_{S\ i,j} + \upsilon\tau\eta_1\ .$$

✦ If $\Phi_{i,j} = 1$ and $r_{i,j} < r_{S\,i,j}$, we have:

$$\begin{aligned}
\text{prox}_3 &= \arg\min_r \frac{1}{2}(r-\tilde{r})^2 + \tau\eta_2 \upsilon(r_S - r) + \delta_{[r_m,r_M]}(r) \\
&= \arg\min_r \frac{1}{2}r^2 - r\tilde{r} - \upsilon\tau\eta_2 r + K + \delta_{[r_m,r_M]}(r) \\
&= \arg\min_r \frac{1}{2}r^2 - r(\tilde{r} + \upsilon\tau\eta_2) + K + \delta_{[r_m,r_M]}(r) \\
&= \arg\min_r \frac{1}{2}\left[r - (\tilde{r} + \upsilon\tau\eta_2)\right]^2 + K' + \delta_{[r_m,r_M]}(r) \\
&= \mathcal{P}_{[r_m,r_M]}\left(\tilde{r} + \upsilon\tau\eta_2\right) \ .
\end{aligned} \tag{A.16}$$

By substituting the optimal value found for r in the splitting condition, we have:

$$r_{i,j} < r_{S\,i,j} \ \Rightarrow \ \tilde{r}_{i,j} < r_{S\,i,j} - \upsilon\tau\eta_1 \ .$$

✦ The remaining case is: $\Phi_{i,j} = 1$ and $r_{i,j} = r_{S\,i,j}$. This directly implies the solution for the proximal operator:

$$\text{prox}_3 = \mathcal{P}_{[r_m,r_M]}\left(r_S\right) \ . \tag{A.17}$$

From the previous cases, we can derive the related validity condition on the calculation point $\tilde{r}_{i,j}$, i.e.:

$$\left|\tilde{r}_{i,j} - r_{S\,i,j}\right| < \upsilon\tau\eta_2 \ .$$

✦ Summing up, we have:

$$\text{prox}_3 = \begin{cases}
\mathcal{P}_{[r_m,r_M]}\left(\tilde{r}\right) & \text{if} \quad \Phi_{i,j} = 0 \\
\mathcal{P}_{[r_m,r_M]}\left(\tilde{r} - \upsilon\tau\eta_2\right) & \text{if} \quad \Phi_{i,j} = 1, \ \tilde{r}_{i,j} > r_{S\,i,j} + \upsilon\tau\eta_2 \\
\mathcal{P}_{[r_m,r_M]}\left(\tilde{r} + \upsilon\tau\eta_2\right) & \text{if} \quad \Phi_{i,j} = 1, \ \tilde{r}_{i,j} < r_{S\,i,j} - \upsilon\tau\eta_2 \\
\mathcal{P}_{[r_m,r_M]}\left(r_S\right) & \text{otherwise}
\end{cases} \tag{A.18}$$

References

Bevilacqua, M., Aujol, J.-F., Brédif, M., Bugeau, A., 2016. Visibility Estimation and Joint Inpainting of Lidar Depth Maps. In: IEEE International Conference on Image Processing (ICIP). pp. 1–5.

Brédif, M., 2013. Image-Based Rendering of LOD1 3D City Models for traffic-augmented Immersive Street-view Navigation. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences 1 (3), 7–11.

Chambolle, A., Pock, T., 2011. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. Journal of Mathematical Imaging and Vision 40 (1), 120–145.

Chan, D., Buisman, H., Theobalt, C., Thrun, S., 2008. A Noise-Aware Filter for Real-Time Depth Upsampling. In: ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications (M2SFA2). pp. 1–12.

Chan, T. F., Esedoglu, S., 2005. Aspects of Total Variation Regularized L1 Function Approximation. SIAM Journal on Applied Mathematics 65 (5), 1817–1837.

Chan, T. F., Golub, G. H., Mulet, P., 1999. A Nonlinear Primal-Dual Method for Total Variation-Based Image Restoration. SIAM Journal on Scientific Computing 20 (6), 1964–1977.

Chen, W.-Y., Chang, Y.-L., Lin, S.-F., Ding, L.-F., Chen, L.-G., 2005. Efficient Depth Image Based Rendering with Edge Dependent Depth Filter and Interpolation. In: IEEE International Conference on Multimedia and Expo (ICME). pp. 1314–1317.

Diebel, J., Thrun, S., 2005. An application of Mmarkov random fields to range sensing. In: Advances in Neural Information Processing Systems (NIPS). Vol. 5. pp. 291–298.

Ferstl, D., Reinbacher, C., Ranftl, R., Rüther, M., Bischof, H., 2013. Image Guided Depth Usampling using Anisotropic Total Generalized Variation. In: IEEE International Conference on Computer Vision (ICCV). pp. 993–1000.

Garcia, F., Mirbach, B., Ottersten, B., Grandidier, F., Cuesta, A., 2010. Pixel weighted average strategy for depth sensor data fusion. In: 17th IEEE International Conference on Image Processing (ICIP). IEEE, pp. 2805–2808.

Greene, N., Kass, M., Miller, G., 1993. Hierarchical Z-buffer visibility. In: 20th Internation Conference on Computer Graphics and Interactive Techniques (SIGGRAPH). ACM, pp. 231–238.

Harrison, A., Newman, P., 2010. Image and Sparse Laser Fusion for Dense Scene Reconstruction. In: Field and Service Robotics (FRS). Springer, pp. 219–228.

Herbort, S., Wöhler, C., 2011. An introduction to image-based 3D surface reconstruction and a survey of photometric stereo methods. 3D Research 2 (3), 1–17.

Huhle, B., Schairer, T., Jenke, P., Straßer, W., Dec. 2010. Fusion of range and color images for denoising and resolution enhancement with a non-local filter. Computer Vision and Image Understanding 114 (12), 1336–1345.

Katz, S., Tal, A., Basri, R., Jul. 2007. Direct Visibility of Point Sets. ACM Transactions on Graphics (TOG) 26 (3), 24.

Kolb, A., Barth, E., Koch, R., Larsen, R., 2010. Time-of-Flight Cameras in Computer Graphics. In: Computer Graphics Forum. Vol. 29. Wiley Online Library, pp. 141–159.

Liu, J., Gong, X., 2013. Guided Depth Enhancement via Anisotropic Diffusion. In: Advances in Multimedia Information Processing – PCM 2013. Springer International Publishing, pp. 408–417.

Nikolova, M., 2004. A Variational Approach to Remove Outliers and Impulse Noise. Journal of Mathematical Imaging and Vision 20 (1-2), 99–120.

Paparoditis, N., Papelard, J.-P., Cannelle, B., Devaux, A., Soheilian, B., David, N., Houzay, E., 2012. Stereopolis II: A multi-purpose and multi-sensor 3D mobile mapping system for street visualisation and 3D metrology. Revue Française de Photogrammétrie et de Télédétection 1 (200), 69–79.

Parikh, N., Boyd, S., 2013. Proximal Algorithms. Foundations and Trends in Optimization 1 (3), 123–231.

Park, J., Kim, H., Tai, Y.-W., Brown, M. S., Kweon, I., 2011. High Quality Depth Map Upsampling for 3D-TOF Cameras. In: IEEE International Conference on Computer Vision (ICCV). IEEE, pp. 1623–1630.

Pierre, F., Aujol, J.-F., Bugeau, A., Papadakis, N., Ta, V.-T., 2015. Luminance-Chrominance Model for Image Colorization. SIAM Journal on Imaging Sciences (SIIMS) 8 (1), 536–563.

Schmeing, M., Jiang, X., 2011. Depth Image Based Rendering. In: Pattern Recognition, Machine Intelligence and Biometrics. Springer, pp. 279–310.

Schwarz, S., Sjöström, M., Olsson, R., 2012. Depth Map Upscaling Through Edge Weighted Optimization. In: Three-Dimensional Image Processing (3DIP) and Applications II. Vol. 8290. Society of Photo-Optical Instrumentation Engineers (SPIE).

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R., Jun. 2006. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). Vol. 1. pp. 519–528.

Stoykova, E., Ayd, A., Benzie, P., Grammalidis, N., Malassiotis, S., Ostermann, J., Piekh, S., Sainov, V., Theobalt, C., Thevar, T., et al., 2007. 3-D time-varying scene capture technologies–A survey. IEEE Transactions on Circuits and Systems for Video Technology 17 (11), 1568–1586.

Yang, Q., Ahuja, N., Yang, R., Tan, K.-H., Davis, J., Culbertson, B., Apostolopoulos, J., Wang, G., 2013. Fusion of median and bilateral filtering for range image upsampling. IEEE Transactions on Image Processing 22 (12), 4841–4852.

Zhang, Z., 2012. Microsoft Kinect sensor and its effect. IEEE MultiMedia 19 (2), 4–10.

Zinger, S., Do, L., de With, P., 2010. Free-viewpoint depth image based rendering. Journal of Visual Communication and Image Representation 21 (5), 533–541.