Spatially Regularized Fusion of Multi-Resolution Digital Surface Models

Georg Kuschk, Pablo d'Angelo, David Gaudrie, Peter Reinartz (Member IEEE), Daniel Cremers

Abstract

In this work we propose an algorithm for robustly fusing digital surface models (DSM) with different ground sampling distances and confidences, using explicit surface priors to obtain locally smooth surface models. Robust fusion of the DSMs is achieved by minimizing the L1-distance of each pixel of the solution to each input DSM. This approach is similar to a pixel-wise median and most outliers are discarded. We further incorporate local planarity assumption as an additional constraint to the optimization problem, thus reducing the noise compared to pixel-wise approaches. The optimization is also inherently able to include weights for the input data, therefore allowing to easily integrate invalid areas, fuse multi-resolution DSMs and to weight the input data. The complete optimization problem is constructed as a variational optimization problem with a convex energy functional, such that the solution is guaranteed to converge towards the global energy minimum. An efficient solver is presented to solve the optimization in reasonable time, e.g. running in real-time on standard computer vision camera images. The accuracy of the algorithms and the quality of the resulting fused surface models is evaluated using synthetic datasets and spaceborne datasets from different optical satellite sensors.

Index Terms

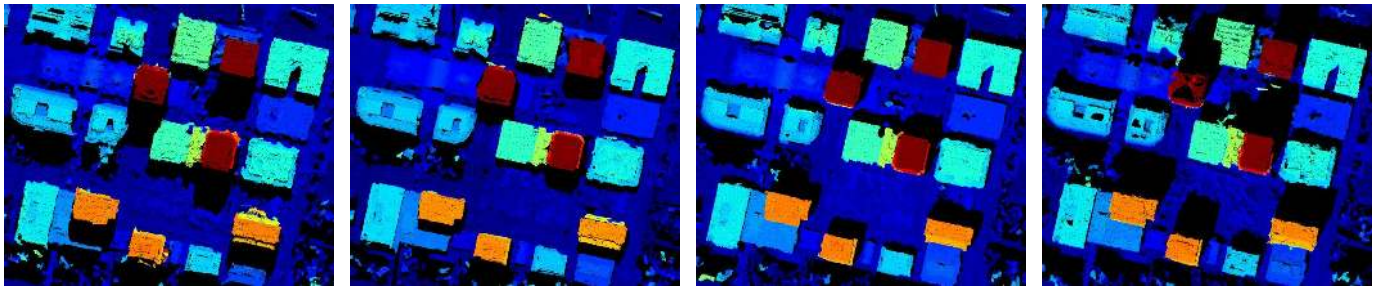
DSM, 3D Reconstruction, Data Fusion, Variational Methods

I. Introduction

With an ever increasing amount of earth observation sensors, the problem of having data at all, increasingly shifts towards the problem of how to make best use of an abundance of data. One aspect of remote sensing data is the 3D information contained in the observed images, resulting in digital surface models (DSM), constituting a basic component for many applications, such as orthophoto creation, mapping, visualization and 3D planning. As many technologies for DSM generation exist (airborne LiDAR, SAR interferometry, automatic image matching, ..) the corresponding results differ in their characteristics and quality in general. Because of the decreasing revisit time for many parts of the Earth's landmass, multiple datasets of DSMs are available for these regions and it is therefore interesting to fuse these into a single DSM with higher accuracy. Depending on the underlying satellite characteristics like ground sampling distance (GSD), the DSMs capture different parts of the scene in different quality, which even can be mutually exclusive to some extent. For example, high resolution sensors like WorldView-2 with a GSD of 0.5m perform very well in urban areas, whereas the results in forest areas are somewhat moderate. In contrast, Cartosat-1 with a GSD of 2.5m performs quite opposite in these areas [1]. Even with the same sensor, a different exposure time can drastically alter the results in shadow areas or in highly reflective areas like glaciers. Clouds are posing an additional problem for optical image sensing, providing no valid data in these areas, thereby requiring these gaps to be filled in by valid data from other sensors or another timestamp. A prominent example for a large data abundance is aerial imaging, which typically produces large image streams with image overlaps $>80\%$. For computing the corresponding 3D reconstruction, many multi-view image matching techniques match stereo image pairs individually and later fuse the resulting DSMs into a common height model, see e.g. [2], [3], [4].

Our work focuses on the fusion of 2.5D DSM grids, with a resolution from several decimeters to a few meters. We use the common notation of 2.5D to explicitly distinguish between 3D point cloud registration / fusion and fusing their projections in a common 2D reference frame. The latter consists of 2D images, each pixel containing its height above ground and is commonly referred as 2.5D DSM, as it contains 3D height information but not to full extent (e.g. no bridges can be modelled). DSM fusion has been considered by various authors previously. The simplest method is based on weighted averaging of two or more height maps [5], [6]. As weighted averaging cannot deal with outliers or blunders in the DSMs, a median fusion is often used for multi-DSM fusion, sometimes followed by weighted averaging of the inliers [2]. Both median and weighted averaging process each pixel independently, and thus cannot take into account the local surface shape, which is regular for many areas. Applying additional mean or median based filtering spatially reduces the amount of noise to some extent, at the cost of blurring potentially sharp edges. An example for context aware fusion algorithms is the use of sparse representations [7], where a DSM patch is computed as a sparse linear combination of dictionary DSM patches. Except for median fusion, pixel-wise error maps are required by weighted averaging and sparse representations. A comparison between weighted averaging and sparse representations [8] found that the quality of the fused DSMs is mostly determined by the quality of these pixel based error maps.

G. Kuschk and D. Cremers are with the Technical University of Munich (TUM), Germany, email: [georg.kuschk , cremers]@tum.de; P. d'Angelo, D. Gaudrie and P. Reinartz are with the Remote Sensing Technology Institute, German Aerospace Center (DLR), Germany, email: [pablo.angelo, david.gaudrie, peter.reinartz]@dlr.de



(a) (b) (c) (d)

Fig. 1. (a)-(d): Four co-registered DSMs, obtained from optical stereo reconstruction using [9] for different camera view points / satellite positions (noticeable by the different invalid occlusion areas in black).

Another direction of work aims at formulating a global energy functional, minimizing the distance of the fused result to all input DSMs simultaneously and additionally incorporating the assumption of the world being locally planar ([10], [11], [12], [13]). Due to its simple structure and theoretically well founded minimization procedure, we build upon this work and extend it to a weighted, multi-resolution, fusion framework.

II. Method

As basic fusion algorithms we are looking at the following pixel-wise fusion methods: mean and median fusion, as well as *medmean fusion*. We define the latter one as median based fusion that reduces the amount of outliers in the fused DSM by averaging the median value for each pixel with all other DSM heights of this pixel being at a distance of less than 2m from the median value. Note that this is an empirical threshold, depending on the overall height range and noise level. In contrast to these simple pixel-wise fusion methods, advanced methods usually enforce some kind of spatial smoothness constraint to get closer to a physical meaningful solution, with neighboring image pixels forced to have a similar height value. Note that this constraint often is in contrast to the data term (height values) of the involved images, where neighboring pixels can differ significantly in height. This leads to the general formulation of our DSM fusion problem as

$$\min_{\mathbf{u}} \left\{ R(\mathbf{u}) + \lambda_d \sum_{k=1}^{K} \|\mathbf{u} - \mathbf{g_k}\|_1 \right\} \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^{M \cdot N}$ is the *'optimal'* DSM to solve for, already written as stacked vector of pixels to simplify notation in the following. The K (noisy) input DSMs are given as $\mathbf{g_k}$ (see e.g. Figure 1), the scalar factor λ_d is balancing the impact of the smoothness term and the data term and $R(u)$ is depicting a general regularizer on u.
In the case of DSM fusion, these smoothness constraints (or regularizers) are the assumption of the world being locally planar, meaning that the height value of each pixel of the DSM depends on its local context and e.g. is highly unlikely to have a significantly different height value than its surrounding pixels.
This smoothness constraint typically is implemented by minimizing the sum of gradients of the resulting DSM in both x and y-direction, resulting in large partial differential equation systems (PDE).
In recent years, Total Variation based methods (TV) for minimizing energy functionals have seen a lot of attention in the research community. One reason is that these algorithms are very well-suited for parallelization and, together with the recent advances of GPU-based computational power, lead to efficient algorithms, solving these optimization problems efficiently. And as the energy functional of our image fusion problem is written in a convex formulation, the solution is globally optimal and independent of its initialization. Since the second term of Equation 1 is always convex in the variable $\mathbf{u}$ to solve for (sum of norms), the complete energy functional is convex, if the regularizer $R(\mathbf{u})$ is convex. The two regularizers used in this paper are described in Section II-A (namely TV and TGV) and are simply linear transformations of the type $K \cdot \mathbf{u}$. Therefore throughout this paper Equation 1 will always be convex.

A. TV-L_1 Fusion

Based upon the Rudin-Osher-Fatemi image denoising model (ROF-model) [14], the extension for multiple image fusion, together with replacing the quadratic data term by the more robust L_1 norm as in [12] is written as

$$\min_{\mathbf{u}} \left\{ \|\nabla \mathbf{u}\|_1 + \lambda_d \sum_{k=1}^{K} \|\mathbf{u} - \mathbf{g_k}\|_1 \right\} \tag{2}$$

Note that the choice of the L_1 norm for both the gradient and the data term plays an important role for the fusion of multiple noisy DSMs (or images in general) for the following reasons: Applied to the regularizer (gradient) it still enables the solution

to exhibit strong edges / discontinuities (e.g. at the transition of house roof tops to street level), as these height value jumps are only penalized linearly. Applying the L_1 norm to the second term - the data term - makes the whole fusion process robust to outliers as well, as these also are only weighted linearly in the optimization process and their influence therefore is limited compared to *e.g.* a least squares minimization approach. While this model already provides good results by smoothing flat areas and preserving sharp discontinuities, it suffers from the so-called staircasing effect. This effect is a direct result of the regularizer, whose assumption is a locally planar world - where planar unfortunately refers to locally fronto-parallel. This staircasing effect of the TV-L_1 algorithm is visible in Figure 2(f), resulting in a slanted roof which is not smooth. One way to overcome this issue is using the Huber norm instead of the pure L_1 norm for the regularizer, thereby penalizing small height differences quadratically and larger difference as before using the L_1 norm. This results in a locally more smooth surface, mitigating the staircasing effect to some extent. The authors of [12] added this Huber regularized fusion method as one baseline method to compare their algorithms against. However, this does not solve the issue of reconstructing large non-fronto-parallel surfaces (slanted planes). To achieve that goal, a more advanced smoothness assumption as in the following section is required. For further details about the results of TV-Huber-based regularization, we refer to the work of [12].

B. TGV-L_1 Fusion

To overcome the fronto-parallel assumption of TV-L_1 minimization, [15] introduced the mathematical model of Total Generalized Variation (TGV) has been introduced as a higher-order extension of Total Variation which favors the solution to consist of piecewise polynomial functions (e.g. fronto-parallel, affine, quadratic). Especially the 2nd order is of high interest, as it forces the solution to consist of piecewise planar functions, which means that compared to the fronto-parallel TV-L_1 model, the regularizer now also favors slanted planes. [12] applied this model to DSM fusion, resulting in the following optimization problem

$$\min_{\mathbf{u},\mathbf{v}} \left\{ \lambda_s \|\nabla_u \mathbf{u} - \mathbf{v}\|_1 + \lambda_a \|\nabla_v \mathbf{v}\|_1 + \lambda_d \sum_{k=1}^{K} \|\mathbf{u} - \mathbf{g_k}\|_1 \right\} \tag{3}$$

Now, before the variation of the image $\mathbf{u}$ is measured, a 2D vector field $\mathbf{v}$ is subtracted from the gradient of $\mathbf{u}$. An affine surface in the image $\mathbf{u}$ has a constant gradient $\nabla\mathbf{u}$, so by coupling and minimizing $|\nabla\mathbf{u} - \mathbf{v}|$, the vector field $\mathbf{v}$ will also be constant and it's gradient $\nabla\mathbf{v}$ therefore zero. Regarding our overall optimization problem, this means that the energy term will be lower, if affine functions can be found in the image, whereas non-affine functions get additional penalties by $|\nabla\mathbf{v}|$. The values $\lambda_s, \lambda_a, \lambda_d$ are scalar weights and balance the impact of the smoothness term, the affine term and the data term. Note that we now notationally need to differ between two gradient operators, $\nabla_u \in \mathbb{R}^{MN\times 2MN}$ and $\nabla_v \in \mathbb{R}^{2MN\times 2MN}$ as the corresponding vector spaces are of different dimension (see Section III-A).

C. Weighted TGV-L_1 Fusion

When fusing DSMs it is desirable to weight the input DSMs on a per pixel base, to be able to incorporate additional prior knowledge into the fusion process. This prior knowledge for example can be based on the different sensor characteristics used to generate the DSM, confidence measures during the 3D reconstruction process itself, information about occluded and therefore unknown areas in each DSM, etc. We therefore extend Equation 3 with a weighting matrix W_k for each input DSM

$$\min_{\mathbf{u},\mathbf{v}} \left\{ \lambda_s \|\nabla\mathbf{u} - \mathbf{v}\|_1 + \lambda_a \|\nabla\mathbf{v}\|_1 + \lambda_d \sum_{k=1}^{K} W_k \|\mathbf{u} - \mathbf{g_k}\|_1 \right\} \tag{4}$$

D. Parameters

This optimization problem (and the ones in Equation 2 and 3) is very parameter dependent, as we need to adapt the influence of the data term λ_d manually for datasets with different ranges of $g_k^{(i,j)} \in \mathbf{g_k}$ as well as for a different number K of input images. To achieve independence of the data range of the input DSMs, we scale all input data to the interval $[0..1]$

$$g_k^{(i,j)} = \frac{g_k^{(i,j)} - g_{min}}{g_{max} - g_{min}} \tag{5}$$

with $g_{min} = \min_{i,j,k} g_k^{(i,j)}$ and $g_{max} = \max_{i,j,k} g_k^{(i,j)}$. The independence from K is achieved by normalizing the influence of the data term w.r.t. the two-image case and using the adaptive

$$\lambda_d^K = \frac{2}{K}\lambda_d \tag{6}$$

Note that we do not need all 3 weighting factors $\lambda_s, \lambda_a, \lambda_d$, as we can multiply the whole energy functional with $\frac{1}{\lambda_d}$. We therefore only have to deal with λ_s, λ_a and $\lambda_d = 1$ implicitly. Additionally it is a good choice to set $\lambda_a = 4\lambda_s$, which leaves

us with only one parameter λ_s to choose between a large smoothing impact ($\lambda_s >>$) or a more data-driven fusion ($\lambda_s <<$). Choosing λ_a too big results in oversmoothing of discontinuities – we loose some of our edge-preserving capability. When choosing λ_a very small, we obtain results closer to pure TV-L_1 (together with the staircasing effects). To avoid an additional free parameter, we coupled the value to the smoothness weighting λ_s and experimented with different correlation factors. In all our empirical tests over different artificial and natural datasets the choice $\lambda_a = 4\lambda_s$ produced consistently good results.

All these extensions and modifications apply to the TV-L_1 method similarly. In the next section we will go into detail about how to solve these optimization problems numerically.

III. Optimization

In the following we describe the numerical optimization of our weighted TGV-L_1 energy functional given in Equation 4. The solution for the TV-L_1 energy functional is similar and can be derived easily from the solution below.

To solve for the fused DSM $\mathbf{u} \in \mathbb{R}^{M\times N}$ (in the following written as stacked vector $\mathbb{R}^{MN\times 1}$) in Equation 4, we need to overcome the non-differentiable L_1-norm, which complicates any gradient descent based minimization scheme. An efficient algorithm which elegantly circumvents the differentiability problem of the gradient operator is the primal-dual algorithm of [16]. By applying the Legendre-Fenchel transform to the terms involving the derivative of the primal variables we obtain the dual formulation / conjugate of these terms as

$$\begin{aligned} \lambda_s\|\nabla\mathbf{u}-\mathbf{v}\|_1 &= \max_{\mathbf{p}\in P}\{\langle\nabla\mathbf{u}-\mathbf{v},\mathbf{p}\rangle\} \\ \lambda_a\|\nabla\mathbf{v}\|_1 &= \max_{\mathbf{q}\in Q}\{\langle\nabla\mathbf{v},\mathbf{q}\rangle\} \end{aligned} \tag{7}$$

such that the transformed saddle-point problem of Equation 4 in the primal variables $\mathbf{u},\mathbf{v}$ and the dual variables $\mathbf{p},\mathbf{q}$ with constraints

$$\begin{aligned} P &= \{\mathbf{p}\in\mathbb{R}^{2MN} : \|\mathbf{p}\|_\infty \le \lambda_s\} \\ Q &= \{\mathbf{q}\in\mathbb{R}^{4MN} : \|\mathbf{q}\|_\infty \le \lambda_a\} \end{aligned} \tag{8}$$

is

$$\min_{\mathbf{u},\mathbf{v}}\max_{\mathbf{p},\mathbf{q}}\left\{\langle\nabla\mathbf{u}-\mathbf{v},\mathbf{p}\rangle + \langle\nabla\mathbf{v},\mathbf{q}\rangle + \lambda_d\sum_{k=1}^{K} W_k\|\mathbf{u}-\mathbf{g_k}\|_1\right\} \tag{9}$$

A detailed explanation of the dual variables and the corresponding vector spaces is given in Section III-A. With the convex saddle-point problem above (Equation 9), we can now directly apply the primal-dual algorithm of [16] to get the following optimization scheme, which is basically iteratively performing gradient descents on the primal variables and gradient ascents on the dual variables:

Input: $\mathbf{u}^0,\mathbf{v}^0,\mathbf{p}^0,\mathbf{q}^0=\mathbf{0}$, $\bar{\mathbf{u}}^0=\mathbf{u}^0,\bar{\mathbf{v}}^0=\mathbf{v}^0$, $\theta=1$, step sizes $\tau_i>0$
Iterations $n\ge 0$:

$$\begin{cases} \mathbf{p}^{n+1} = \Pi_P\left(\mathbf{p}^n + \tau_p\lambda_s(\nabla\bar{\mathbf{u}}^n - \bar{\mathbf{v}}^n)\right) \\ \mathbf{q}^{n+1} = \Pi_Q\left(\mathbf{q}^n + \tau_q\lambda_a(\nabla\bar{\mathbf{v}}^n)\right) \\ \mathbf{u}^{n+1} = \text{prox}_f\left(\mathbf{u}^n + \tau_u\lambda_s\nabla^*\mathbf{p}^{n+1}\right) \\ \mathbf{v}^{n+1} = \mathbf{v}^n + \tau_v(\lambda_a\nabla^*\mathbf{q}^{n+1} + \lambda_s\mathbf{p}^{n+1}) \\ \bar{\mathbf{u}}^{n+1} = \mathbf{u}^{n+1} + \theta(\mathbf{u}^{n+1}-\mathbf{u}^n) \\ \bar{\mathbf{v}}^{n+1} = \mathbf{v}^{n+1} + \theta(\mathbf{v}^{n+1}-\mathbf{v}^n) \end{cases}$$

Listing 1. Primal-dual optimization algorithm for TGV-L1-based image fusion

For details about the linear operators ∇ and their negative adoints ∇^*, as well as the step sizes τ_i for the gradient descents see Section III-A. To ensure the constraints of Equation 8, the corresponding proximal mappings of the dual variables are given as simple point-wise projections

$$\begin{aligned} \Pi_P(\mathbf{p}) &= \frac{\mathbf{p}}{\max\{1,\|\mathbf{p}\|/\lambda_s\}} \\ \Pi_Q(\mathbf{q}) &= \frac{\mathbf{q}}{\max\{1,\|\mathbf{q}\|/\lambda_a\}} \end{aligned} \tag{10}$$

The proximal mapping of the primal variable u, enforcing the data constraints $\min\sum_k\|u-g_k\|$ is slightly more complicated. In previous work, [12] and [17] added Lagrange multipliers for each observation ($\langle r_k, u-g_k\rangle$) and optimized the energy functional with an additional gradient descent scheme for these auxiliary variables. Here we build upon the work of [18] to solve this constraint exactly and directly, thus avoiding an additional iterative scheme. We therefore don't need further dual variables for every observation as in [12], resulting in less memory consumption. As the closed-form solution of the

proximal mapping is computationally simple, it further results in a noticeable speedup compared to solving it via an iterative gradient-descent based primal-dual scheme. Defining

$$f(x) = \lambda\tau \sum_{k=1}^{K} w(x,k) \cdot \|x - g_k\|_1 \tag{11}$$

the proximal mapping is given as

$$\begin{aligned} &\text{prox}_f(x) = \\ &\arg\min_y \left\{ \frac{1}{2}\|x-y\|_2^2 + \lambda\tau \sum_{k=1}^{K} w(y,k) \cdot \|y - g_k\|_1 \right\} \end{aligned} \tag{12}$$

whose solution is given by a generalized shrinkage formula according to [18]:

$$\text{prox}_f(x) = \text{median}\{g_1, ..., g_K, p_0, p_1, ..., p_K\} \tag{13}$$

with

$$p_i = x + \tau\lambda W_i \tag{14}$$

$$W_i = -\sum_{j=1}^{i} w(x,j) \; + \; \sum_{j=i+1}^{K} w(x,j) \tag{15}$$

A. Implementation Details

For discretization of the gradient operators $\nabla_u : \mathbb{R} \to \mathbb{R}^2$ and $\nabla_v : \mathbb{R}^2 \to \mathbb{R}^4$, we use forward finite differences with Neumann boundary conditions

$$\nabla_u = \begin{pmatrix} \nabla_x \\ \nabla_y \end{pmatrix}, \quad \nabla_v = \begin{pmatrix} \nabla_x & 0 \\ \nabla_y & 0 \\ 0 & \nabla_x \\ 0 & \nabla_y \end{pmatrix}, \quad \nabla_x, \nabla_y \in \mathbb{R}^{MN \times MN} \tag{16}$$

where

$$\begin{aligned} (\nabla_x \mathbf{u})_{\gamma(i,j)} &= \begin{cases} \mathbf{u}_{\gamma(i+1,j)} - \mathbf{u}_{\gamma(i,j)} & \text{if } i < M \\ 0 & \text{if } i = M \end{cases} \\ (\nabla_y \mathbf{u})_{\gamma(i,j)} &= \begin{cases} \mathbf{u}_{\gamma(i,j+1)} - \mathbf{u}_{\gamma(i,j)} & \text{if } j < N \\ 0 & \text{if } j = N \end{cases} \end{aligned} \tag{17}$$

are the forward finite differences in x and $y-$direction and the function $\gamma : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ mapping the indices from 2D image space to 1D stacked vector notation

$$\gamma(i,j) = (i-1)M + j \tag{18}$$

The corresponding negative adoint operators ∇^*, needed for the gradient descent in the dual variables of Algorithm 1, are simply the corresponding transposed and negated matrices $\nabla^* = -\nabla^T$. Note that these are sometimes in literature also referred to as divergence operators. When written explicitly, the above definition naturally reads as backward finite differences with Dirichlet boundary conditions

$$\nabla_u^* \mathbf{p} = -\begin{pmatrix} \nabla_x & \nabla_y \end{pmatrix} \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \end{pmatrix}, \qquad \begin{aligned} (\nabla_u^* \mathbf{p})_{i,j} = &\begin{cases} \mathbf{p}^1_{i,j} - \mathbf{p}^1_{i-1,j} & \text{if } 1 < i < N \\ \mathbf{p}^1_{i,j} & \text{if } i = 1 \\ -\mathbf{p}^1_{i-1,j} & \text{if } i = N \end{cases} \\ &+ \begin{cases} \mathbf{p}^2_{i,j} - \mathbf{p}^2_{i,j-1} & \text{if } 1 < j < M \\ \mathbf{p}^2_{i,j} & \text{if } j = 1 \\ -\mathbf{p}^2_{i,j-1} & \text{if } j = M \end{cases} \end{aligned} \tag{19}$$

The implementation is similar for the second operator ∇_v and its negative adjoint. Although the mathematical notation may imply a very large optimization problem (e.g. $\nabla_x \in \mathbb{R}^{MN \times MN}$), the corresponding matrices are very sparse: ∇_u, ∇_v only have two non-zero elements per matrix row. Therefore implementation can be done efficiently either using a sparse matrix representation, or avoiding this overhead by directly computing the gradient and divergence per pixel using Equations 17 and 19.

To ensure convergence of the primal-dual algorithm, the step sizes of the gradient ascents/descents are bound to the operator norm of the linear operators described in Equation 16 according to [16] as follows

$$\tau_u \tau_p \leq \frac{1}{||\nabla_u||^2_{op}} \qquad \text{and} \qquad \tau_v \tau_q \leq \frac{1}{||\nabla_v||^2_{op}} \quad . \tag{20}$$

Due to the simple structure of the forward differences the step sizes can be explicitly computed as $\tau_u = \tau_p = \tau_v = \tau_q = 1/\sqrt{8}$. The whole algorithm stops, if either a predefined maximum number of iterations has been reached or the energy change between successive iterations drops below a relative threshold. Due to the stacked vector notation, the input weights are denoted as diagonal matrices W_k and the corresponding multiplications are actually a pixel-wise multiplication.
Since the algorithm is inherently suited for parallelization, the algorithm was implemented on GPU, allowing for a processing speed of 40ms for 10 images with a size of 640×480 (using a Nvidia GTX 970). Since GPU memory cannot be easily swapped to the harddrive and the DSMs to fuse are usually quite large (near Gigapixel range for satellite data), we process larger data by tiling it into overlapping smaller regions, solving these separately. The overlap is chosen as 5% of the corresponding width of the tiles, means that for *e.g.* quadratic tiles of 1000 pixel width, the overlap w.r.t. to the neighboring tile amounts to 50 pixel. To further account for the less accurate results at the tile borders, we employ the same strategy as used by [2]. Instead of just computing the mean value of neighboring tiles in the overlapping area, a weighted mean is used, such that the corresponding weights decrease linearly towards the tile border. Of course, when handling such large DSMs and processing them in tiles the overall solution is not globally optimal anymore. The tiling size is computed as large as possible while the complete data still fits into GPU memory. With the memory overhead of TGV-L_1 based optimization and *e.g.* 5 input DSMs, this amounts to tiles of roughly 8000×8000 pixel for a current GPU having 8GB of memory.

IV. Evaluation

A. Artificial Tests

The first evaluation is done on synthetic data. A given ground truth DSM $\mathbf{g}$ with a height range of $[0..170]$ is perturbed with Gaussian noise and with salt and pepper noise to simulate different noisy observations of the scene. Five of these noisy DSMs are then given as input to the fusion algorithms and the accuracy of the output DSM $\mathbf{u}$ is measured by the logarithmic signal-to-noise ratio:

$$SNR = 10 \log_{10} \left(\frac{I^2_{\text{signal}}}{I^2_{\text{noise}}} \right) = 10 \log_{10} \left(\frac{||\mathbf{g}||^2}{||\mathbf{u} - \mathbf{g}||^2} \right) \tag{21}$$

In Figure 2, visual and numerical results are given, showing a significantly higher accuracy of the global optimization methods for DSM fusion over simple mean and median based fusion. We can also remark the staircasing effects provided by TV-L_1 fusion resulting in a non-smooth roof in Figure 2 (f), as well as the smoothness of TGV-L_1 fusion, which has both the best SNR and the best visual aspect. To obtain a fair comparison between TV-L_1 and TGV-L_1 based fusion, we ran the algorithms for varying λ_d values and chose the parameter which resulted in the highest SNR value – compare Figure 3. Furthermore the noise was fixed for the different runs as well.

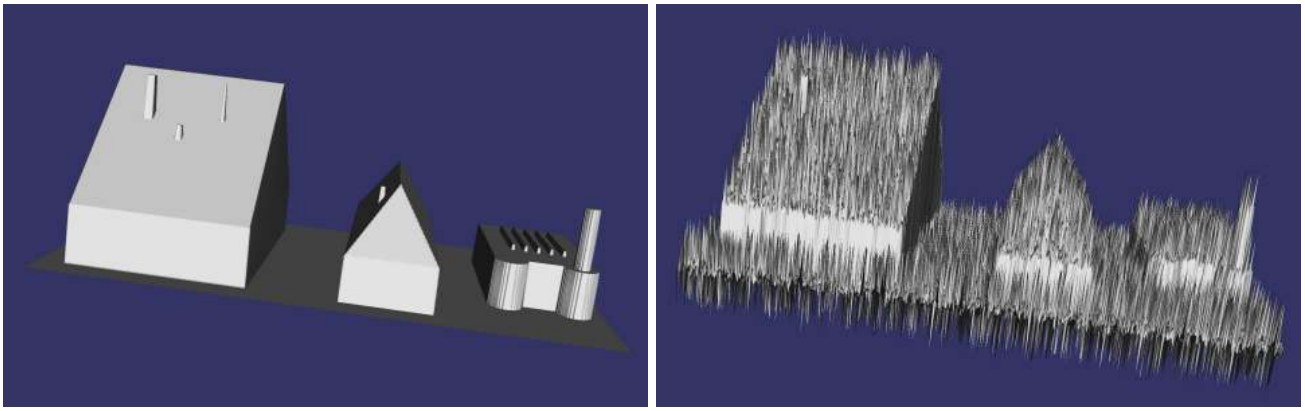
B. Artificial Tests - Weights

In this experiment, we compare the basic fusion of Equation 2 and 3 against the formulation using an explicit weighting scheme as proposed in Equation 4. To this end, we add a wrong systematic bias to 3 of our 5 input images (compare Figure 4 (c) and set corresponding weights $w = 0.2$ for these areas, whereas the rest is set to $w = 1.0$. Note that we deliberately did not set the weights for the wrong areas to zero, to simulate some uncertainty about our knowledge of these areas. As can be seen in Figure 4 (e) and (h), the absence of an explicit weighting results in fused DSMs with a remaining systematic error in the two modified areas, as 3 out of 5 images exhibit the same systematic offset, although with different noise. When incorporating additional prior information (here: down-weighting the image areas with the wrong offset), the optimization process is able to reconstruct the intended surface, compare Figure 4 (f) and (i). To obtain a fair comparison of the 4 different energy functionals, we ran the algorithm for varying λ_d values and chose the parameter which resulted in the highest SNR value – compare Figure 4 (g) and (j). Furthermore the noise was fixed for the different runs as well.

C. Artificial Tests - Varying DSM resolution / Sparse DSM

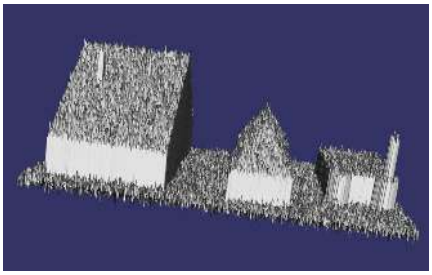
In this experiment, we compare the fusion results of the following two cases

- One noisy input DSM is given. This reduces the algorithm to a pure denoising algorithm.
- Additionally to the noisy DSM given before, an additional accurate DSM is given, exhibiting strong sparsity. This can be the result of either projecting a coarse-resolution DSM to the coordinate frame of a fine-resolution DSM or general depth priors resulting from completely different sensors as for example radar satellites.

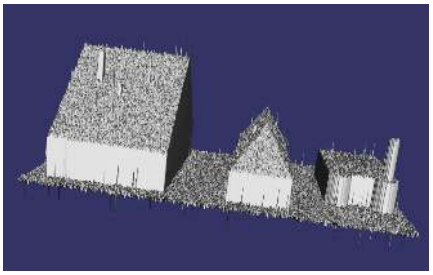


(a) Groundtruth data

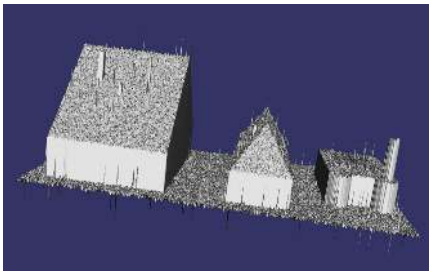
(b) 1 of 5 noisy input images



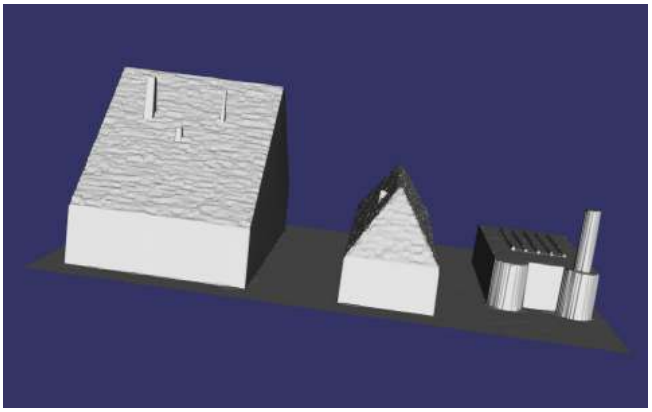
(c) Fusion using pixel based mean filtering, $SNR = 22.55$



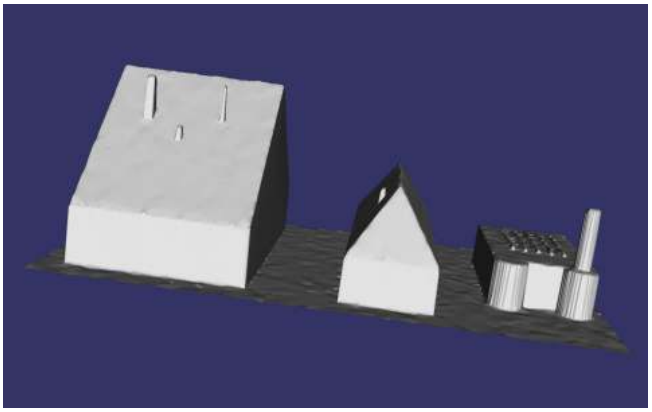
(d) Fusion using pixel based median filtering, $SNR = 28.39$



(e) Fusion using combined medmean filtering, $SNR = 29.35$

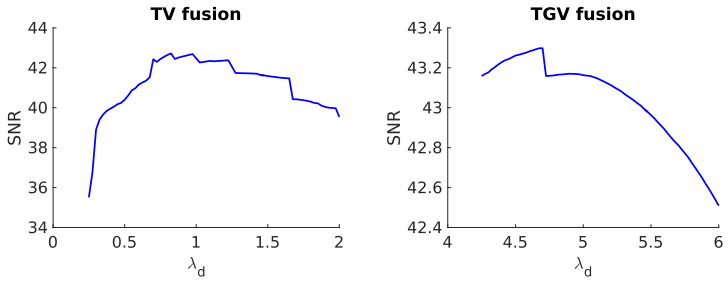


(f) TV-L_1 fusion, $SNR = 42.72$



(g) TGV-L_1 fusion, $SNR = 43.30$

Fig. 2. Comparison of local fusion method versus global optimization methods. Both numerical results and visual appearance show the benefit of the latter ones.

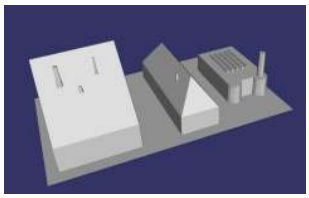


(a) Groundtruth data

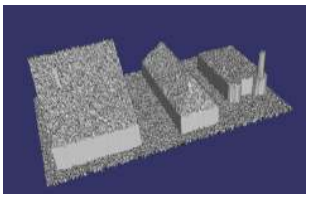
(b) 1 of 5 noisy input images

Fig. 3. SNR values with varying λ_d to obtain best parameter.

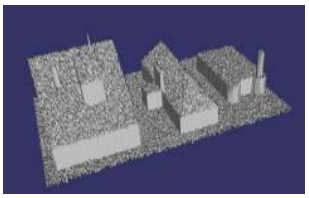
In Figure 5 the two abovementioned synthetic input DSMs are depicted, together with the corresponding fusion results of either using only one input DSM or adding the second sparse DSM to the optimization process as well. The latter case improves the accuracy, if not by very much. But please note that the sparsity of the second DSM is only $1/16 = 6.25\%$ compared to the



(a) Groundtruth data

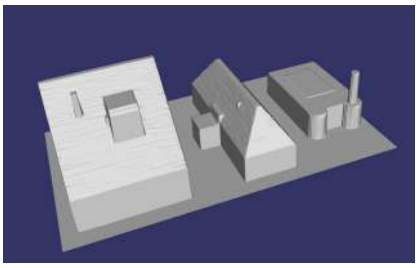


(b) 2 of 5 input images with added noise

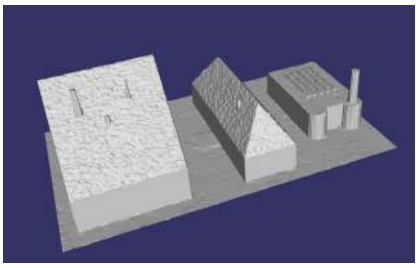


(c) 3 of 5 input images with added noise and having wrong systematic errors

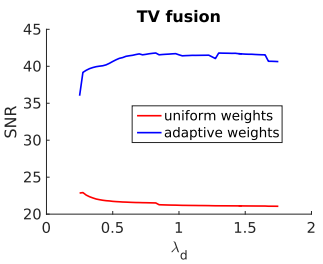
(d) Corresponding weights used in the optimization scheme: The areas with systematic error are given a weight $w = 0.2$ (dark gray), for the rest $w = 1.0$ (light gray).



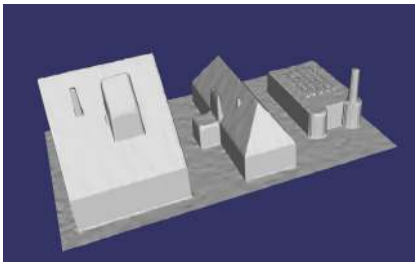
(e) TV-L_1 fusion, $SNR = 22.91$



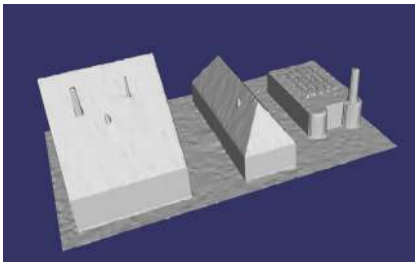
(f) Weighted TV-L_1 fusion, $SNR = 41.80$



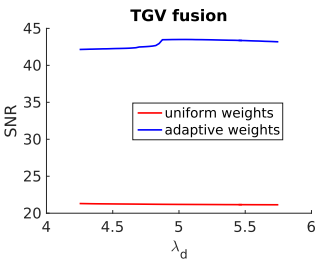
(g) Plotting SNR with varying λ_d



(h) TGV-L_1 fusion, $SNR = 21.31$



(i) Weighted TGV-L_1 fusion, $SNR = 43.50$



(j) Plotting SNR with varying λ_d

Fig. 4. Evaluation of using explicit weights for simulated systematic errors in some of the input data (c). Standard TV-L_1 or TGV-L_1 fusion is not able to remove this systematic error, since it is consistent in 3 of 5 input images. When explicitly down-weighting these areas, (f) and (i), the surface is reconstructed as intended.

first input DSM. For this experiment, both DSMs (or their valid depth pixels respectively) are weighted equally.

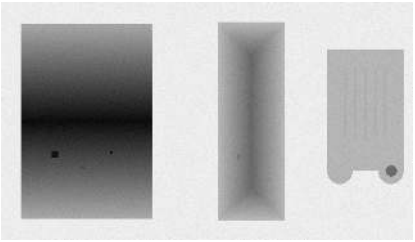
D. Unimodal DSM fusion

In our second evaluation, we created 14 different DSMs of the same 4.5km^2 area of the inner city of Las Vegas using a stereo reconstruction framework as proposed in [3]. For this we have a collection of 60 Skybox images, taken from different positions. The ground sampling distance (GSD) of these images are 1.5m and for evaluation purposes, we obtained a LiDAR measurement of the same area by aerial laser scanning having a point density of 0.375 points per m^2. As the Skybox images were taken from with a high off-nadir angle, areas behind high buildings are occluded, and cannot be reconstructed. Points in the occluded areas were not considered during the statistical evaluation.

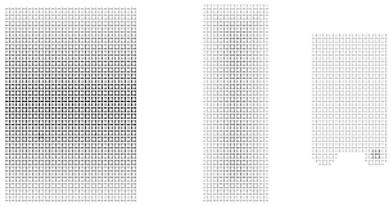
We also created 20 different DSMs of two different areas of London, using 5 in-track WorldView-2 images with a GSD of 0.5 m. First, we focused on a 1km $\times$ 1km area of the inner city of London, and second on a 1.5km $\times$ 1km park area. A LiDAR dataset, with a GSD of 1.0 m is used as reference. A satellite image of each area is shown in Figure 6. Figure 7 shows the computed fused DSMs of the inner city of London using medmean, TGV-L_1 and TV-L_1 fusion.

The accuracy of the fused DSMs with respect to the LiDAR ground truth for the Las Vegas and London data set is given in Tables I, II and III in the common error metrics Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Normalized Median Absolute Deviation (NMAD). Here the improvements are hardly detectable at all, with all algorithms exhibiting similar numerical results. As of yet we do not have further explanation for these results, but strongly suspect the quality of the input DSMs, and of the LiDAR ground truth. Indeed, we noticed and removed some strong outliers in the LiDAR points, but we imagine some less strong outliers were still used during the evaluation.

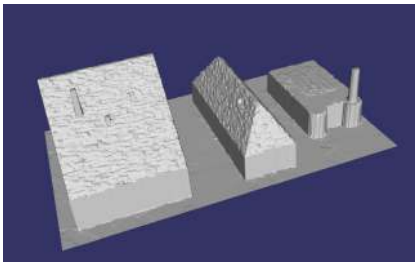
In fact, the statistics appear to be a little better for medmean fusion than for TGV and TV fusion. However, visual inspection of TGV and TV results show less noise and better definition of building boundaries and small streets. This may be due to the fact that for each LiDAR point, we do not calculate the z-axis distance between this point and the DSM, but the Euclidian



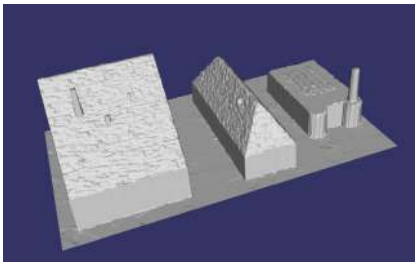
(a) Noisy input DSM #1



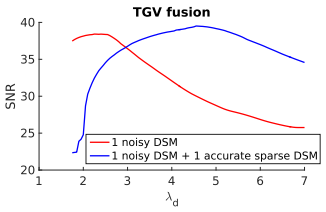
(b) Accurate but sparse input DSM #2. For illustration purpose the sampling points where thickened - the real input DSM has a sparsity of $1/16 = 6.25\%$.



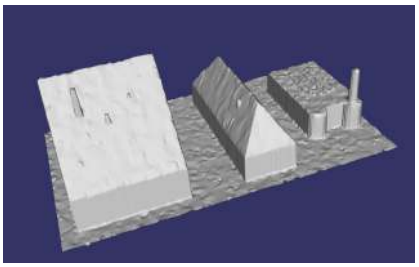
(c) TV-L_1 denoising (1 DSM), $SNR = 37.75$



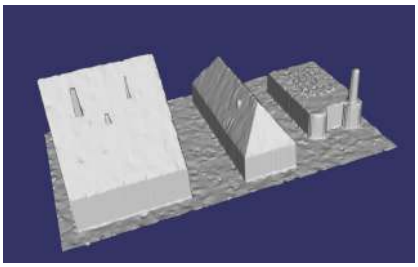
(d) TV-L_1 fusion (2 DSMs of different resolution), $SNR = 38.59$



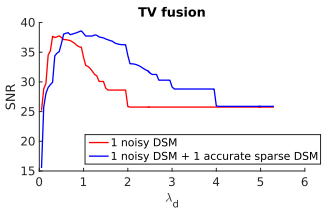
(e) Plotting SNR with varying λ_d



(f) TGV-L_1 denoising (1 DSM), $SNR = 38.41$



(g) TGV-L_1 fusion (2 DSMs of different resolution), $SNR = 39.51$



(h) Plotting SNR with varying λ_d

Fig. 5. Evaluation of fusing DSMs with different ground sampling distance (simulated by a sparsity of 1/16 = 6.25% of the second DSM.

	MAE [m]	RMSE [m]	NMAD [m]
Medmean	1.82	4.06	1.16
TV-L_1	1.93	4.23	1.21
TGV-L_1	1.95	4.22	1.22

TABLE I
LAS VEGAS DATASET: ACCURACY OF THE FUSED DSM W.R.T. GROUND TRUTH OBTAINED BY AERIAL LASERSCANNING (LIDAR)

	MAE [m]	RMSE [m]	NMAD [m]
Medmean	1.36	2.20	1.01
TV-L_1	1.62	2.72	1.16
TGV-L_1	1.63	2.72	1.15

TABLE II
LONDON DATASET (INNER CITY): ACCURACY OF THE FUSED DSM W.R.T. GROUND TRUTH OBTAINED BY AERIAL LASERSCANNING (LIDAR)

	MAE [m]	RMSE [m]	NMAD [m]
Medmean	1.05	1.85	0.65
TV-L_1	1.13	1.99	0.67
TGV-L_1	1.17	2.06	0.68

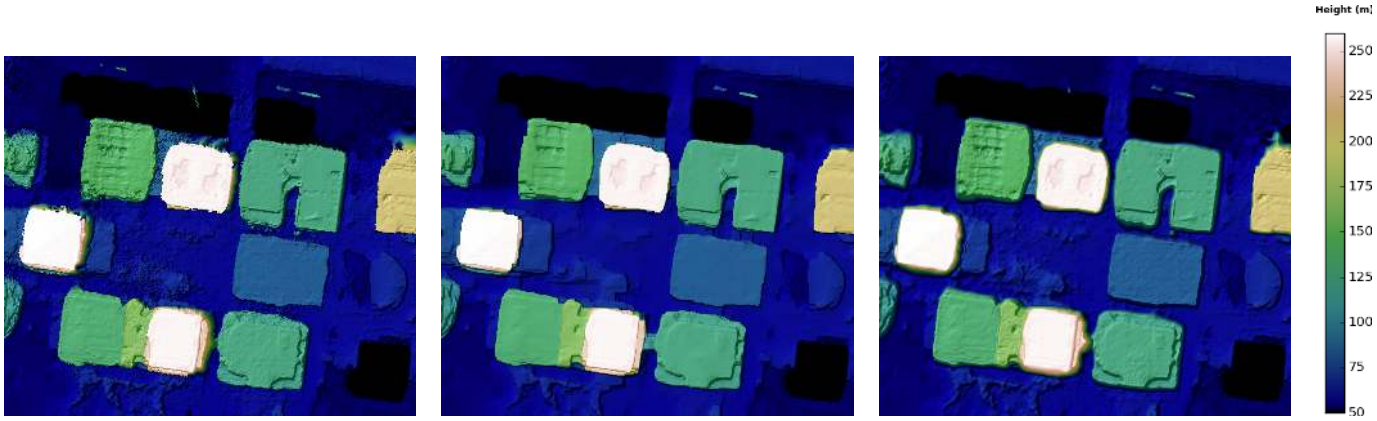
TABLE III
LONDON DATASET (PARK): ACCURACY OF THE FUSED DSM W.R.T. GROUND TRUTH OBTAINED BY AERIAL LASERSCANNING (LIDAR)

distance between LiDAR point and DSM surface. This leads to not taking big outliers into account in the evaluation. For

example, huge outliers located between two buildings will lead to reasonably small errors.
Furthermore, we also noticed that medmean fusion leads to a few visually erroneous results in areas for which the LiDAR data are not defined, and thus are not taken into account in the statistics. We can see those phenomena in Figure 8 : First, on the right side of the building (Zone A), we remark that medmean fusion yields artifacts which are not taken into account in the statistics as no LiDAR points were available for this region.
Second, on the upper edge of the building (Zone B), medmean fusion yields slowly decreasing artifacts, which are approximately 30m high, the building being 255m high, the neighbouring building 85m, and the artifacts having a height of 115m. But as we are taking the Euclidian distance into account for the evaluation, the calculated error in this place is only about 2m which is even a little smaller than for the correct TV-L_1 result.
Last, upper the building (Zone C) we notice an artifact which is a 50m high crane, and which was removed using TV-L_1. Despite this, we observe an error of about 3m for TV-L_1 fusion, and about 50cm for medmean fusion there. Moreover, we can also observe visual differences between TGV-L_1 and medmean fusion in Figure 9. Indeed, the edges seem to be sharper and the surfaces more regular using TGV-L_1 fusion than using medmean fusion. Finally, we also notice two points visualizing the height profiles in Figure 9: First, medmean fusion is indeed less smooth and contains more noise than TV-L_1 and TGV-L_1 fusion. Second, the LiDAR ground truth also contains some outlier points inside and below the buildings, which might additionally compromise the evaluation results.

(a) London inner city (1) (b) London inner city (2) (c) London park (d) Las Vegas

Fig. 6. Input images

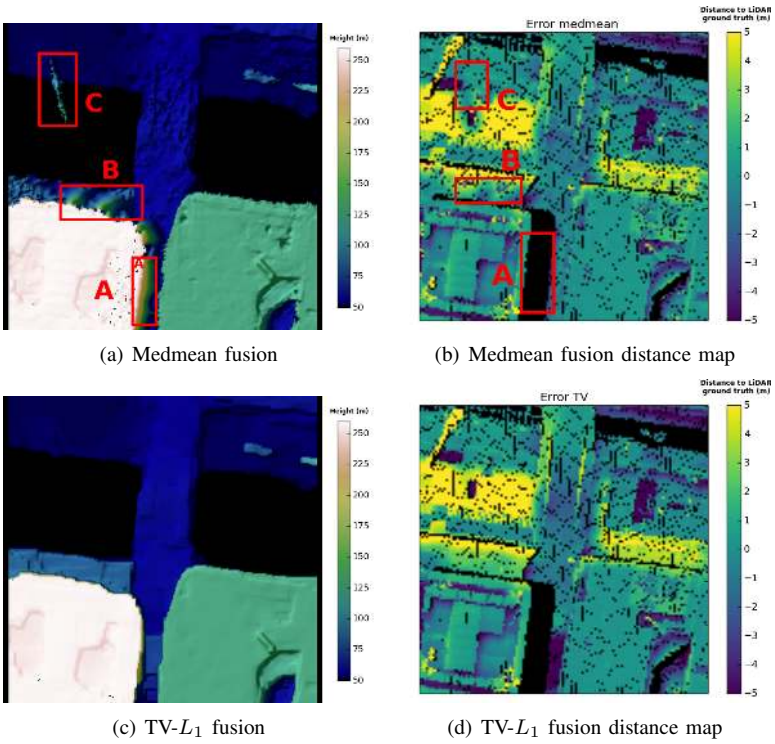


(a) Medmean fusion (b) TV-L_1 fusion (c) TGV-L_1 fusion

Fig. 7. London dataset: medmean, TV-L_1 and TGV-L_1 fusion for inner city (1)

E. Multimodal DSM fusion

Our third evaluation is investigating the results of fusing DSMs derived from different sensors and different spatial resolutions. The test data is taken from the ISPRS benchmark [19] and consists of 3 different scenes (hilly forest = *Vacarisses*, city = *Terrassa*) near Barcelona, Spain. For each scene, we compute DSMs from the a Pleiades triplet and a Worldview-1 stereo pair with a GSD of 1 m. As reference we use a LiDAR point cloud a density of 0.3 points per square meter. DSMs for all 3 possible image pairs of the Pleiades were computed and merged. To evaluate the filtering effect of TV-L_1 and TGV-L_1 the WorldView-1 DSM was additionally processed with the TV and TGV algorithms. The numerical results of local median fusion, global TV-L_1, and TGV-L_1 fusion are given in Table IV.



(a) Medmean fusion (b) Medmean fusion distance map

(c) TV-L_1 fusion (d) TV-L_1 fusion distance map

Fig. 8. London dataset: medmean and TV-L_1 fusion together with distance to LiDAR ground truth

Algorithm	Terrassa			Vacarisses		
	MAE[m]	RMSE[m]	NMAD[m]	MAE[m]	RMSE[m]	NMAD[m]
WV-1	1.05	2.23	0.59	1.62	2.88	1.09
WV-1 TV-L_1	1.04	2.20	0.59	1.81	3.48	1.11
WV-1 TGV-L_1	1.06	2.24	0.59	2.45	6.41	1.12
PL medmean	0.97	1.73	0.68	1.43	2.28	1.30
PL TV-L_1	1.03	1.84	0.67	1.58	2.54	1.38
PL TGV-L_1	1.03	1.85	0.67	1.64	2.76	1.39
PL & WV-1 medmean	0.88	1.62	0.61	1.22	1.98	1.12
PL & WV-1 TV-L_1	1.03	1.84	0.67	1.58	2.54	1.38
PL & WV-1 TGV-L_1	0.98	1.80	0.64	1.41	2.26	1.26

TABLE IV

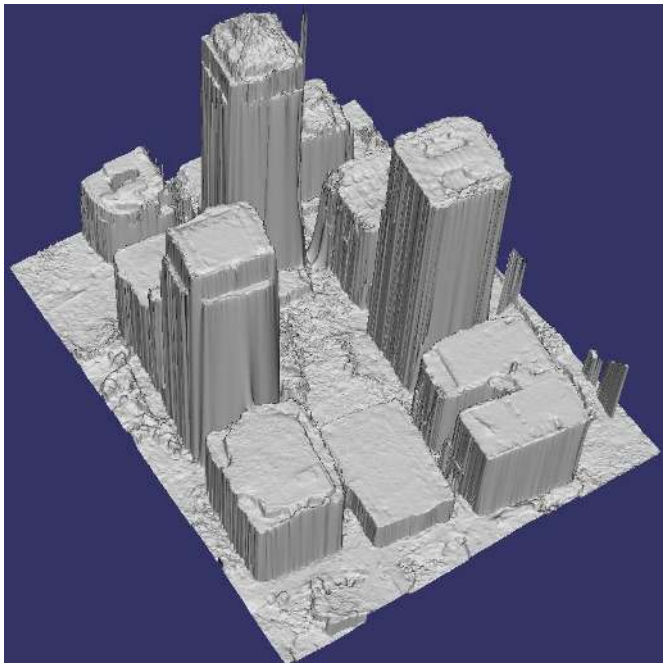
RESULTS OF LOCAL MEDIAN FUSION AND GLOBAL TGV-L_1 FUSION FOR HETEROGENOUS SENSOR DATA (PLEIADES AND WORLDVIEW-1 SATELLITE IMAGES). THE FIRST ROW SHOWS THE UNFUSED RESULT FOR WORLDVIEW-1 STEREO PAIR, THE NEXT 2 LINES A "SMOOTHING" WITH TV AND TGV. RESULTS FOR MERGING THE INDIVIDUAL STEREO PAIRS OF THE PLEIADES TRIPLET ARE SHOWN IN LINE 3 TO 6, AND A FUSION OF PLEIADES AND WORLDVIEW-1 DSMS IS SHOWN IN THE LAST 3 LINES.

While the filtering of the WorldView-1 DSM does not significantly change the statistics for the Terrassa dataset, which to a larger extend consists of manmade structures and fields, the filter has a stronger smoothing effect on the mainly forested and hilly landscape of the Vacarisses area. A larger RMSE value is observed for the TGV-L_1 solution. In this special case, the TGV solution propagated outliers in the textureless shadow areas, and at steep slopes, leading to worse results. As in the London areas, objects such as building contours and bridges appear sharper, but this effect cannot be mesured properly by the relatively sparse LiDAR reference data.

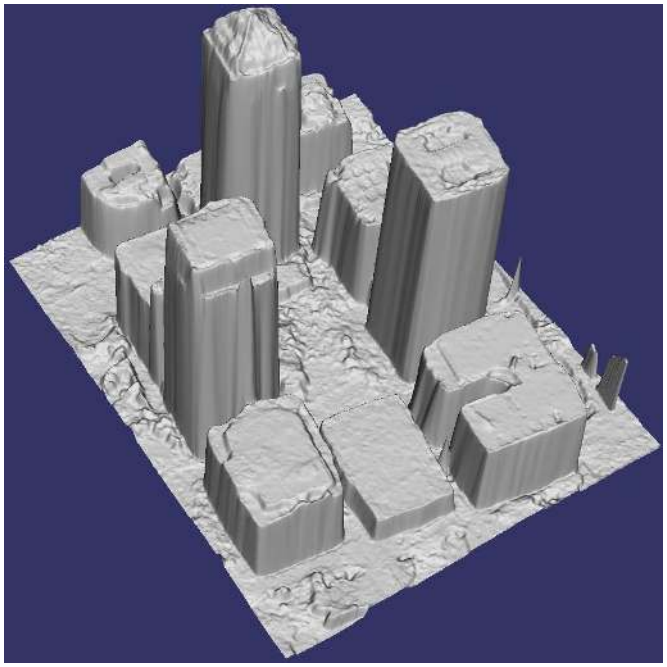
V. CONCLUSION

IN this paper we proposed a global optimization algorithms for fusing multi-resolution DSMs obtained by heterogenous sensors. These global optimization algorithms are based on adaptively weighted TV-L_1 and TGV-L_1 optimization problems, allowing for fusion of multiple DSMs enforcing additional spatial regularization. As a result, single pixels are not fused independently but a local consensus about the optimal height is achieved by taking all valid measurements in a local neighborhood into account and additionally enforcing a local planarity assumption.

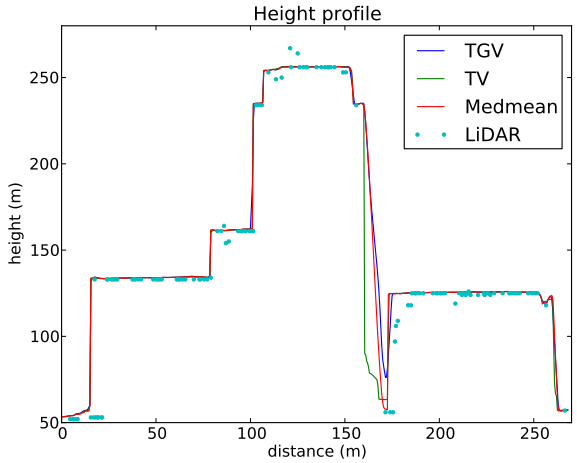
In all different evaluations, both synthetic and real world data sets, a significant improvement of the visual accuracy was shown. However, the numerical accuracy is only superior for the synthetic data sets, as the ground truth for the real world data sets is too sparse and unevenly distributed - we again refer strongly to Figure 9 illustrating this problem. As a result, our



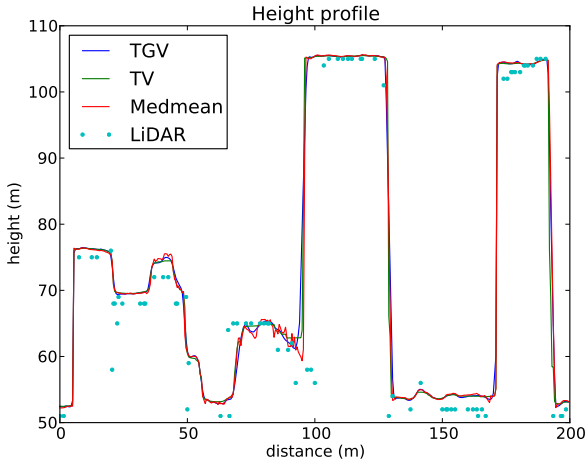
(a) Result of medmean fusion



(b) Result of TGV fusion



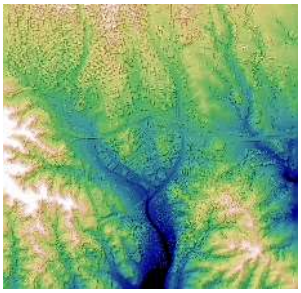
(c) Height profile, see Figure 6(a)



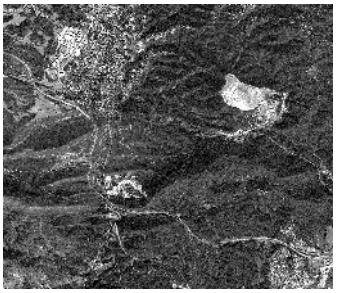
(d) Height profile, see Figure 6(b)

Fig. 9. London dataset inner city: Fusion results

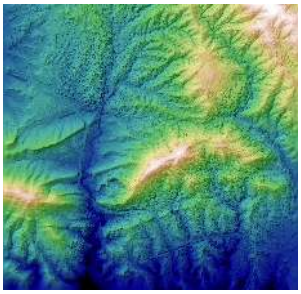
(a) WorldView-1, *Terrassa*



(b) fused DSM, *Terrassa*



(c) WorldView-1, *Vacarisses*



(d) fused DSM, *Vacarisses*

Fig. 10. ISPRS dataset: Exemplary WorldView-1 images of the scenes used in the evaluation

future work will especially focus on obtaining detailed 3D ground truth within ground sampling distance of the corresponding sensors to evaluate.

ACKNOWLEDGMENT

The authors would like to thank European Space Imaging (EUSI) for providing the WorldView-2 data of London, and the SkyBox video of Las Vegas. For the datasets near Barcelona, we thank DigitalGlobe for the Worldview-1 data, Airbus Space and Defense for the Pleiades data and Institut Cartogràfic i Geològic de Catalunya for the LiDAR Reference.

REFERENCES

[1] C. Straub, J. Tian, R. Seitz, and P. Reinartz, "Assessment of cartosat-1 and worldview-2 stereo imagery in combination with a lidar-dtm for timber volume estimation in a highly structured forest in germany," *Forestry*, vol. 86, no. 4, pp. 463–473, 2013.

[2] H. Hirschmueller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 328–341, 2008.

[3] G. Kuschk, "Large Scale Urban Reconstruction from Remote Sensing Imagery," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 5, p. W1, 2013.

[4] M. Rumpler, A. Irschara, A. Wendel, and H. Bischof, "Rapid 3D City Model Approximation from Publicly Available Geographic Data Sources and Georeferenced Aerial Images."

[5] H. Schultz, E. M. Riseman, F. R. Stolle, and D.-M. Woo, "Error Detection and DEM Fusion using Self-Consistency," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE InternationalConference on*, vol. 2. IEEE, 1999, pp. 1174–1181.

[6] P. Reinartz, R. Mueller, D. Hoja, M. Lehner, and M. Schroeder, "Comparison and Fusion of DEM Derived from SPOT-5 HRS and SRTM Data and Estimation of Forest Heights," in *Proc. EARSeL Workshop on 3D-Remote Sensing, Porto*, 2005.

[7] H. Papasaika, E. Kokiopoulou, E. Baltsavias, K. Schindler, and D. Kressner, "Fusion of Digital Elevation Models using Sparse Representations," *Photogrammetric Image Analysis*, pp. 171–184, 2011.

[8] K. Schindler, H. Papasaika-Hanusch, S. Schuetz, and E. Baltsavias, "Improving Wide-Area DEMs Through Data Fusion - Chances and Limits," *Proceedings of the 54th Photogrammetric Week, Stuttgart, Germany*, 2011.

[9] G. Kuschk and D. Cremers, "Fast and Accurate Large-scale Stereo Reconstruction using Variational Methods," in *ICCV Workshop on Big Data in 3D Computer Vision*, Sydney, Australia, December 2013.

[10] C. Zach, T. Pock, and H. Bischof, "A Globally Optimal Algorithm for Robust TV-L1 Range Image Integration," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conferenceon*. IEEE, 2007, pp. 1–8.

[11] C. Zach, "Fast and High Quality Fusion of Depth Maps," in *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, vol. 1, 2008.

[12] T. Pock, L. Zebedin, and H. Bischof, "TGV-Fusion," *Rainbow of computer science*, pp. 245–258, 2011.

[13] R. Perko and C. Zach, "Globally optimal robust dsm fusion," *European Journal of Remote Sensing*, vol. 49, pp. 489–511, 2016.

[14] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[15] K. Bredies, K. Kunisch, and T. Pock, "Total Generalized Variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.

[16] A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging," *Journal of Mathematical Imaging and Vision*, pp. 1–26, 2011.

[17] G. Kuschk and P. d'Angelo, "Fusion of Multi-Resolution Digital Surface Models," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-1/W3, pp. 247–251, 2013.

[18] Y. Li and S. Osher, "A New Median Formula with Applications to PDE Based Denoising," *Commun. Math. Sci*, vol. 7, no. 3, pp. 741–753, 2009.

[19] P. Reinartz, P. d'Angelo, T. Krauß, D. Poli, K. Jacobsen, and G. Buyuksalih, "Benchmarking and Quality Analysis of DEM Generated from High and Very High Resolution Optical Stereo Satellite Data," in *ISPRS Symposium Commission I*, 2010.

Georg Kuschk Georg Kuschk received his diploma degree (Dipl.-Inf) in computer science in 2007 from the Martin-Luther-University Halle-Wittenberg, Halle, Germany. After working in Augmented Reality research industry, he joined the *Photogrammetry and Image Analysis* department of the Remote Sensing Technology Institute at the German Aerospace Center (DLR) in 2011 as a research scientist. His research topics include computer vision, image processing, inverse problems, numerical optimization and image based 3D reconstruction in particular.

Pablo d'Angelo Pablo d'Angelo received his diploma degree in computer engineering (Dipl.-Ing FH) from the University of Applied Scienes Ulm in 2004. He has worked with Daimler AG from 2004 to 2007 on industrial computer vision and received his Ph.D (Dr.-Ing.) in computer science from Bielefeld University in 2007 with a dissertation on joint use of geometric and photometric methods for 3D reconstruction from optical images. In 2007, he joined the *Photogrammetry and Image Analysis* department of the Remote Sensing Technology Institute at the German Aerospace Center (DLR) in Oberpfaffenhofen. His main research topic are 3D reconstruction from remotely sensed stereo imagery, with a focus on operational systems for large scale image orientation and generation of digital elevation models.

David Gaudrie David Gaudrie is currently in the fourth grade of his Mathematical Engineering and Modeling studies at the National Instiute of Applied Sciences (INSA) of Toulouse. In 2015, he did a three-month internship at the *Photogrammetry and Image Analysis* department of the Remote Sensing Technology Institute at the German Aerospace Center (DLR) in Oberpfaffenhofen. During his internship, he was involved with fusion of digital elevation models.

Peter Reinartz Peter Reinartz received his Diploma (Dipl.-Phys.) in theoretical physics in 1983 from the University of Munich and his PhD (Dr.-Ing) in civil engineering from the University of Hannover, in 1989. His dissertation is on optimization of classification methods for multispectral image data. Currently he is department head of the department *Photogrammetry and Image Analysis* at the German Aerospace Centre (DLR), Remote Sensing Technology Institute (IMF) and holds a professorship for geomatics at the University of Osnabrueck. He has more than 25 years of experience in image processing and remote sensing and over 300 publications in these fields. His main interests are in direct georeferencing, stereo-photogrammetry and data fusion of space borne and airborne data, generation of digital elevation models and interpretation of VHR data from sensors like WorldView, GeoEye, and Pleiades. He is also engaged in using remote sensing data for disaster management and using high frequency time series of airborne image data for real time image processing and their operational use in case of disasters as well as for traffic monitoring.

Daniel Cremers Daniel Cremers received Bachelor degrees in Mathematics (1994) and Physics (1994), and a Master's degree in Theoretical Physics (1997) from the University of Heidelberg. In 2002 he obtained a PhD in Computer Science from the University of Mannheim, Germany. Subsequently he spent three years at the University of California at Los Angeles (UCLA) and at Siemens Corporate Research in Princeton, NJ. From 2005 until 2009 he was associate professor at the University of Bonn, Germany. Since 2009 he holds the chair for Computer Vision and Pattern Recognition at the Technical University of Munich. His publications received several awards, including the 'Best Paper of the Year 2003' (Int. Pattern Recognition Society), the 'Olympus Award 2004' (German Soc. for Pattern Recognition) and the '2005 UCLA Chancellor's Award for Postdoctoral Research'. In December 2010 he was listed among Germany's top 40 researchers below 40 (Capital). In 2016, Prof. Cremers received the Gottfried-Wilhelm-Leibniz Award, the biggest award in German academia.