# Stereo Depth Map Fusion for Robot Navigation

Christian Häne[1], Christopher Zach[1], Jongwoo Lim[2], Ananth Ranganathan[2] and Marc Pollefeys[1]

Department of Computer Science[1]
ETH Zürich, Switzerland

Honda Research Institute[2]
Mountain View, CA, USA

*Abstract*— **We present a method to reconstruct indoor environments from stereo image pairs, suitable for the navigation of robots. To enable a robot to navigate solely using visual cues it receives from a stereo camera, the depth information needs to be extracted from the image pairs and combined into a common representation. The initially determined raw depthmaps are fused into a two level heightmap representation which contains a floor and a ceiling height level. To reduce the noise in the height maps we employ a total variation regularized energy functional. With this 2.5D representation of the scene the computational complexity of the energy optimization is reduced by one dimension in contrast to other fusion techniques that work on the full 3D space such as volumetric fusion. While we show only results for indoor environments the approach can be extended to generate heightmaps for outdoor environments.**

## I. INTRODUCTION

To enable a robot to navigate with visual data captured from a stereo camera it needs to be able to have some kind of representation of the observed world that is suitable for this task. There is a large literature on visual simultaneous localization and mapping (VSLAM) generating a sparse representation of the environment and the respective camera/robot poses. However such a sparse representation generally only includes points that are salient enough in the considered images and thus lie in well textured regions. For instance, uniformly painted walls and homogeneous regions in general are not covered by the model obtained from VSLAM and therefore problematic for robot navigation.

The representation of the robot's surroundings needs to be appropriate for higher-level tasks such as path planning and collision avoidance. Although one could use the reconstructed sparse feature points from a VSLAM pipeline directly for collision avoidance, this can potentially miss whole objects that lack some feature points. In such a case the possible obstacle would just be invisible for the robot. In indoor environments such objects are quite common e.g. a uniformly colored wall. Thus a denser representation of the scene is favorable.

Many local and global stereo methods are available to determine depth from images, but the returned depth maps are generally still contaminated by inaccurate and erroneous depth estimates. This problems emerge especially in textureless parts of images, which are quite common in indoor environments. Although global stereo methods optimize an energy functional over the whole image domain in order to hypothesize reasonable depth values in ambiguous regions, the resulting depth maps may still contain errors.

In order to overcome this problem we combine several depth maps calculated from multiple stereo images into a common representation. Thus, inconsistencies between several acquired depth maps can be discovered and removed. We propose to use a height map based representation of the environment for the globally combined representation of the scene. For indoor environments the height map is composed of two levels, a floor and a ceiling level.

Our target setting is a humanoid robot that is equipped with a stereo camera, which enables it to observe its surroundings. The two cameras are mounted on a stereo rig with fixed relative position. It is further assumed that the intrinsic calibration of the two cameras are known. For the fusion procedure the camera poses are needed as well. For this we use the output of existing VSLAM pipelines.

The remainder of the paper is structured as follows. In section II prior work related to this paper is presented. Our height map based reconstruction approach is explained in section III. In section IV we show the results we obtained for indoor environments and compare them to a full 3D reconstruction of the scene. Finally we draw some conclusions in section V.

## II. Related Work

There exists a vast literature on computational stereo, fusion of range data, and general 3D modeling from images. We focus on approaches suitable for efficient implementation or explicitly addressing reconstruction of indoor environments. Merell et al. [1] propose a visibility-based approach capable of real-time operation to fuse a sequence of nearby depthmaps to a single depthmap with a higher confidence. This is done by projecting the depthmaps to a reference view and then choosing one of the depths projecting to the same position that reduces the number of visibility conflicts. A visibility conflict is the situation where a measurement lies in the free space of some other measurement. While this gives more accurate depthmaps, which could be certainly used for collision avoidance there is no direct way to generate a globally consistent representation of the complete observed space. Because there is no regularization used in this fusion the resulting depthmaps still contain some noise.

In order to obtain globally consistent 3D models from a set of range images Zach et al. [2], [3] use an implicit representation of the space. The input depthmaps are converted to signed distance fields. A total variation energy functional with a $L_1$ distance data fidelity term, is defined on a regularized signed distance field that simultaneously approximates all the input fields. This convex energy functional is minimized to get the final reconstructed scene.

In [4] Furukawa et al. use the Manhattan world assumption, which means that all the surfaces are oriented according to the coordinate axes, to formulate a stereo algorithm that is able to handle flat textureless surfaces. From the directions obtained by the patch-based multi view stereo (PMVS) software [5], which extracts a semi-dense oriented point cloud from a set of input images with known camera orientations, the three orthogonal main directions are extracted. Afterwards the problem of assigning a plane to each of the pixels is formulated as a Markov random field (MRF) and solved using the graph cut method. This finally leads to a depthmap that contains flat surfaces oriented according to the three main directions. In [6] they use this depthmaps to reconstruct building interiors. In a similar way as in [2], [3] they integrate the depth maps to a volumetric structure. Here the voxels have a binary label interior or exterior. To get a full labeling of the space that best approximates all the given input depthmaps a MRF is formulated and solved using graph cuts. This leads to complete polygonal meshes of building interiors but comes with a high computational cost: reported run-times range from a few hours to multiple days.

Gallup et al. [7], [8] use a height map representation to reconstruct urban environments. They integrate the raw depth maps to a three dimensional occupancy grid, which is aligned such that the $z$-axis points to the upright direction. For each voxel a number is stored, free space gets a negative weight and occupied space a positive weight. The space in front of the measured depth is assumed to be free and the space behind the measurement is expected to be occupied with an exponential drop of the weight. One or multiple height levels are then extracted by getting the minimum of an energy functional for each $z$-column. Because each column is optimized independently, this can be calculated very efficiently by GPUs, but has the drawback that there is no regularization between the columns.

In this work we focus on representing indoor environments by two level height maps. But in contrast to [7], [8] we use a total variation energy functional to obtain regularized results. This leads to reconstructions where most of of the noise is removed. It has the benefit over full 3D reconstructions like the one in [2], [3] that it can be computed faster due to the reduced representation, which is still able to represent the scene accurately enough to be used for robot navigation.

## III. TWO LEVEL HEIGHT MAPS

As a prerequisite for a humanoid robot to do path planning tasks it needs to know the ground where it is able to move around. By knowing the height profile of the floor, the area where it is regular enough for the movement capabilities of the robot can be determined. The additional knowledge of the ceiling position enables a path planning software to decide if the robot would fit into the free space. This informations combined define the area where the robot is able to pass. According to these considerations it makes sense to constrain the final reconstruction to be represented by a floor and ceiling height level (Fig. 1).

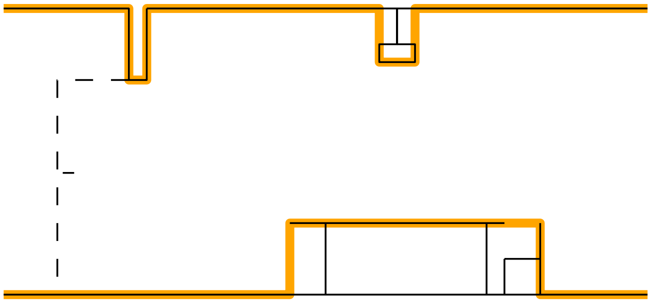

Fig. 1. Two level height map, the floor and the ceiling are approximated by a height level.

### A. Depth Integration

As first step the raw depthmaps are integrated into a volumetric representation of the space. This is done by creating an occupancy grid that stores a negative number for free space and a positive one for occupied space. With increasing absolute value the decision for free or occupied space gets more certain. This is similar to the probabilistic formulation of occupancy grids e.g. [9]. The $z$-axis is aligned with the upright direction of the scene such that the discontinuities in the height map are vertical walls. We detect the upright direction in the input images by detecting the associated vanishing point.

In order to integrate a new depthmap to the common volumetric representation, each voxel is projected onto the new depthmap. For every voxel $v$ the weight that it adds to the grid is calculated. It is dependent on the depth $z_v$ that the voxel $v$ has according to the new view and on the depth measurement $z_p$ of the depthpixel $p$ to which $v$ projects. Due to the constant disparity resolution in the stereo image the depth resolution is decreasing with increasing depth [10]. To include this into our cost function a band around the measurement whose width is dependent on the measured depth $z_p$ is defined. The width of the band is given by,

$$l_p(z_p) = \max\left\{\frac{z_p^2}{bf}\delta_d, \varepsilon\right\}. \tag{1}$$

The first operand approximates the depth uncertainty [10], where $\delta_d$ is the disparity resolution, $b$ the stereo rig base line length and $f$ the focal length in image pixels. To ensure a minimal width of the band the constant $\varepsilon$ is used as a lower bound. This guarantees that the weight is spread to at least some voxels. Having all the weight accumulated in too few voxels could result in very noisy reconstructions. The actual weight added to the grid inside this band has the same absolute value for each voxel with the appropriate sign for free and occupied space. The region outside this band is weighted differently. Being further away from the camera center than the band around $z_p$ means that we are already behind the surface seen in the image, which means

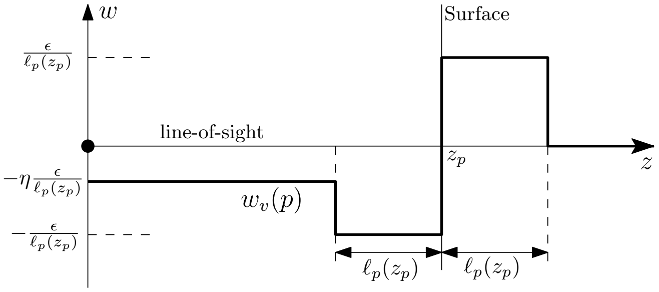
$w$
Surface
$\frac{\epsilon}{\ell_p(z_p)}$
line-of-sight
$-\eta\frac{\epsilon}{\ell_p(z_p)}$
$w_v(p)$
$z_p$
$z$
$-\frac{\epsilon}{\ell_p(z_p)}$
$\ell_p(z_p)$
$\ell_p(z_p)$

Fig. 2. Weight that a depth pixel $p$ adds to the occupancy grid.

that nothing is known about the space being free or occupied and thus no weight is added to the grid. Being nearer to the camera center means that the space should be free, but adding the same weight to the voxels lying on this viewing ray would result in adding much more weight for free space than occluded space to the grid for one depth measurement. This gets problematic if a depth measurement is erroneously too far away. To account for this, the weight entered into this part is reduced by a factor $\eta < 1$. The final weight that is added to the voxel $v$ projected onto a depth pixel $p$ is the following:

$$w_v(p) = \begin{cases} \frac{\epsilon}{l_p(z_p)} & \text{if } z_v \geq z_p \wedge (z_v - z_p) \leq l_p(z_p) \\ -\frac{\epsilon}{l_p(z_p)} & \text{if } z_v < z_p \wedge (z_p - z_v) \leq l_p(z_p) \\ -\eta\frac{\epsilon}{l_p(z_p)} & \text{if } z_v < z_p \wedge (z_p - z_v) > l_p(z_p) \\ 0 & \text{else.} \end{cases} \quad (2)$$

The absolute value of the weight is dependent on the band width $l_p(z_p)$, but is at most one for a single voxel. Inside the band the individual weights are normalized such that the sum of all the weights that are entered for one measurement stays constant. By using this weight function a measurement that is less certain adds the same total weight to the grid than a more certain one. The weight is just spread further around the measurement into the grid.

Multiple weights for the same voxel are accumulated by summing them up. A visualization of the weight function is shown in Fig. 2. An example of a slice of the occupancy grid is given in Fig. 3.

### B. Minimal Cost Height Levels

The aim of this paper is to get regularized height values of a floor and a ceiling level for an indoor environment. In the following sections we will define appropriate energy functionals that can be minimized efficiently in order to extract the final regularized height levels. However they depend on minimal cost height levels that are extracted for each $z$-column of the voxel grid independently. The procedure to extract them is introduced in the remainder of this section.

For each point $(x,\ y)$ in the height map the two height values $\underline{h}$ and $\overline{h}$ for floor and ceiling need to be determined. This is done by minimizing the cost

$$C_{x,y}(\overline{h},\underline{h}) = -\sum_{z<\underline{h} \text{ or } z\geq\overline{h}} w_{x,y}(z) + \sum_{\underline{h}\leq z<\overline{h}} w_{x,y}(z) \quad (3)$$

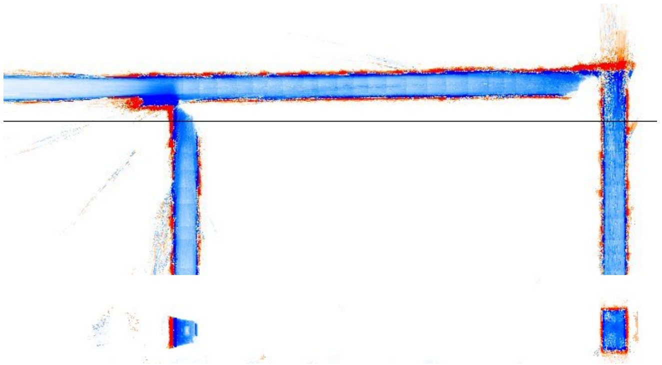

Fig. 3. Horizontal (top) and vertical (bottom) slice through the occupancy grid. The blue colors denote negative and the red colors positive values. With the black line the position of the vertical slice is indicated. The absolute value of the weights is visualized by the saturation of the colors. (best viewed in color)

where $w$ denotes the accumulated weight of a voxel. In order to calculate the minimum of the data cost function it is expressed in a recursive formulation. This is done in the same way as in [8], there a similar function is used for an $n$ layer configuration. The minimum can then be calculated by dynamic programming. After having the table filled in, the height values that minimize the cost function are extracted by backtracking.

At positions where no floor or ceiling is observed it happens that the whole $z$-column has only negative entries. They are lying in the middle of the $z$-column and the bottom and top part are unobserved and thus filled with zeros. An example of such a situation can be taken from the left junction area of the vertical slice in Fig. 3. In such cases the minimal cost height levels are not unique. All floor positions below the negatively weighted part and all ceiling positions above it would result in a global minimum for the regarded $z$-column. Section III-D introduces a convex approximation to the datacost function $C_{x,y}(\overline{h},\underline{h})$, which is used to get regularized height levels. The chosen approximation only considers the cost function around a neighborhood of the extracted minimum. Therefore for non unique minima the specific choice is not done arbitrarily. Because only the region around the minimum is approximated it is beneficial to have non zero weight values inside it. This is achieved by choosing the minimum that has the smallest floor to ceiling distance. For the mentioned case with only negative entries this aligns the floor and ceiling level around the negative weights. If there are still multiple minima with the same floor to ceiling distance one of them is taken arbitrarily.

### C. Regularized Labeling

The regularized height levels are extracted in two stages. The final height values for the floor and ceiling are calculated by two consecutive minimizations of appropriate energy functionals. In a first step the space is partitioned into a region where the whole space is occupied, and into a region where there was some free space observed. Thus in a second step a height for the floor and ceiling needs to be calculated for this area. The remainder of this section explains how the regularized labeling is calculated. Eventually the next section introduces the regularization functionals for the actual height levels, which lead to the final reconstruction.

There are regions in the heightmap, where it makes no sense to have a floor and a ceiling because it is unobserved or just completely occupied e.g. a wall spans the whole height of the grid. At such positions the floor and ceiling would only fit to noise. To prevent this it is ensured that there is enough evidence that there really is a floor and a ceiling in the regions where the height levels are calculated. This can be done by minimizing an appropriate total variation energy functional.

$$\begin{aligned} E^{\text{Labeling}} = &\int_\Omega |\nabla l| + \\ &+ \lambda_l (l(C_{x,y_{\min}} + \gamma) + (1-l) C_{x,y_{\text{occ}}}) \mathrm{d}\mathbf{x} \end{aligned} \tag{4}$$

$C_{x,y_{\min}}$ and $C_{x,y_{\text{occ}}}$ describe the data cost $C_{x,y}$ for the two possible labelings $l = 1$, for regions with height levels and $l = 0$ for completely occupied areas. $C_{x,y_{\min}}$ is the minimal data cost from section III-B, which is used to represent the cost for having a floor and a ceiling. In the other case the floor and ceiling collapse, which means that the space is completely occupied in that column. The position of the two height levels does not affect the cost $C_{x,y}$ in this case, which leads to $C_{x,y_{\text{occ}}} = C_{x,y}(h, h)$. We relax the domain of $l$ to the set $[0, 1]$ to get a convex energy functional, which means that in between the two cases the cost function is linearly interpolated. The parameter $\gamma$ is used as a penalty for choosing the more complex model with a floor and a ceiling. The total variation part of the energy controls the smoothness of the labeling and $\lambda_l$ is used to weight the data fidelity.

To apply the same optimization technique as proposed in [2] for similar energy functionals, we further relax the already convex energy functional to

$$\begin{aligned} E_\theta^{\text{Labeling}} = &\int_\Omega |\nabla l_u| + \frac{1}{2\theta}(l_u - l_v)^2 \\ &+ \lambda_l (l_v (C_{x,y_{\min}} + \gamma) + (1 - l_v) C_{x,y_{\text{occ}}}) \mathrm{d}\mathbf{x}. \end{aligned} \tag{5}$$

The regularization and the data fidelity cost are now defined on two separate scalar fields $l_u$ and $l_v \in [0, 1]$. The quadratic term $\frac{1}{2\theta}(l_u - l_v)^2$ ensures that $l_u$ and $l_v$ are similar enough to get reasonable results for the minimizer. This relaxation allows to minimize the functional by alternating between minimizing according to $l_u$ and $l_v$.

- Letting $l_v$ fixed and minimize for the first two terms the functional to minimize becomes to,

$$\int_\Omega |\nabla l_u| + \frac{1}{2\theta}(l_u - l_v)^2 \mathrm{d}\mathbf{x}. \tag{6}$$

  This energy functional is known as ROF energy and can be minimized efficiently by the gradient descent/reprojection algorithm from [11].
- Letting $l_u$ fixed the minimization can be calculated analytically. This can be done point-wise as the $l_v$ in the remaining functional

$$\begin{aligned} &\int_\Omega \frac{1}{2\theta}(l_u - l_v)^2 \\ &\quad + \lambda_l (l_v (C_{x,y_{\min}} + \gamma) + (1 - l_v) C_{x,y_{\text{occ}}}) \mathrm{d}\mathbf{x}, \end{aligned} \tag{7}$$

  is not dependent on its spatial context.

The algorithm used to minimize the regularization part of the energy functional was proposed in [11]. It is an important part of our method thus we briefly mention the main result. By standard duality arguments it is shown that the minimizer of Eq. (6) is given by

$$\hat{l}_u = l_v + \theta \operatorname{div} \mathbf{p}. \tag{8}$$

$\mathbf{p} \in \mathbb{R}^2$ is the dual vector field. It can be computed by the following gradient descent/reprojection scheme:

$$\mathbf{p}^{n+1} = \frac{\mathbf{p}^{\mathbf{n}} + \tau \nabla(l_v/\theta + \operatorname{div}\mathbf{p}^n)}{\max\{1, |\mathbf{p}^n + \tau\nabla(l_v/\theta + \operatorname{div}\mathbf{p}^n)|\}}, \tag{9}$$

with $\tau \leq 1/4$.

For minimizing $E_\theta^{\text{Labeling}}$ according to Eq. (7) the derivative with respect to $l_v$ is calculated. By setting the derivative to zero the following update rule is deduced.

$$\hat{l}_v = \operatorname{clamp}_{[0,1]}(\theta\lambda_l(C_{\min} - C_{\text{occ}} + \gamma)) \tag{10}$$

The clamping to the interval $[0, 1]$ is necessary because otherwise $l_v$ would increase arbitrarily for positions where the space is labeled to have floor and ceiling and decrease arbitrarily otherwise. In this case the minimal energy would no longer be bounded.

The minimization strongly pushes the labelings $l_u$ and $l_v$ to be binary although they are not constrained to be binary. In order to get a final binary labeling a value of $l_u > 0.5$ is defined to be a position with floor and ceiling, which leads to a new domain $\Omega_{\text{inside}}$.

### *D. Regularized Height Levels*

The height values $\underline{h}$ and $\overline{h}$ for floor and ceiling are only calculated in the reduced domain $\Omega_{\text{inside}}$. Again a total variation energy functional is employed to get regularized height levels. It is chosen as

$$E^{\text{Height}} = \int_{\Omega_{\text{inside}}} |\nabla\underline{h}| + |\nabla\overline{h}| + \lambda_h C_{x,y}(\overline{h}, \underline{h}) \mathrm{d}\mathbf{x} \tag{11}$$

This energy functional is not convex because of the non convex data fidelity term. To overcome this problem we use a convex approximation to the data fidelity term as a relaxation. This guarantees that a global optimum can always be found. The approximation is defined as:

$$\begin{aligned} C_{x,y}^{\text{conv}}(\overline{h}, \underline{h}) = &\alpha_1[\overline{H} - \overline{h}]^+ + \alpha_2[\overline{h} - \overline{H}]^+ + \\ &\alpha_3[\underline{H} - \underline{h}]^+ + \alpha_4[\underline{h} - \underline{H}]^+ + \\ &C_{x,y}(\overline{H}, \underline{H}), \end{aligned} \tag{12}$$

with $\overline{H}$ and $\underline{H}$ being the ceiling and floor position that minimize $C_{x,y}$ determined according to section III-B. The operator $[\cdot]^+$ has the meaning

$$[a]^+ = \begin{cases} a & \text{if} \quad a > 0 \\ 0 & \text{else.} \end{cases} \tag{13}$$

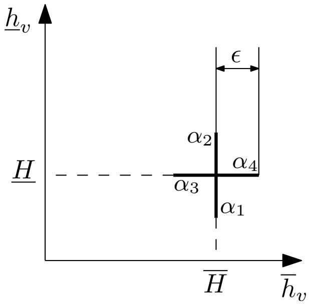

Fig. 4. Subset of the cost function used for the convex approximation

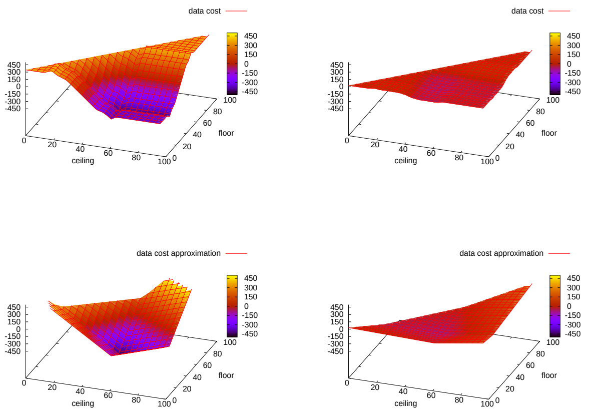

Fig. 5. Top row original data cost functions for a clear and vague minimum. Bottom row convex approximations.

The $\alpha$s are parameters of the cost function that need to be defined for each position in the heightmap. This is done by using a least squares approximation to the original datacost function. By using only a subset of the values for the approximation each $\alpha$ can be approximated independently. For the approximation of $\alpha_1$ and $\alpha_2$ only the position of the ceiling $\overline{h}$ is varied, the position of the floor is fixed at $\underline{h} = \underline{H}$. Analogously for the approximation of $\alpha_3$ and $\alpha_4$ only the position of the floor is varied. Normally the optimal height levels of the floor and ceiling are near an observed surface. This means that only the measurements around that height are important for its position. To account for this observation the approximation of the cost function regards only the weight of the voxels that are at most $\varepsilon$ away from the optimal floor and ceiling position. In Fig. 4 the values used for the approximation are visualized. Using this subset of the original function for the approximation, the $\alpha$s can be determined independent of each other. In Fig. 5 a comparison between the original non convex cost function and the convex approximation of two positions one with a very clear and one with a very vague optimum are shown.

By substituting the data cost by its convex approximation the energy functional can also be optimized with the same alternating approach used for the regularization of the labeling. Again the regularization is separated from the data fidelity by introducing two separate height fields $\underline{h}_u$, $\overline{h}_u$ and $\underline{h}_v$, $\overline{h}_v$ that are coupled by the quadratic term $\frac{1}{2\theta}(\underline{h}_u - \underline{h}_v)^2 + \frac{1}{2\theta}(\overline{h}_u - \overline{h}_v)^2$.

$$E_\theta^{\text{Height}} = \int_{\Omega_{\text{inside}}} |\nabla \underline{h}_u| + |\nabla \overline{h}_u| + \frac{1}{2\theta}(\underline{h}_u - \underline{h}_v)^2 + \frac{1}{2\theta}(\overline{h}_u - \overline{h}_v)^2 + \lambda_h C^{\text{conv}}_{x,y}(\overline{h}_v, \underline{h}_v) \mathrm{d}\mathbf{x} \tag{14}$$

The optimization is done by alternating the following two steps that update $\underline{h}_u$, $\overline{h}_u$ and $\underline{h}_v$, $\overline{h}_v$:

- Having $\underline{h}_v$ and $\overline{h}_v$ fixed the energy reduces to
$$\int_{\Omega_{\text{inside}}} |\nabla \underline{h}_u| + \frac{1}{2\theta}(\underline{h}_u - \underline{h}_v)^2 + |\nabla \overline{h}_u| + \frac{1}{2\theta}(\overline{h}_u - \overline{h}_v)^2 \mathrm{d}\mathbf{x}. \tag{15}$$
This is the sum of two ROF energies, which allows to minimize it with the already in the labeling regularization presented gradient descent/reprojection algorithm from [11].
- Having $\underline{h}_u$ and $\overline{h}_u$ fixed the minimum can be calculated directly by taking the gradient of the remaining terms.
$$\int_{\Omega_{\text{inside}}} \frac{1}{2\theta}(\underline{h}_u - \underline{h}_v)^2 + \frac{1}{2\theta}(\overline{h}_u - \overline{h}_v)^2 + \lambda_h C^{\text{conv}}_{x,y}(\overline{h}_v, \underline{h}_v) \mathrm{d}\mathbf{x}. \tag{16}$$
This can be done point-wise because the $h_v$ are not dependent on the spatial context. However because $C^{conv}_{x,y}$ is not differentiable everywhere it leads to a case distinction of 13 cases, where the minimum can lie. In contrast to the update step for the regularized labeling, here no clamping is necessary.

For the calculation of the regularized height levels it is not necessary to have the full voxel grid stored in memory, every z-column can be calculated independently. To optimize the two energy functionals only the minimal cost, its position and the $\alpha$s need to be stored.

### *E. Anisotropic Total Variation*

In the standard definition of the total variation each direction is weighted equally. However for architectural scenes some directions are much more probable then others for example in most of the buildings the walls, floors and ceilings are aligned to three orthogonal directions. In [12] the total variation is extended to a more general model that replaces the $L_2$ norm by a positively 1-homogeneous function $\phi(\cdot)$.

$$E_\phi(u) = \int_\Omega \phi(\nabla u) \mathrm{d}\mathbf{x}, \tag{17}$$

is then defined as the anisotropic total variation. The set

$$W_\phi = \{\mathbf{p} \in \mathbb{R}^N : \langle \mathbf{p}, \mathbf{x} \rangle \leq \phi(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{R}^N\} \tag{18}$$

is called Wulff Shape $W_\phi$ associated to $\phi$. It is connected to the dual vector field $\mathbf{p}$ from the gradient descent/reprojection scheme given in Eq. (9). For the anisotropic total variation it is required that the dual vectors $\mathbf{p}$ are within the Wulff shape

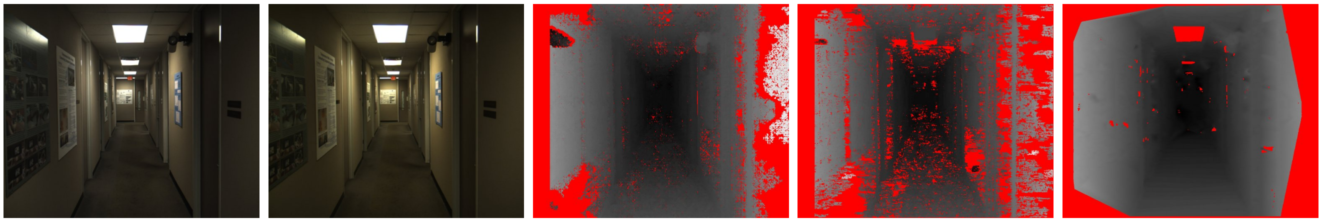

Fig. 6. Input stereo image pair and resulting disparity maps for SGBM, STDP and ELAS. To filter out wrong matches uniqueness ratio tests and speckle filtering are used.

$W_\phi$. By setting $\phi(\cdot)$ to the $L_2$ norm the associated Wulff shape turns out to be the unit ball. It is directly verifiable that the update step in Eq. (9) in fact reprojects $\mathbf{p}$ back to the unit ball.

We replace the standard total variation with the anisotropic model in our energy functionals to align the directions present in the final reconstruction to three orthogonal directions. This is achieved by using the $L_1$-norm or a rotated $L_1$-norm for the function $\phi(\cdot)$. In this case the associated Wulff shape is the unit square or a rotated version of it. For the axis aligned case the reprojection is a simple clamping of the individual components of the dual vector $\mathbf{p}$.

By using more general functions for $\phi(\cdot)$ than the $L_1$-norm also cases where walls are not orthogonal could be handled with this model.

## IV. Implementation and Results

As input data to our fusion we used the results generated with the VSLAM pipelines described in [13], [14]. Their outputs are pairs of keyframes with their camera poses. As a first step the input images are rectified such that it is possible to apply standard stereo algorithms. This rectification process is done with the algorithm described in [15].

For our experiments we used three different stereo matching algorithms. The semi-global block matching (SGBM) algorithm from OpenCV[1], which is a variation of the semi-global matching algorithm from [16], the dynamic programming algorithm on simple tree structures (STDP) from [17] and the efficient large scale stereo (ELAS) algorithm from [18]. In Fig. 6 the results for the different stereo matching algorithms for one stereo image pair are given.

In order to align the occupancy grid with the upright direction we detect the vanishing point associated to this direction in the input images. For this we detect line segments with [19] and optimize the vanishing point with Levenberg-Marquart iterations on the biggest inlier set found with RANSAC.

To compare the proposed heightmap fusion with a reconstruction that does not restrict the possible three dimensional solutions, we also run the TV-Flux fusion from [3] on the same occupancy grids. In Fig. 7 three input images of the hallway junction that is used below for the comparison of the different methods are shown.

Figs. 8 and 9 show the resulting 3D model of a hallway junction when using the TV-Flux fusion. In the case of the $L_2$ norm the junction is reconstructed with rounded parts which is because this part of the scene is unobserved (see Fig. 11) and thus the fusion tries to fill in the missing part with a surface that has low variation. When using the $L_1$ norm this parts are filled in with surfaces that are aligned with the main coordinate axes, which leads to a more reasonable reconstruction for an architectural environment where sharp corners are more likely than rounded surfaces.

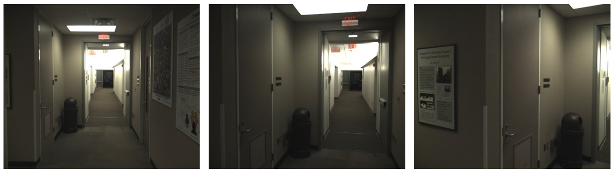

Fig. 7. Input images of the hallway junction

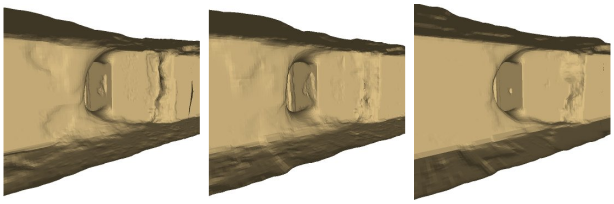

Fig. 8. Fusion results for TV-Flux fusion using the $L_2$-norm with the three different stereo matching algorithms SGBM, STDP and ELAS

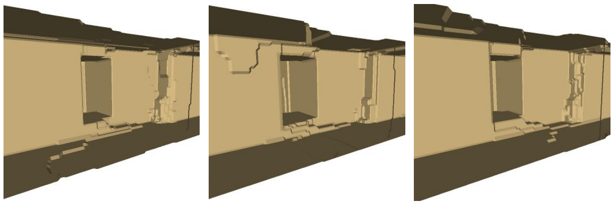

Fig. 9. Fusion results for TV-Flux fusion using the $L_1$-norm with the three different stereo matching algorithms SGBM, STDP and ELAS

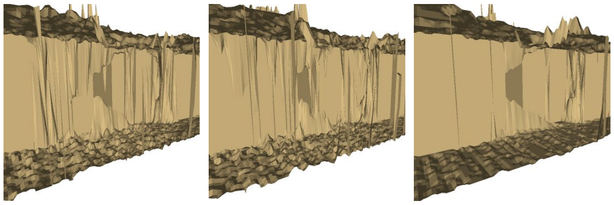

Fig. 10. Fusion results for the two level height map without any regularization for the three different stereo matching algorithms SGBM, STDP and ELAS

[1] http://opencv.willowgarage.com/

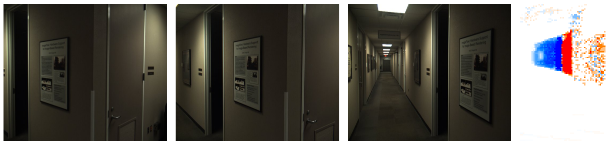

Fig. 11. Input images showing the hallway junction. The floor and ceiling are not observed in some region in the junction. Also the wall on the opposite side is not observed. The last image shows a vertical slice of the occupancy grid. The floor, ceiling and left wall are missing only the right wall is shown in red, which means a positive weight.

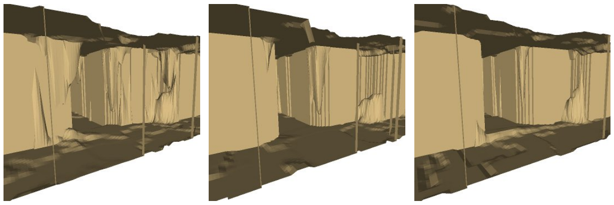

Fig. 12. Our regularized height level fusion results for the two level height map with regularization using the $L_2$-norm for the three different stereo matching algorithms SGBM, STDP and ELAS

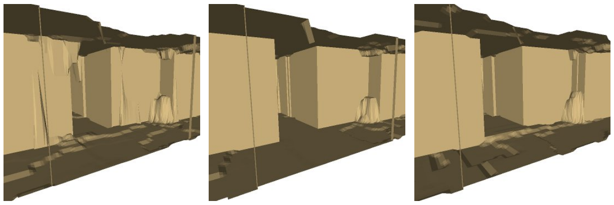

Fig. 13. Our regularized height level fusion results for the two level height map with regularization using the $L_1$-norm for the three different stereo matching algorithms SGBM, STDP and ELAS

In Fig. 10 the resulting 3D model of the junction when directly extracting the height levels without any regularization is shown. The input depth maps contain noise and erroneous depth estimates this leads to a noisy reconstruction. For the unobserved area in the junction the reconstruction aligns with the border of observed and unobserved space, which leads to an unwanted constriction in this area.

By using the regularization a lot of noise can be removed from the reconstruction, Figs. 12 and 13. The $L_2$-norm leads to rounded corners which are introduced in the labeling part of the fusion. By constraining the final reconstruction to surfaces that are aligned with the coordinate axes by using the $L_1$ norm, the sharp corners of the walls are present in the final model. Items standing on the ground like the garbage can (see Figs. 7, 12 and 13) in the hallway corner are still present in the final reconstruction, which is important for robot navigation tasks.

It is noteworthy that by decoupling the walls from the floor and ceiling by the two pass optimization, the constrictions in the junction area are mostly removed from the reconstruction. This works better in case of the anisotropic total variation. Using the standard isotropic total variation it is possible to remove them from the floor and ceiling too. For the walls the data fidelity must be set very low to completely remove them, which results in very rounded corners.

Another benefit of the two pass optimization is that the floor and ceiling position are only calculated where they are really necessary. In some datasets like the hallway (Fig. 14) the reduced domain can be much smaller then the complete height map in this cases the computational benefit can be significant.

In Fig. 15 the reconstruction of a whole big office room with multiple desks and aisles in between is presented. All the relevant details to navigate through the aisles are contained in the final reconstruction. The main errors in the reconstruction are due to the glass walls that surround the reconstructed area, which are not present in the final reconstruction.

## V. Conclusion

In this paper we presented how height levels can be used to reconstruct indoor environments suitable for the navigation of a robot. By introducing a regularization step to the extraction of raw height maps proposed in [7] a lot of noise can be removed. In regions where the floor and ceiling of the scene are unobserved more likely positions are deduced with a total variation prior.

By restricting the space of possible reconstructions to the two level heightmap representation we reduce the dimension of the space on which an energy functional is optimized by one in contrast to other fusion techniques such as the TV-Flux fusion from [3]. This allows to still represent the scene accurate enough for the navigation of a robot, because all the relevant information for the navigation can be expressed in this representation. The floor level can be used as space where the robot moves, it needs to be decided which parts of the floor are regular enough for the movement capabilities of the robot. With additionally having a ceiling level it can be decided if the robot fits into the free space between floor and ceiling. These informations can be used to implement path planning tasks.

By using an anisotropic version of the standard isotropic total variation the surfaces of the final reconstruction can be aligned with the dominant directions in the scene. This allows the reconstruction of sharp edges in architectural environments.

At the moment the reconstruction is done as post-processing after all the images are acquired. To build the map online future work might focus on calculating the maps only in a neighborhood around the current position and move the grid as soon as the robot comes too close to the border. A full map could then be built by merging together all the local maps. Further all the algorithms used are directly implementable parallelized on GPUs, which means that real-time performance could be possible.

## References

[1] P. Merrell, A. Akbarzadeh, L. Wang, J.-M. Frahm, R. Yang, and D. Nistér, "Real-time visibility-based fusion of depth maps," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2007.

[2] C. Zach, T. Pock, and H. Bischof, "A globally optimal algorithm for robust TV-$L^1$ range image integration," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2007.

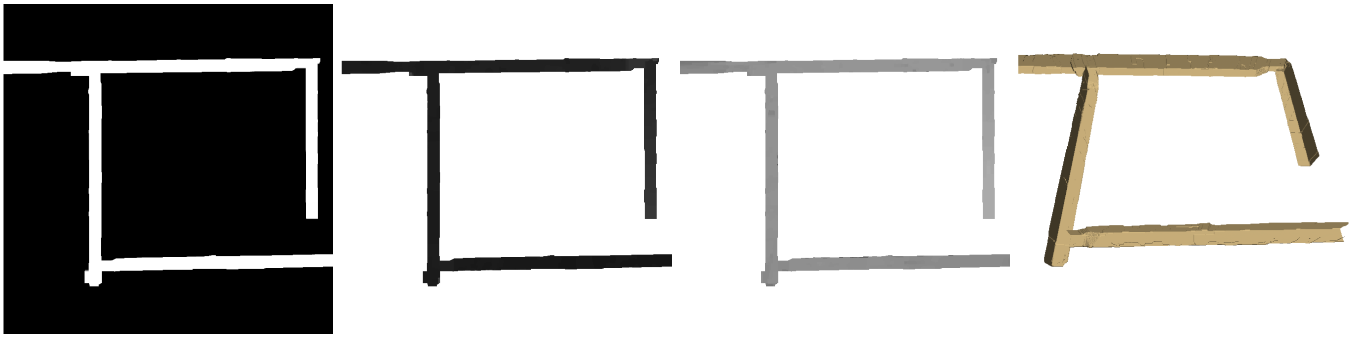

Fig. 14. Result of the hallway dataset: from left to right, labeling, regularized floor and ceiling height level with $L_1$-norm total variation (brighter means higher), 3D model.

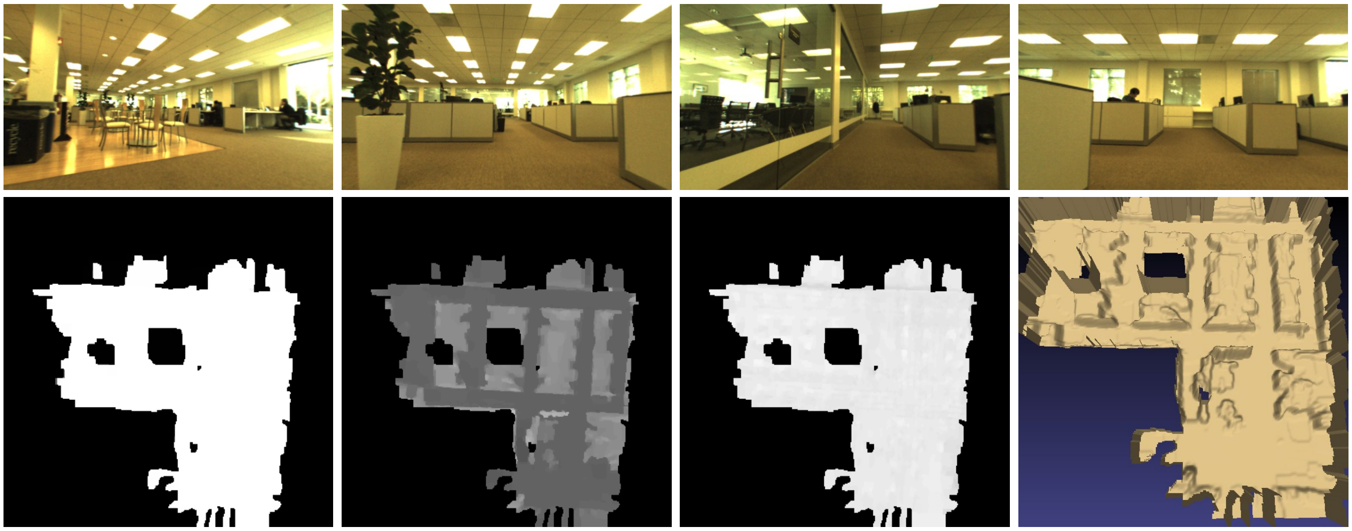

Fig. 15. Top row: 4 out of 518 input images. Bottom row: from left to right, regularized binary labeling, regularized floor and ceiling height map with $L_1$-norm total variation (brighter means higher), 3D model of the whole office data set.

[3] C. Zach, "Fast and high quality fusion of depth maps," in *Proc. International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, 2008.

[4] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Manhattan-world stereo," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[5] Y. Furukawa and J. Ponce, "Patch-based multi-view stereo software," http://grail.cs.washington.edu/software/pmvs/.

[6] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Reconstructing building interiors from images," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2009.

[7] D. Gallup, J.-M. Frahm, and M. Pollefeys, "A heightmap model for efficient 3D reconstruction from street-level video," in *Proc. International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, 2010.

[8] D. Gallup, M. Pollefeys, and J.-M. Frahm, "3D reconstruction using an n-layer heightmap," in *Proc. Annual Symposium of the German Association for Pattern Recognition (DAGM)*, 2010.

[9] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[10] D. Gallup, J.-M. Frahm, and M. Pollefeys, "Variable baseline/resolution stereo," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[11] A. Chambolle, "Total variation minimization and a class of binary MRF models," in *Proc. International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2005.

[12] S. J. Osher and S. Esedoglu, "Decomposition of images by the anisotropic Rudin-Osher-Fatemi model," *Communications on Pure and Applied Mathematics*, vol. 57, no. 12, pp. 1609–1626, 2004.

[13] B. Clipp, J. Lim, J.-M. Frahm, and M. Pollefeys, "Parallel, real-time visual slam," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[14] J. Lim, J.-M. Frahm, and M. Pollefeys, "Online environment mapping," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[15] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision and Applications (MVA)*, vol. 12, no. 1, pp. 16–22, 2000.

[16] H. Hirschmüller, "Stereo processing by semi-global matching and mutual information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, no. 2, pp. 328–341, 2008.

[17] M. Bleyer and M. Gelautz, "Simple but effective tree structures for dynamic programming-based stereo matching," in *Proc. International Conference on Computer Vision Theory and Applications (VISAPP)*, 2008.

[18] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Proc. Asian Conference on Computer Vision (ACCV)*, 2010.

[19] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 32, no. 4, pp. 722–732, 2010.