

Discrete Optimization of Ray Potentials for Semantic 3D Reconstruction

Nikolay Savinov¹, Ľubor Ladický¹, Christian Häne and Marc Pollefeys
ETH Zürich, Switzerland

{nikolay.savinov, lubor.ladicky, christian.haene, marc.pollefeys}@inf.ethz.ch

Abstract

Dense semantic 3D reconstruction is typically formulated as a discrete or continuous problem over label assignments in a voxel grid, combining semantic and depth likelihoods in a Markov Random Field framework. The depth and semantic information is incorporated as a unary potential, smoothed by a pairwise regularizer. However, modelling likelihoods as a unary potential does not model the problem correctly leading to various undesirable visibility artifacts.

We propose to formulate an optimization problem that directly optimizes the reprojection error of the 3D model with respect to the image estimates, which corresponds to the optimization over rays, where the cost function depends on the semantic class and depth of the first occupied voxel along the ray. The 2-label formulation is made feasible by transforming it into a graph-representable form under QPBO relaxation, solvable using graph cut. The multi-label problem is solved by applying α -expansion using the same relaxation in each expansion move. Our method was indeed shown to be feasible in practice, running comparably fast to the competing methods, while not suffering from ray potential approximation artifacts.

1. Introduction

In this paper we are studying the problem of jointly inferring dense 3D geometry and semantic labels from multiple images, formulated as an optimization over rays. The problem of dense 3D reconstruction from images and semantic segmentation is up to date still a hard problem. A particularly powerful approach to the problem of dense 3D reconstruction from images is to pose it as a volumetric labeling problem. The volume is segmented into occupied and free space (the inside and the outside of an object) and the surface is extracted as the boundary in between. Traditionally, the data costs are extracted from the input images either directly by computing matching scores per voxel or by first

computing depth maps and deriving a per pixel unary potential based on the depth maps. In both cases after the image data has been converted to a per voxel unary term the input images are discarded. Unary terms, approximately modelling the likelihood the depth for a given pixel agrees with the estimate, typically encourage voxels in an interval just before the matched 3D point to take the free-space label and voxels in an interval right after the matched 3D point to take the foreground label. However, this assumption does not hold in general. The interval right behind the corner of an object does not necessarily have to belong to foreground. Failures due to this problem lead to blowing up corners, roofs of buildings or thin objects. The problem can be partly fixed by decreasing the length of an interval, however, in that case in presence of noise the matched neighbouring points start to compete against each other. Another problem of unary approximation is that the unary potential does not model, whether the voxel is visible. If there is a hole in the wall and a matched object behind, there is no penalty associated with a data term for closing the hole, which might get filled in by regularization. This problem can also be partially resolved by penalizing foreground for all pixels in front of the matched voxel, however such solution is not robust to outliers. The standard approach is also not suitable for incorporating multiple candidate matches along the viewing ray in the optimization together.

We propose to formulate an optimization problem which measures the data fidelity directly in image space while still having all the benefits of a volumetric representation. The main idea is to use a volumetric representation, but describe the data cost as a potential over rays. Traversing along a ray from the camera center we observe free space until we first hit an occupied voxel of a certain semantic class and we cannot assume anything about the unobserved space behind. The potential we introduce correctly assigns for each ray the cost, based on the depth and semantic class of the first occupied voxel along the ray. The key to make such a formulation feasible is the transformation of the potential into a graph-representable form under QPBO relaxation [2], solvable using graphcut-based methods. Our proposed optimization method can be directly used also for multi-label

¹The authors assert equal contribution and joint first authorship

problems by applying α -expansion [3] with QPBO used to calculate each expansion move. Our method runs comparably fast to the competing methods, while not suffering from ray potentials approximation artifacts.

1.1. Related Work

Generating dense 3D models out of multiple images is a well-studied problem in computer vision. An overview is given in [21]. Posing the problem of dense 3D reconstruction as a volumetric segmentation was first proposed in [7]. The initial formulation does not use any regularization. However, often the data is contaminated by noise and strong assumptions about the smoothness have to be made. Regularizing the surface by penalizing the surface area has been proposed in the discrete graph cut formulation [16, 25] and also as continuous convex optimization formulation [28]. The solution space is not restricted to regular voxel grids. [27] uses a tetrahedronization of the space. The data cost is formulated as a pairwise potential along viewing rays. They put a unary prior for a tetrahedron right behind the initial estimated depth match and a penalty for every cut before it. The photo-consistency of faces is used as a weight for the cost to cut a ray with a given face. A Markov Random Field (MRF) formulation over rays has been proposed to estimate surface geometry and appearance jointly [17, 18]. The energy is formulated as a reprojection error of the reconstructed voxels to the input images, which jointly estimates voxel occupancy and color. The high quality refining of approximate mesh has also been formulated as a reprojection error minimization [26, 8].

The silhouettes of objects in the input image contain important information about the geometry. They constrain the solution, such that for every ray passing through the silhouette there must be at least one occupied voxel, and every ray outside of the silhouette consists of free space voxels only. This constraint has been used in form of a convex relaxation in [6]. [12] proposes an intelligent unary ballooning visibility term based on the consensus from different views. In [23], the silhouettes are handled in a two-stage process, where the initial surface is reprojected into each image and the interior is heuristically corrected using the sets of erroneous pixels, by finding the most photo-consistent voxels along the ray. Recently, also approaches which jointly reason about geometry and semantic labels have been proposed [11]. For volumetric 3D reconstruction in indoor environments a Conditional Random Field (CRF) model was proposed in [13]. It includes higher order potentials over groups of voxels to include priors from 2D object detections and 3D surface detections in the raw input depth data. Furthermore, potentials over rays are used to enforce visibility of only one voxel along a ray.

2. Ray Energy Formulation

We are interested in finding the smooth solution, whose projection into each camera agrees with the depth and semantic observations. Thus, the energy will take the form:

$$E(\mathbf{x}) = \sum_{r \in \mathcal{R}} \psi_r(\mathbf{x}^r) + \sum_{(i,j) \in \mathcal{E}} \psi_p(x_i, x_j), \quad (1)$$

where each $x_i \in \mathcal{L}$ is the voxel variable taking a label from the label set \mathcal{L} with a special label $l_f \in \mathcal{L}$ corresponding to free space; \mathcal{R} is the set of rays, $\psi_r(\cdot)$ is the ray potential over the set of voxels \mathbf{x}^r , \mathcal{E} is the set of local voxel neighbourhoods, and $\psi_p(\cdot)$ is a smoothness enforcing pairwise regularizer. Each ray r of length N_r consists of voxels $x_i^r = x_{r_i}$, where $i \in \{0, 1, \dots, N_r - 1\}$. The ray potential takes the cost depending only on the first non-free space voxel along the ray K^r (if there is any):

$$K^r = \begin{cases} \min(i | x_i^r \neq l_f) & \text{if } \exists x_i^r \neq l_f \\ N_r & \text{otherwise.} \end{cases} \quad (2)$$

The ray potential is defined as:

$$\psi_r(\mathbf{x}^r) = \phi_r(K^r, x_{K^r}^r), \quad (3)$$

where $x_{N_r}^r = l_f$. The costs for each depth and semantic label $\phi_r(\cdot)$ could be arbitrary and typically come from the corresponding semantic and depth classifiers or measurements.

2.1. Optimization of the 2-label Ray Potentials

Let us first consider the two label case. Each variable x_i may take a value from the set $\{0, 1\}$, where $x_i = 0$ corresponds to the occupied voxel and $x_i = 1$ corresponds to free space l_f . The ray potential (3) for arbitrary costs could be non-submodular even for a ray of the length 2 making the 2-label problem NP-hard in general. Thus, we propose a solution using QPBO relaxation [2], where the energy $E(\mathbf{x})$ is transformed to a submodular energy $E(\mathbf{x}, \bar{\mathbf{x}})$ with additional variables $\bar{x}_i = 1 - x_i$, and solved by relaxing these constraints.

In our case, the non-submodular ray potentials are of larger order than 2. To make our problem solvable using graph-cut, our goal is to transform these potentials into a pairwise energy with additional auxiliary variables \mathbf{z} , such that:

$$\psi_r(\mathbf{x}^r) = \min_{\mathbf{z}} \psi_q(\mathbf{x}^r, \bar{\mathbf{x}}^r, \mathbf{z}), \quad (4)$$

where $\psi_q(\cdot)$ is pairwise submodular. Additionally, to keep the problem feasible, our goal is to find a transformation, for which the number of edges in the graph with auxiliary variables grows at most linearly with length of a ray. We achieve this goal using these five steps:

1. Polynomial representation of the ray potential,

2. Transformation into higher order submodular potential using additional variables \bar{x} ,
3. Pairwise graph construction of a higher order submodular potential using auxiliary variables z ,
4. Merging variables [20] to get the linear dependency of the number of edges on length,
5. Transformation into a normal form, symmetric over x and \bar{x} , suitable for QPBO [2].

Next we describe in details each one of these steps. The two-label equivalent of the ray potential takes the form:

$$\psi_r(\mathbf{x}^r) := \begin{cases} \phi_r(\min(i|x_i^r = 0)) & \text{if } \exists x_i^r = 0 \\ \phi_r(N_r) & \text{otherwise,} \end{cases} \quad (5)$$

where $\phi_r(i) := \phi_r(i)$ is the cost taken, if i is the first foreground pixel along the ray, and $\phi_r(N_r)$ is the cost for the whole ray being free space. We would like to transform this potential into the polynomial representation - the sum of products:

$$\psi_r(\mathbf{x}^r) = k^r + \sum_{i=0}^{N_r-1} c_i^r \prod_{j=0}^i x_j^r. \quad (6)$$

Applying to equation (5), we get $\phi_r(K) = k^r + \sum_{i=0}^{K-1} c_i^r$, thus $k^r = \phi_r(0)$ and $c_i^r = \phi_r(i+1) - \phi_r(i)$, $\forall i \in \{0, \dots, N_r-1\}$.

It is well-known that the product $c_i^r \prod_{j=0}^i x_j^r$ is submodular only if $c_i^r \leq 0$ [9]. For $c_i^r > 0$ we can transform the product into submodular function using additional variable $\bar{x}_i^r = 1 - x_i^r$ as:

$$c_i^r \prod_{j=0}^i x_j^r = c_i^r (1 - \bar{x}_i^r) \prod_{j=0}^{i-1} x_j^r = -c_i^r \bar{x}_i^r \prod_{j=0}^{i-1} x_j^r + c_i^r \prod_{j=0}^{i-1} x_j^r. \quad (7)$$

That means, that starting from the last term we can iteratively check if $c_i^r \leq 0$, and if it is not, we transform the term using (7) and update $c_{i-1}^r := c_{i-1}^r + c_i^r$. The transformation algorithm is explained in details in Algorithm 1.

Ignoring the constant term we transform the potential into:

$$\psi_r(\mathbf{x}^r) = \sum_{i=0}^{N_r-1} \left(-a_i^r \prod_{j=0}^i x_j^r - b_i^r \bar{x}_i^r \prod_{j=0}^{i-1} x_j^r \right). \quad (8)$$

Each product in the sum is submodular and graph-representable using one auxiliary variable $z_i \in \{0, 1\}$. The standard pairwise graph constructions [9] for a negative product terms are:

$$-a_i^r \prod_{j=0}^i x_j^r = a_i^r \min_{z_i} \left(-z_i + \sum_{j=0}^i z_i (1 - x_j^r) \right) \quad (9)$$

Algorithm 1 Transformation into submodular potential.

Input: c^r, N_r

Output: a^r, b^r

$i = N_r - 1;$

while $i \geq 0$ **do**

if $c_i^r \leq 0$ **then**

$a_i^r = -c_i^r, b_i^r = 0$

else

$a_i^r = 0, b_i^r = c_i^r$

if $i > 0$ **then** $c_{i-1}^r = c_{i-1}^r + c_i^r$

end if

$i = i - 1$

end while

$$-b_i^r \bar{x}_i^r \prod_{j=0}^{i-1} x_j^r = b_i^r \min_{z'_i} \left(-z'_i + z'_i (1 - \bar{x}_i^r) + \sum_{j=0}^{i-1} z'_i (1 - x_j^r) \right), \quad (10)$$

however, such constructions for each term in the sum would lead to a quadratic number of edges per ray. Instead, we first build a more complex graph constructions with $(i+1)$ auxiliary variables $\mathbf{z}^i \in \{0, 1\}^{i+1}$ and $\mathbf{z}'^i \in \{0, 1\}^{i+1}$ respectively with the foresight, that this will lead to a graph construction with linear growth of the number of edges:

$$-a_i^r \prod_{j=0}^i x_j^r = a_i^r \min_{\mathbf{z}^i} \left(-z_i^i + z_i^i (1 - x_i^r) + \sum_{j=0}^{i-1} (z_{j+1}^i (1 - z_j^i) + z_j^i (1 - x_j^r)) \right) \quad (11)$$

$$-b_i^r \bar{x}_i^r \prod_{j=0}^{i-1} x_j^r = b_i^r \min_{\mathbf{z}'^i} \left(-z'_i{}^i + z'_i{}^i (1 - \bar{x}_i^r) + \sum_{j=0}^{i-1} (z'^i{}_{j+1} (1 - z'^i{}_j) + z'^i{}_j (1 - x_j^r)) \right), \quad (12)$$

where the (one of the) optimal assignment for auxiliary variables is:

$$\forall j \in \{0, \dots, i\} : z_j^i = \prod_{k=0}^j x_k^r \quad (13)$$

$$z'_i{}^i = \bar{x}_i^r \prod_{k=0}^{i-1} x_k^r \quad (14)$$

$$\forall j \in \{0, \dots, i-1\} : z_j^i = \prod_{k=0}^j x_k^r \quad (15)$$

Both equations are structurally the same, so we just prove (11). It can be easily checked, that given the assignments of auxiliary variables (13) every term is always going to be 0 except $-z_i^i$, that is by definition (13) equal to the negative product of all \mathbf{x}^r on the left side of (11). Next we have to show, that if $\mathbf{x}^r \neq \mathbf{1}$, there is no assignment of auxiliary variables leading to a negative cost. Let us try to construct such \mathbf{x}^r . The only term, that could be potentially negative is $-z_i^i$, thus $z_i^i = 1$. However, to keep the whole expression negative, the remaining terms must be 0 and thus:

$$(z_{j+1}^i(1 - z_j^i) = 0) \implies ((z_{j+1}^i = 1) \implies (z_j^i = 1)) \quad (16)$$

$$(z_j^i(1 - x_i^r) = 0) \implies ((z_j^i = 1) \implies (x_i^r = 1)). \quad (17)$$

(16) implies $\mathbf{z}^i = \mathbf{1}$ and then (17) implies $\mathbf{x}^r = \mathbf{1}$. Thus, the assignment (13) is always optimal and (11) holds. There could be more possible optimal assignments, however for the next step we only need this one to exist. The alternative graph construction leads to much more complex graphs than the standard one, however, we can decrease its growth in terms of edges by applying the merging theorem [20]. The theorem states, that if for every assignment of input variables \mathbf{x} there exists at least one assignment of two or more auxiliary variables z_i, \dots, z_j , such that $z_i = \dots = z_j$, these variables can be replaced by a single variable without altering the cost for any assignment of \mathbf{x} . In our graph constructions we can see (13), that the value $z_j^i = \prod_{k=0}^j x_k^r$ is independent on i , and thus:

$$\forall i \in \{0, \dots, N_r - 1\}, \forall j, k \in \{0, \dots, i\} : z_i^j = z_i^k. \quad (18)$$

Furthermore, based on (15):

$$\forall i \in \{0, \dots, N_r - 1\}, \forall j, k \in \{0, \dots, i - 1\} : z_i^j = z_i^k. \quad (19)$$

After removing the unnecessary top index $z_i^j = z_i$ and $z_i^i = z_i'$, and summing up all corresponding weights, the resulting pairwise construction takes the form:

$$\begin{aligned} \psi_r(\mathbf{x}^r) &= \min_{\mathbf{z}, \mathbf{z}'} \left(\sum_{i=0}^{N_r-1} \left(-a_i^r z_i - b_i^r z_i' \right. \right. & (20) \\ &+ f_i^r(1 - z_i)x_i^r + b_i^r(1 - z_i)\bar{x}_i^r \left. \right) \\ &+ \sum_{i=1}^{N_r-2} \left((f_{i+1}^r(1 - z_{i+1})z_i + b_i^r(1 - z_{i+1}')z_i) \right), \end{aligned}$$

where $f_i^r = \sum_{i=0}^{N_r-1} a_j^r + \sum_{i+1}^{N_r-1} b_j^r$. Unlike standard graph construction, this one leads to a number of edges growing at most linearly with the length of the ray. The process of merging is visually depicted in Figure 1.

Finally, the potential is converted into a symmetric nor-

mal form, suitable for QPBO [2] as:

$$\begin{aligned} \psi_r(\mathbf{x}^r) &= \frac{1}{2} \left(\min_{\mathbf{z}} \psi^r(\mathbf{x}, \bar{\mathbf{x}}, \mathbf{z}) \right. \\ &+ \left. \min_{\bar{\mathbf{z}}} \psi^r(\mathbf{1} - \bar{\mathbf{x}}, \mathbf{1} - \mathbf{x}, \mathbf{1} - \bar{\mathbf{z}}) \right), \quad (21) \end{aligned}$$

leading to the symmetric quadratic representation:

$$\begin{aligned} \psi_r(\mathbf{x}^r) &= \frac{1}{2} \min_{\mathbf{z}, \mathbf{z}', \bar{\mathbf{z}}, \bar{\mathbf{z}'}} \left(\sum_{i=0}^{N_r-1} \left(-a_i^r z_i - b_i^r z_i' \right. \right. & (22) \\ &+ f_i^r(1 - z_i)x_i^r + b_i^r(1 - z_i)\bar{x}_i^r \left. \right) \\ &+ \sum_{i=1}^{N_r-2} \left(f_{i+1}^r(1 - z_{i+1})z_i + b_i^r(1 - z_{i+1}')z_i \right) \\ &+ \sum_{i=0}^{N_r-1} \left(-a_i^r(1 - \bar{z}_i) - b_i^r(1 - \bar{z}'_i) \right) \\ &+ f_i^r \bar{z}_i(1 - \bar{x}_i^r) + b_i^r \bar{z}_i(1 - x_i^r) \left. \right) \\ &+ \sum_{i=1}^{N_r-2} \left(f_{i+1}^r \bar{z}_{i+1}(1 - \bar{z}_i) + b_i^r \bar{z}'_{i+1}(1 - \bar{z}_i) \right). \end{aligned}$$

To remove fractions, we multiply all potentials by a factor of 2. The resulting graph construction is shown in Figure 2. Our method inherits the properties of the standard QPBO for pairwise graphs [2]. The set of variables, for which $x_i = 1 - \bar{x}_i$, is the part of the globally optimal solution. The remaining variables can be labelled using Iterated Conditional Modes (ICM) [1] iteratively per variable.

2.2. Optimization of the Multi-label Ray Potentials

To solve the multi-label case, we adapt the α -expansion algorithm. The moves proposed by α -expansion [3], are encoded by a binary transformation vector \mathbf{t} , which encodes, how should the variable x_i change after the move. There are two distinct cases - expansion move of the free space label l_f and expansion move of any other foreground label $l \neq l_f$.

For the free space label we define the transformation function as:

$$T_{l_f}(x_i, t_i) = \begin{cases} l_f & \text{if } t_i = 1 \\ x_i & \text{if } t_i = 0 \end{cases} \quad (23)$$

Next we have to find the 2-label projection of multi-label ray potential (3) under this transformation function $T_{l_f}(\cdot)$. Let $s(r)$ be the ordered set of variables in a ray r , which before the move do not take the label l_f : $s(r) = \{r_i | x_i^r \neq l_f\}$. After the move, the first occupied voxel of r will be the first occupied voxel of $s(r)$, and thus the ray potential (3)

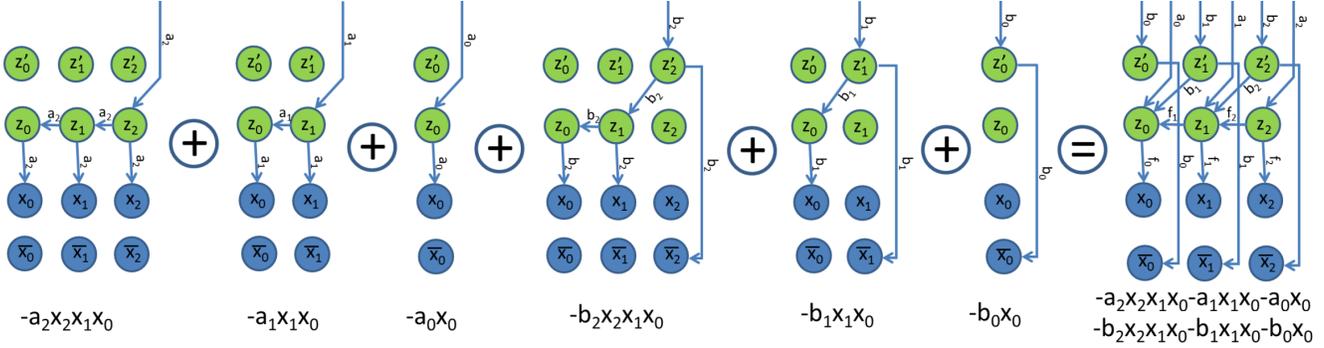


Figure 1. The process of merging for ray of the length 3. Because the optimal assignment of auxiliary variables is the same for every construction, the variables can be merged and thus we can build the final graph just by summing up corresponding edges in each one of these 6 graphs. The weights in the final graph will be $f_0 = a_2 + a_1 + a_0 + b_2 + b_1$, $f_1 = a_2 + a_1 + b_2$ and $f_2 = a_2$.

projects into 2 labels as:

$$\psi_r(\mathbf{t}^r) = \begin{cases} \phi_r(K(\mathbf{t}), x_{K(\mathbf{t})}^{s(r)}) & \text{if } \exists t_i^{s(r)} = 0 \\ \phi_r(N_r, l_f) & \text{otherwise,} \end{cases} \quad (24)$$

where $K(\mathbf{t}) = \min(i \in s(r) | t_i^{s(r)} = 0)$. The projection takes the form of (5), and thus can be solved using the graph-construction (22).

For $l \neq l_f$ we define the transformation function as:

$$T_l(x_i, t_i) = \begin{cases} l & \text{if } t_i = 0 \\ x_i & \text{if } t_i = 1 \end{cases} \quad (25)$$

After the move the first non-free space label could be only in front of the first non-free space label of the current solution. Let $s'(r)$ be the ordered set of variables in front of current first occupied voxel $K_r : s'(r) = \{r_i | i \leq K_r\}$ (including K_r if K_r exists). The ray potential will project into:

$$\psi_r(\mathbf{t}^r) = \begin{cases} \phi_r(K(\mathbf{t}), l) & \text{if } \exists t_i^{s'(r)} = 0 \\ \phi_r(N_{s'(r)}, x_{K_r}^{s'(r)}) & \text{otherwise.} \end{cases} \quad (26)$$

The projection also takes the form of (5) and can be solved using (22).

3. Implementation Details

As an input our method uses semantic likelihoods, predicted by a pixel-wise context-based classifier from [15], and depth likelihoods obtained by a plane sweep stereo matching algorithm using zero-mean normalized cross-correlation. In general the ray costs for a label l take the form:

$$\phi_r(i, l) = \left((\lambda_{sem} C(l) + \lambda_{dep} C(d(i))) \right) d(i)^2, \quad (27)$$

where $C(l)$ is the semantic cost for label l (or sky for free space label l_f), $C(d(i))$ is the cost for depth $d(i)$ of a pixel

i , and λ_{sem} and λ_{dep} are the weights of both domains. Because each pixel corresponds to patch in a volume, which scales quadratically with distance, thus the costs have to be weighted by a factor of $d(i)^2$ to keep constant ratio between ray potentials and regularization terms. In theory, our method allows for optimization over the whole depth cost volume, however this would require too much memory to store. In practice, a few top matches (in our case we used at most 3) for each pixel contain all the relevant information and all remaining scores are typically random noise. In our experiments the costs for given depth took the form:

$$C(d(i)) = \begin{cases} w_n (-1 + \frac{|d(i) - d_{top}^n|}{\Delta}) & \text{if } |d(i) - d_{top}^n| \leq \Delta \\ 0 & \text{otherwise.} \end{cases} \quad (28)$$

where w_n is the weight of the n -th match with the depth d_{top}^n , calculated as a ratio of confidence scores of the n -th match with respect to the best one. As a smoothness enforcing pairwise potential we used the discrete approximation [14] of the continuous anisotropic pairwise regularizer from [11]. To deal with the high resolution of the 3D we use a coarse-to-fine approximation with 3 subdivision stages. As a graph-cut solver, we eventually used the IBFS algorithm [10] algorithm, which was typically 5–50× faster than commonly used Boykov-Kolmogorov [3] algorithm optimized for lattice graphs. The run-time depends not only on the resolution, but also on the number of rays sampled. For example for the Castle dataset with 50 million voxels and 150 million rays the optimization took approximately 40 minutes on 48 CPU cores, which is comparable or faster than other methods [11]. To extract mesh out of voxelized solution we used Marching cubes algorithm [19]. Final models were smoothed using Laplacian smoothing to reduce discretization artifacts.

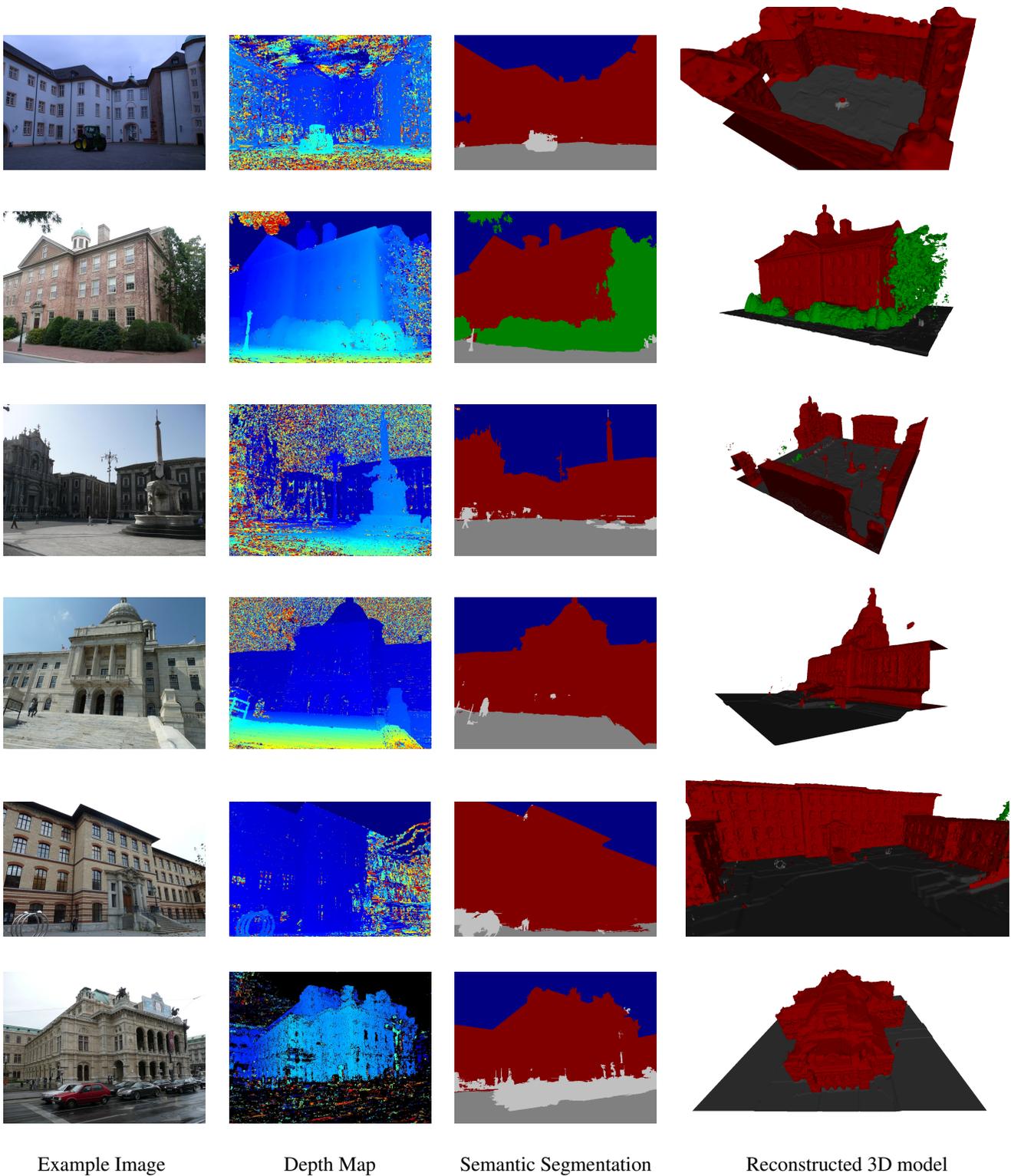
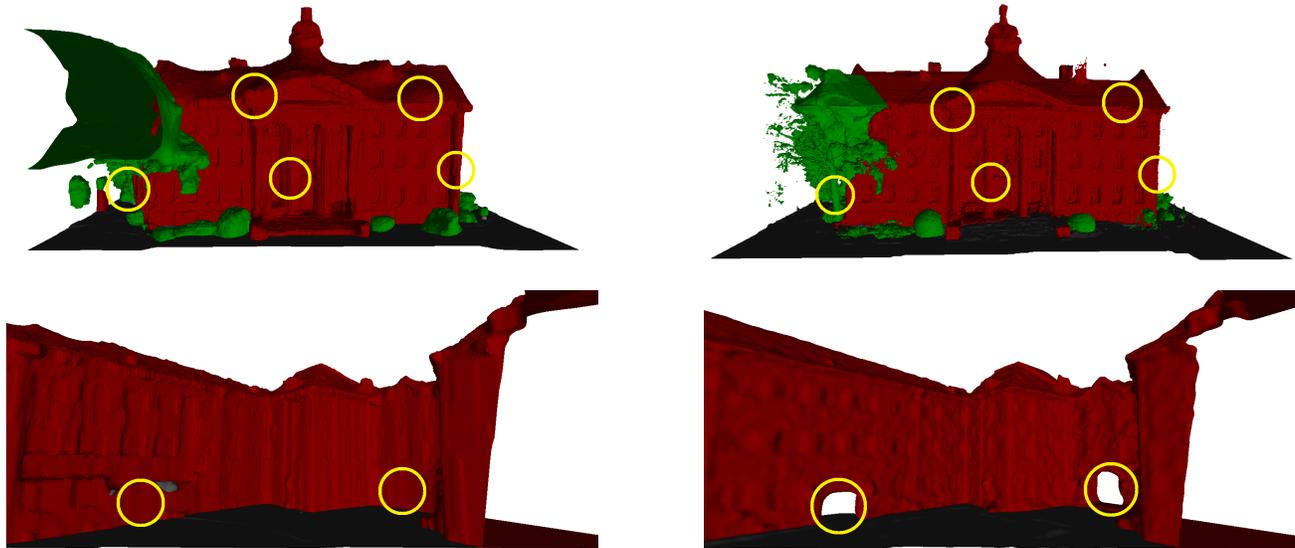


Figure 3. Qualitative results on 6 data sets (from top to bottom): Castle [24], South Building [11], Catania [11], Providence [11], CAB [5] and Opera [5]. Our method successfully reconstructed challenging 3D data with high level of detail. Minor errors in the reconstructions were caused by the combination of errors of the semantic classifier, insufficient amount data from certain viewpoints or errors in the depth prediction for smooth texture-less surfaces.



Joint Volumetric Fusion [11]

Our Method

Figure 4. Comparison of method with the state-of-the-art dense volumetric reconstruction [11] on South Building [11] and Castle datasets [24]. Incorporation of true semantic and depth ray likelihoods in the optimization framework led to the corrections of systematic reconstruction artifacts, due to the approximations in the formulation of the problem. As can be seen in the South Building reconstruction - state-of-the-art methods, enforcing estimated label in the range behind the depth match, typically causes thickening of the thin structures - columns, tree branches or building corners. Furthermore, as seen in the reconstruction of the Castle, the regularization, minimizing the surface, often closes openings in the flat walls, such as arches in this case, because there is no cost associated. In our framework, such solution has a higher cost, because it disregards otherwise valid matches behind the arch.

- [14] V. Kolmogorov and Y. Boykov. What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In *International Conference on Computer Vision*, 2005.
- [15] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr. Associative hierarchical CRFs for object class image segmentation. In *International Conference on Computer Vision*, 2009.
- [16] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *Conference on Computer Vision and Pattern Recognition*, 2007.
- [17] S. Liu and D. B. Cooper. Ray markov random fields for image-based 3d modeling: Model and efficient inference. In *Conference on Computer Vision and Pattern Recognition*, 2010.
- [18] S. Liu and D. B. Cooper. A complete statistical inverse ray tracing approach to multi-view stereo. In *Conference on Computer Vision and Pattern Recognition*, 2011.
- [19] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, 1987.
- [20] S. Ramalingam, C. Russell, L. Ladicky, and P. H. Torr. Efficient minimization of higher order submodular functions using monotonic boolean functions. *Arxiv preprint arXiv:1109.2304*, 2011.
- [21] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Conference on Computer Vision and Pattern Recognition*, 2006.
- [22] J. Shotton, J. Winn, C. Rother, and A. Criminisi. *Texton-Boost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, 2006.
- [23] S. N. Sinha, P. Mordohai, and M. Pollefeys. Multi-view stereo via graph cuts on the dual of an adaptive tetrahedral mesh. In *International Conference on Computer Vision*, 2007.
- [24] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Conference on Computer Vision and Pattern Recognition*, 2008.
- [25] G. Vogiatzis, P. H. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Conference on Computer Vision and Pattern Recognition*, 2005.
- [26] H.-H. Vu, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *Conference on Computer Vision and Pattern Recognition*, 2009.
- [27] H.-H. Vu, P. Labatut, J.-P. Pons, and R. Keriven. High accuracy and visibility-consistent dense multiview stereo. *Transactions on Pattern Analysis and Machine Intelligence*, 2012.
- [28] C. Zach. Fast and high quality fusion of depth maps. In *3D Data Processing, Visualization and Transmission*, 2008.