Stereo Reconstruction of Building Interiors with a Vertical Structure Prior

Bernhard Zeisl, Christopher Zach, Marc Pollefeys
Computer Vision and Geometry Group
ETH Zürich, Switzerland
{*bernhard.zeisl, christopher.zach, marc.pollefeys*}@*inf.ethz.ch*

***Abstract*—Image-based computation of a 3D map for an indoor environment is a very challenging task, but also a useful step for vision-based navigation and path planning for autonomous systems, and for efficient visualization of interior spaces. Since computational stereo is a highly ill-posed problem for the typically weakly textured, specular, and even sometimes transparent indoor environments, one has to incorporate very strong prior assumptions on the observed geometry. A natural assumption for building interiors is that open space is bounded (i) by parallel ground and ceiling planes, and (ii) by vertical (not necessarily orthogonal) wall elements. We employ this assumption as a strong prior in dense depth estimation from stereo images. The additional assumption of smooth vertical elements allows our approach to fill in plausible extensions of e.g. walls in case of (non-vertical) occlusions. It is also possible to explicitly detect non-vertical regions in the images, and to revert to more general stereo methods only in those areas. We demonstrate our method on several challenging stereo images of office environments.**

***Keywords*-computational stereo, 3D reconstruction, dynamic programming**

I. Introduction

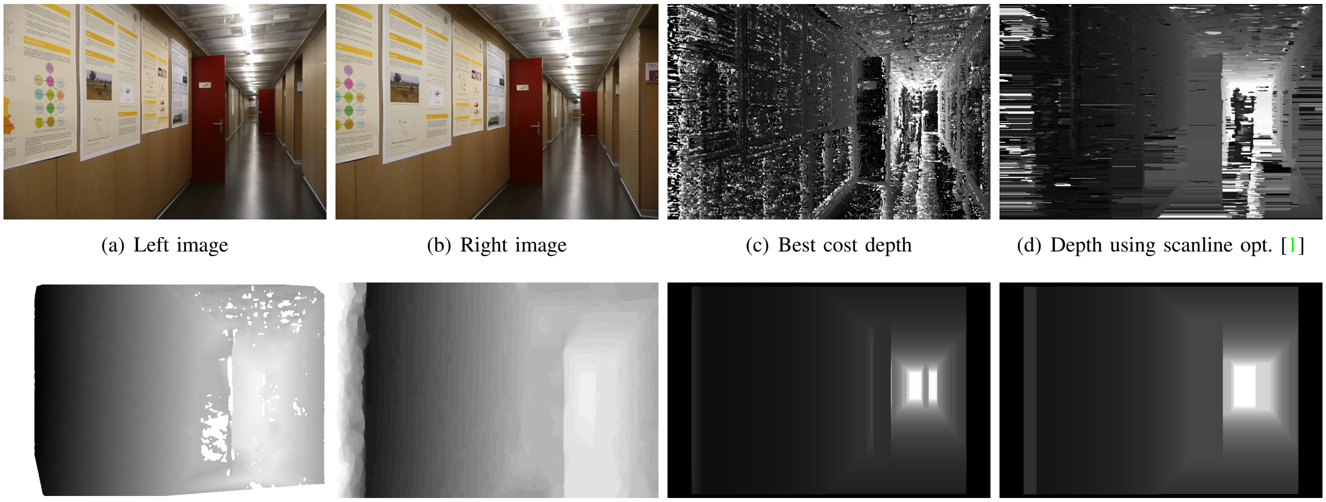
Having a 3D representation of the surrounding environment is essential for robot navigation and path planning. Obtaining such a representation becomes very challenging, if active sensors are not available and the 3D virtual model has to be generated solely from image data. In particular, man-made environments, which are typically comprised of few visually salient objects and tend to have only weak textures, are very demanding for automated image-based 3D modeling. Office-like indoor environments often also contain specular or even transparent objects violating the Lambertian surface assumption, thus making image-based reconstruction even harder. Sophisticated methods for computational stereo can overcome some of these difficulties, but those methods come at a high computation cost and are therefore less suitable e.g. for obstacle avoidance and online navigation of autonomous systems.

In order to obtain a sufficiently accurate, but computationally cheap 3D map of the environment, suitable prior knowledge on the encountered surrounding is necessary. In man-made environments the Manhattan-world assumption (i.e. that the major surfaces are parallel with either the ground plane, or with one of two orthogonal planes) is recently utilized as a strong prior in several approaches for image-based modeling. We replace the Manhattan-world assumption by a related, but somewhat different prior for indoor environments: the open/maneuverable space is bounded by parallel ground and ceiling planes, and by purely vertical structures (i.e. mostly walls). Further, vertical elements are assumed to be (piecewise) smooth in 3D. Under this assumption our method is able to "hallucinate" the most probable vertical structure whenever it is obscured by non-vertical elements (e.g. people or furniture), or alternatively it can detect non-vertical objects and insert depth measurements from a different source (e.g. local stereo).

We illustrate the difficulties of obtaining a meaningful depth map in an indoor environment in Figure 1, where a stereo pair depicting a hallway is shown (Figures 1(a) and (b)). Columns in the images already correspond to vertical structures in 3D. The floor and the ceiling have significant view-dependent highlights, and the scene is partially weakly textured. These properties result in poor depth maps using local (best cost) and scanline optimization (Figures 1(c) and (d)). Incorporating a strong piece-wise planarity prior (Figure 1(e)) or even global optimization for stereo (Figure 1(f)) returns visually appealing disparity maps, but both methods have major difficulties in the ground region (due to the specularities). Explicit incorporation of a vertical world assumption significantly stabilizes depth estimation with and even without vertical smoothness (Figures 1(g) and (h)).

In this work we explore the utility of the vertical structure prior for challenging indoor environments. Due to the substantial simplification of the overall problem we propose to employ efficient dynamic programming to obtain a depth map suitable e.g. for autonomous system navigation. Further, we introduce an optional model selection stage to detect image columns violating the vertical assumption. All processing steps run at interactive frame rates.

In the following Section II we cover related work, whereas Section III explains the underlying idea of a vertical structure and needed preprocessing steps. Next, the utilization of vertical structures in the algorithm is outlined in Section IV, followed by the incorporation of smoothness assumptions via dynamic programming in Section V. Experiments, illustrating the effectiveness of our approach, are presented in Section VI. Finally Section VII concludes with a summarizing and prospective discussion.



(a) Left image

(b) Right image

(c) Best cost depth

(d) Depth using scanline opt. [1]

(e) Depth from libELAS [2]

(f) Depth using global opt. [3]

(g) Depth using vertical aggregation

(h) Our result (with DP)

Figure 1. The difficulty of dense stereo in indoor environments. Note the outputs in (e) and (f) are disparity rather than depth maps, explaining the differences in visual appearance.

II. Related Work

Reconstructing man-made environments from images typically requires strong assumptions, i.e. scene priors, to be able to handle texture-less regions successfully. In particular this applies to indoor environments where weakly textured, homogeneous surfaces (e.g. uniform walls) are dominant in the image. Consequently, several strong priors for reconstructing urban environments and building interiors are proposed in the literature.

Man-made outdoor environments are usually composed of mainly piece-wise planar surfaces. This strong assumption can be incorporated at different steps in the image-based reconstruction process: first, computation of the matching costs between images can be improved by considering several surface orientations (derived e.g. from dominant vanishing directions or from a sparse 3D point model) [4], [5]. Further, the robustness of depth map extraction and the efficiency of 3D model representation can be significantly enhanced [6], [7]. A different, but usually even stronger model for outdoor urban environments assumes purely vertical facades emerging from a ground plane [8]. The corresponding computation and representation of depth maps is extremely efficient: after image alignement with the vertical direction only one depth value per image column needs to be determined and stored in the depth map. Further, depth map computation is very robust, since the matching costs along a complete image column can be (robustly) fused to determine the single required depth value. Due to these advantages our work is heavily inspired by [8] .

Reconstructing indoor environments, e.g. office spaces or corridors, from images poses even a more challenging task, since texture-less or only weakly textured surfaces are predominant. In many cases line structures corresponding to (orthogonal) vanishing directions allow the inference of simple planar, Manhattan-like models from single images [9], [10]. Unfortunately, these methods are not suitable for (near) real-time applications due to the expensive inference stage to determine the most likely 3D configuration. Fusing several depth maps, generated under the Manhattan-world assumption, can give impressive results [11]. Since our application is targeted towards robot navigation and path planning, such a high-quality approach is not feasible because of run-time constraints, and the potential lack of required redundancy in the captured image data (e.g. if a stereo setup is utilized).

In order to handle weakly textured regions, dense correspondence methods typically utilize some prior model on the resulting depth map. Usually, this prior is very generic and formulated in terms of pairwise (sometimes higher order) clique potentials in a Markov random field favoring small depth discontinuities (see e.g [1] for a review of computational stereo methods). The assumption of piecewise planarity of the imaged environment can be explicitly incorporated by assigning locally planar depth hypotheses to image regions induced by super-pixel segmentation (e.g. [12], [13], [14]). A fast stereo method strongly using the piecewise planar assumption was recently proposed in [2]. This approach first determines a sparse set of very confident correspondences, and uses the induced Delaunay triangulated surface model as strong prior for the generation of a complete depth map. [15] extends the piecewise planar model and explicitly introduces an additional label for non-planar surfaces. Images are segmented into planar and non-planar regions by means of photoconsistency and learned appearance, and finally non-planar regions are modeled with a standard stereo approach.

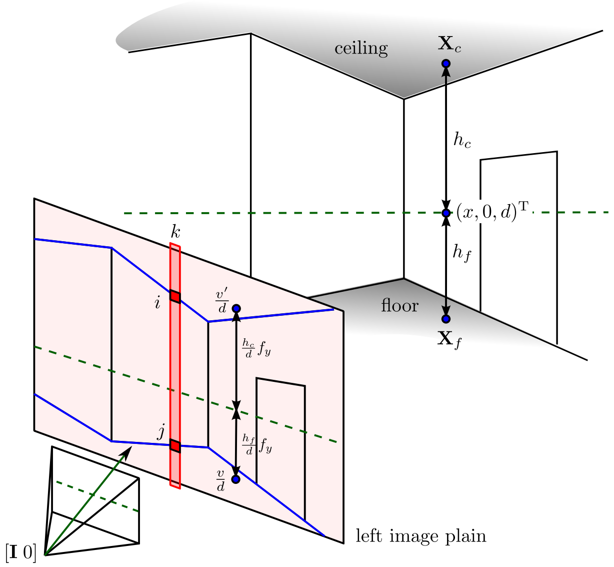


Figure 2. Layout of a vertical structure in 3D and its representation in the image domain. Corresponding ceiling and floor points $(\mathbf{X}_c, \mathbf{X}_f)$ are coupled via their common depth d, resulting in a general coupling of image points v', v and finally of boundary points i, j (describing a vertical structure).

III. Preprocessing Steps

We want to start our explanation by motivating for the layout of a vertical structure in the scene and its representation in the image domain. According to the illustration in Figure 2 let us assume for the left camera that (i) the optical axis is parallel to the ground plane, (ii) it has extrinsic parameters $[\mathbf{I}\ 0]$ and thus defines the reference coordinate system, (iii) vertical structures in the scene posses a vertical layout in the image domain, and (iv) that the hight of ceiling and ground plane are known. Sections III-A, III-B will explain how we can guarantee these requirements, but first it is important to note that under these assumptions the intersection points of a purely vertical element (which can be seen as an upright line in 3D) with the ground and ceiling plane share the same depth.

Given heights h_c and h_f for ceiling and floor plane with plane normal $\mathbf{e}_y = (0, 1, 0)^T$, two corresponding points are $\mathbf{X}_f = (x, h_f, d)^T$ and $\mathbf{X}_c = (x, h_c, d)^T$. Thus, we obtain for the respective image positions

$$(u, v, d)^{\mathrm{T}} = \mathbf{K}\mathbf{X}_f \quad \text{and} \quad (u', v', d)^{\mathrm{T}} = \mathbf{K}\mathbf{X}_c,$$
$$\text{with} \quad \mathbf{K} = \begin{pmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{pmatrix}$$

describing the camera intrinsics. Since $\mathbf{X}_c = \mathbf{X}_f + (0, h_c - h_f, 0)^T$ we are only interested in the parameter change in y direction. Given $\mathbf{X}_f/d = \mathbf{K}^{-1}(u/d, v/d, 1)^T$ and $\mathbf{X}_c/d = \mathbf{K}^{-1}(u'/d, v'/d, 1)^T$ the respective projections on $\mathbf{e}_y$ are

$$\mathbf{X}_f/d \cdot \mathbf{e}_y = h_f/d = f_y^{-1} v/d - f_y^{-1} p_y$$
$$\mathbf{X}_c/d \cdot \mathbf{e}_y = h_c/d = f_y^{-1} v'/d - f_y^{-1} p_y.$$

Due to the upper triangle structure of $\mathbf{K}^{-1}$ the above relation is independent from the horizontal image locations u/d and u'/d. In the remainder of the paper we will denote indices

$$i = \frac{v'}{d} = \frac{h_c}{d} f_y + p_y \quad \text{and} \quad j = \frac{v}{d} = \frac{h_f}{d} f_y + p_y \tag{1}$$

for ceiling and floor boundary, respectively. These quantities are dependent on the depth d and define the mapping

$$i = \frac{h_c}{h_f} j + \left(1 - \frac{h_c}{h_f}\right) p_y. \tag{2}$$

Thus, with known camera intrinsics and heights h_c, h_f *either* d, i or j allows to fully specify a vertical structure.

A. Image Alignment with Vertical Direction

This simple relation between image projections of corresponding points on the floor and the ceiling plane only holds for cameras aligned with the vertical direction. This constraint is not fulfilled a priori, but can be met by warping images by an appropriate homography. We utilize the vertical vanishing point to determine the upright direction by first detecting edges, followed by a line growing and clustering step, and finally by rejecting outliers via a RANSAC approach [16].

Vertical scene structures match with image columns if the corresponding vertical vanishing point $\mathbf{v}_v$ lies at infinity, i.e. at $(0, 1, 0)^T$. Let $\mathbf{r}_v = \mathbf{K}^{-1}\mathbf{v}_v$ be the ray in the vertical vanishing direction, then the needed homography is represented via a camera coordinate system rotation, aligning $\mathbf{e}_y = (0, 1, 0)^T$ with $\mathbf{r}_v$. The remaining axes of the new coordinate system are chosen to be orthogonal to $\mathbf{r}_v$. Assuming the original reference camera system was the identity matrix $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$, the new coordinate system coincides with the 3D rotation

$$\mathbf{R}_v = \begin{pmatrix} \mathbf{r}_v \times \mathbf{e}_3 \\ \mathbf{r}_v \\ \mathbf{r}_v \times \mathbf{e}_3 \times \mathbf{r}_v \end{pmatrix},$$

and the homography in the image domain is given via

$$\mathbf{H}_v = \mathbf{K}\mathbf{R}_v\mathbf{K}^{-1}.$$

The pose of the second stereo image $\mathbf{P} = [\mathbf{R}\ \ \mathbf{t}]$ changes accordingly to $\mathbf{P}_v = \left[\mathbf{R}_v^{-1}\mathbf{R}\ \ \mathbf{t}\right]$.

B. Identification of Floor and Ceiling plane

Knowledge of the ceiling and floor heights h_f, h_c is important, since they define the relation between hypothesis depths and vertical structures. We take a data driven approach, where the mapping of boundary points, i.e. the ratio h_c/h_f of Eq. (2), is determined by robustly voting for corresponding points on edges above and below horizon (in one image) [10], thereby relying on strong edges at structural boundaries. Next we fit vertical structures (see following Section IV) with random boundary pairs (i, j) in the matching cost volume and determine the depth with minimum vertical cost. Implicitly we retrieve corresponding ceiling and floor heights and in this way vote for the most likely ground and ceiling configuration. Alternatively, in robotics it is likely that the height of the camera(s) above ground is fixed and known. A sampling of ground contact points similar to [8] will give a stable estimate of the ground plane over time. A line based reconstruction scheme (inspired by [17]) may also be utilized for the estimation of corresponding plane heights. It is applicable if at least two boundary points lie on the same edge.

C. Calibration and Stereo Image Matching

Calculation of matching costs for various depths requires the knowledge of camera poses and intrinsics. In our target setting in robotics we can assume that either the robot is equipped with a calibrated stereo camera pair, or that structure from motion/visual SLAM is applied for self localization. Therefore, we can assume camera poses and intrinsics are given.

Matching costs for different depth hypotheses may be calculated along scan lines for a rectified image pair. In general, aligning the cameras with vertical elements in the images usually destroys the rectified setup, hence we employ a plane sweep approach [4] to calculate the matching cost volume. Sweeping directions are set along the optical axis (i.e. fronto-parallel and thus aligned with column-wise vertical structures) and in direction of the vertical axis to match ceiling and ground plane. Planes are chosen such that depth hypothesis exhibit a linear spacing in the disparity domain. In the following the resulting cost volume is denoted by $\mathbf{C}(x, y, \mathbf{p})$, where $\mathbf{p} = (\mathbf{e}, d)$ encodes the sweeping direction $\mathbf{e}$ and distance (depth) d from the reference camera center $[\mathbf{I}\ 0]$.

IV. Vertical Structure Hypothesis

Assuming a vertical structure along an image column k, its start point i, end point j and depth d can be used synonymously for parametrization as was described in Section III. In this way all possible depth hypotheses d relate to index pairs (i, j), i.e. $d \mapsto (i, j)$ according to Eq. (1), and encode the cost table D_k for column k.

The cost for an assumed vertical structure in image column k is given by the sum over individual matching costs at its depth hypothesis d via

$$D_k^V(d) = \sum_{r=i}^{j} \mathbf{C}(k, r, (\mathbf{e}_z, d)).$$

If boundary points (i, j) lie within the image, the support of accumulated matching costs along the fixed ceiling and floor plane can be facilitated with

$$D_k^C(d) = \sum_{r=0}^{i-1} \mathbf{C}(k, r, (\mathbf{e}_y, h_c))$$

$$D_k^F(d) = \sum_{r=j+1}^{m-1} \mathbf{C}(k, r, (\mathbf{e}_y, h_f)).$$

For the calculation of D_k^C and D_k^F we make use of the cumulative structure along the ceiling and floor plane, i.e. we calculate running sums of matching costs. For D_k^V cost accumulation is not possible, since each depth d in the cost volume is just considered only once.

Consequently the combined cost D_k for a vertical structure at depth d and image column k is described by the aggregation of previous three terms by

$$D_k(d) = D_k^C(d) + D_k^V(d) + D_k^F(d).$$

The most suitable combination of vertical structures simplifies to solving for the column-wise minimum over possible depths

$$d_k^* = \arg\min_d D_k(d) \quad \forall k \in \{0, \ldots, m-1\} \tag{3}$$

Figures 5(c) and 6(c) illustrate the depth maps obtained by this local optimization.

V. Optimization via Dynamic Programming

Given the best cost solution obtained via Eq. (3) one can observe undesired depth discontinuities, especially at locations where the solution is ambiguous. In this section we will present how smoothness between neighboring vertical structures can be enforced and how the optimization problem can be solved efficiently via dynamic programming.

In general dynamic programming guarantees to find the global optimum for an energy function like

$$E = D_0(l_0) + \sum_{k=1}^{n-1} \{D_k(l_k) + V(l_k, l_{k-1})\},$$

with labels $l_k \in \mathcal{L}$, unary terms $D_k(l_k)$ and binary terms $V(l_k, l_{k-1})$. In the simplest setting $\mathcal{L}$ contains the set of possible depths and we have $l_k = d_k$. Then $D_k(d_k)$ is the cost for a vertical structure at depth d_k as computed in Section IV. The resulting smoothness term $V(d_k, d_{k-1})$ constitutes a penalty for large label changes, i.e. it penalizes deviations in depth. It could be spatially varying with location k as well, but we did not make use of this generalization. Note that smoothness along columns is already encoded in the data terms, because the vertical structure prior only allows one single depth. In our setting we use a linear cost model for $V(\cdot, \cdot)$ with slope λ_d and truncated by t to allow for large depth changes, if the data term indicates so. The penalty for a depth change reads as

$$V(d_k, d_{k-1}) = \lambda_d \min\left(|d_k - d_{k-1}|, t\right). \tag{4}$$

The dynamic programming algorithm is traversing over image columns, left to right, and accumulates costs up to the current position. In column k for label d_k it searches over all previous depths d_{k-1} and selects the one with minimum accumulated costs and regularization penalties. The related dynamic programming cost table, denoted as $C_k(d)$, is written as (for better readability we will drop the index from labels and depths in the following)

$$\begin{aligned} C_k(d) &= D_k(d) + \min_{\hat{d}} \left\{ C_{k-1}(\hat{d}) + V(d, \hat{d}) \right\} \qquad (5) \\ &= D_k(d) + \min_{\hat{d}} \left\{ C_{k-1}(\hat{d}) + \lambda_d \min\left(|d - \hat{d}|, t \right) \right\}, \end{aligned}$$

with the initialization $C_0(d) = D_0(d)$. We use the fast min-convolution [18] to update $C_k(d)$ for all depths d in linear time. The optimal solution is found by backtracking over C_k for $k = m-1 \ldots 0$.

A. First Extension: Slope based Smoothness Term

The regularization term in Eq. (4) prefers structures with constant depths, which is not always suitable for the often observed piecewise linear assembly of vertical structures (recall Figure 1). Directly adding a curvature regularization as proposed in [19] via ternary cliques is expensive due to the quadratic complexity in the number of labels. The alternative is to extend the labels by a slope value, hence a label $l_k = (d_k, s_k)$ consists of a depth and a respective local slope value. Thus, the binary cliques for the smoothness are sufficient. We obtain a speed-up by limiting the values of s_k to a small range. The smoothness term now reads as

$$V(l_k, l_{k-1}) = \lambda_s |s_k - s_{k-1}| + \lambda_d |d_k - d_{k-1} - s_{k-1}|.$$

Consequently changes in direction and depth discontinuities (compensated by the local slope value) are penalized.

Similar to Eq. (5) the search for the best previous state in a dynamic programming step now has to consider both, previous depths and slopes:

$$\begin{aligned} C_k(d,s) =& D_k(d) + \min_{\hat{l}=(\hat{d},\hat{s})} \left\{ C_{k-1}(\hat{d},\hat{s}) + V(l,\hat{l}) \right\} \\ =& D_k(d) + \min_{\hat{s}} \Big\{ \lambda_s |s - \hat{s}| \\ & + \min_{\hat{d}} \left\{ C_{k-1}(\hat{d},\hat{s}) + \lambda_d \cdot |d - \hat{d} - \hat{s}| \right\} \Big\}. \end{aligned}$$

The minimization over $\hat{d}$ has the same structure as beforehand and thus can be speeded up again by efficient computation of the lower envelope [18]. Due to the introduction of the slope variable we face a two-dimensional minimization problem. However, the number of possible slopes is quite small, e.g. $\mathcal{S} = \{-2,-1,0,1,2\}$. For each slope the min-convolution can be executed separately, which increases complexity by a factor of $|\mathcal{S}|$. Figure 3 shows the improved recovery of depths with the slope-based regularization (by means of a smoother and more accurate intersection boundary between a vertical structure and floor).

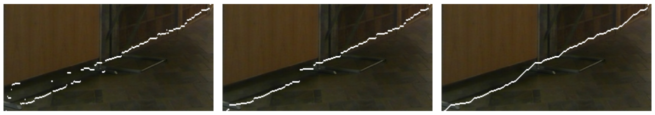
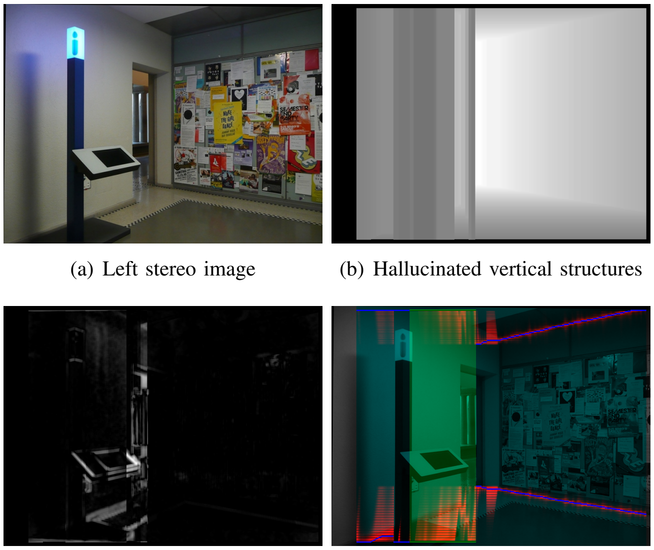


Figure 3. Improvement by introducing slope-based regularization (coutout from lower left part of scene in last row of Figure 6). Left to right: best cost, DP without slope, DP with 5 slopes.



(a) Left stereo image

(b) Hallucinated vertical structures

(c) Difference in matching costs

(d) Non-vertical parts (in green)

Figure 4. Model selection between vertical and non-vertical structures. (c) illustrates the matching cost difference between the pixel-wise best cost solution and the fitted vertical structures from (b); clearly visible is the error in columns containing the info screen.

B. Second Extension: Model Selection

Given that a scene contains a non-vertical structure, the algorithm tries to fit the best vertical model in terms of matching costs. Figure 4 illustrates such a case and shows the result for the vertical approximation in (b). In (c) we are comparing pixel-wise best matching costs (minimum in cost volume over all depth hypotheses) with the matching costs at depths described by the optimal fitted vertical structure. The result highlights exactly these areas where non-vertical structures (and also occlusions) are present. We can make use of this property by adding a new label to the optimization problem describing a non-vertical structure. The goal in the optimization then is to decide for a certain depth (assuming a vertical structure) or for a non-vertical structure.

The cost for a non-vertical structure along a column is the sum over pixel-wise minimal matching costs as motivated before. This sum will always be smaller than any cost aggregation over vertical structures; therefore, we add a constant bias B leading to

$$D_k(l = \text{non-vertical}) = B + \sum_{r=0}^{m-1} \min_d \mathbf{C}(k, r, (\mathbf{e}_z\ d)).$$

In the regularization a constant penalty t_2 is added for a label change between a vertical an a non-vertical structure and visa verse. Since we do not optimize for specific start and end points of a non-vertical structure, linearity terms are omitted. The final smoothness term results in

$$\bar{V}(l_k, l_{k-1}) = \begin{cases} V(l_k, l_{k-1}) & \text{if } l_k, l_{k-1} \text{ are vertical} \\ t_2 & \text{otherwise.} \end{cases}$$

VI. Experiments

In our experimental setup, we capture a scene from several view points. First, we run a structure from motion (SfM) pipeline to estimate camera poses. A pair of images is chosen from the sequence and aligned with the vertical direction. Second, we execute a plane sweep stereo matching to generate the cost volume; thereby 256 plane hypotheses are tested. Intensity differences are measured via SAD in a 7×7 matching window. Finally, costs for vertical structures are calculated and an optimal sequence is retrieved via dynamic programming. We optimize over 256 discrete depth values, resulting from the number of plane sweep planes. Costs for vertical structures are normalized to lie in the range $[0, 1]$; the same applies for disparity values. With that $\lambda_d = 3$ and smoothness terms are truncated above $t = 0.2$. The bias for costs supporting a non-vertical structure was set to $B = 1200$ (before normalization) and the penalty for a change between vertical and non-vertical models was set to $t_2 = 0.2$. We incorporate 5 possible slopes with $\lambda_s = 0.5$.

In Figure 5 results are illustrated for scenes predominantly featuring vertical structures. Computed depth maps are not absolutely accurate due to the strong vertical structure presumption, but provide dense depth estimates without artifacts for texture-less regions. Results for scenes were vertical and non-vertical structures coexist are shown in Figure 6. Occlusions and non-vertical structures correctly cause a model change. Finally, Figure 7 exhibits scenes were our depth estimation fails. It mainly occurs if the vertical assumption is clearly hurt or matching costs are inaccurate because of specular, transparent environments.

In terms of speed our plane sweep algorithm (GPU implementation) requires 160ms to generate the cost volume for images of sizes 768 × 576. The cost calculation for vertical and non-vertical structures takes 50ms (CPU based). Finally, basic dynamic programming is executed in 5ms; using 3 and 5 slope values execution times are 46ms and 120ms, respectively. Based on this measurements our approach is well suited for real time applications. Global stereo optimization [3] in comparison takes 2 seconds on a GPU processing down-sampled images of size 384×256. By comparison ELAS [2] is also very fast and has a run-time of 320ms (but takes rectified images as input). Full scanline optimization for stereo [1] (with GPU accelerated matching cost calculation) requires about 1.4s.

VII. Discussion

In addition to the approach presented in Section V we explored additional, potentially more powerful methods. First, for an image aligned with the vertical direction the semantic layout of floor, middle, and ceiling regions is a tiered one [20]. Hence the *simultaneous* determination of floor and ceiling boundaries in the image, and deciding whether the pixel column in between has either a vertical or a general depth structure is in principle possible in one dynamic programming pass. We initially considered using a label set consisting of depth values (for vertical columns) and index pairs indicating the floor and ceiling boundaries (for general columns). Using similar acceleration techniques as presented in [18] the complexity of dynamic programming is $O(n(L+2m))$ for an $m \times n$ image and considering L depth values, which we decided is too expensive for our target application. As reference, the presented implementation has a complexity of $O(nL)$.

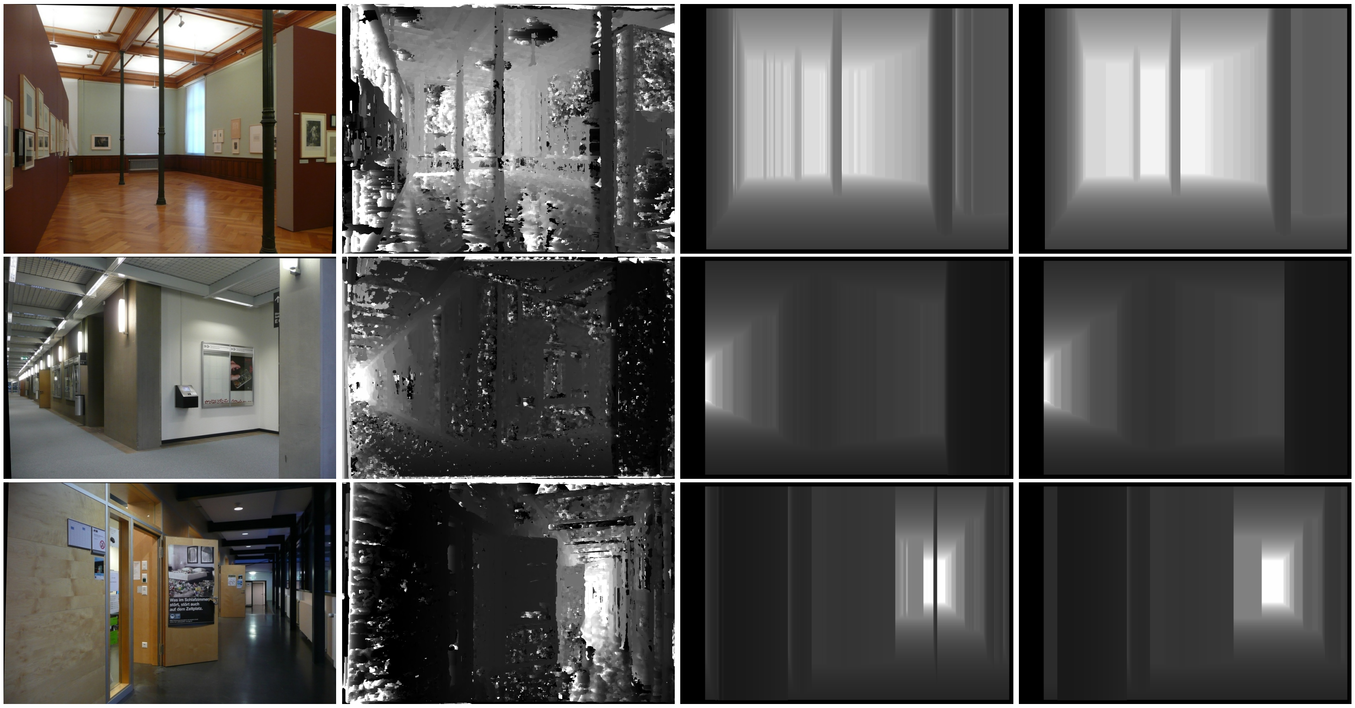
Since the computationally most expensive step is the matching cost calculation, one aims on replacing the general, expensive plane-sweep approach by a cheaper method. The plane-sweep method is only fast, if hardware support (e.g. a GPU providing fast texture sampling) is available. Otherwise, a standard stereo setup with aligned scanlines is preferable. This can be achieved, but only if the baseline between the cameras is parallel to the ground plane (or is very close of being parallel). In such a setting changing the depth of a fronto-parallel 3D plane amounts to shifting the image in horizontal direction. With the appropriate samples of depth values (corresponding to integral disparities), subpixel access can be avoided. This simplification is only available e.g. for driving robots, but not for humanoid (walking) ones or micro aerial vehicles.

VIII. Conclusion

We presented a stereo algorithm, which utilizes a strong vertical structure prior for dense depth map estimation. The prior is encoded by calculating costs along image columns, which are aligned with vertical 3D structures. This allows to employ a single dynamic programming optimization step over image columns to estimate the depth map. To account for planar structures we introduced a slope based regularization, and extended the approach to automatically detect areas where the vertical assumption is not met; in addition, the algorithm is also able to fill in these regions with plausible vertical elements. Finally, robust depth estimation for several scenes was demonstrated, and fast execution times enable the application of the proposed method in interactive and autonomous systems.

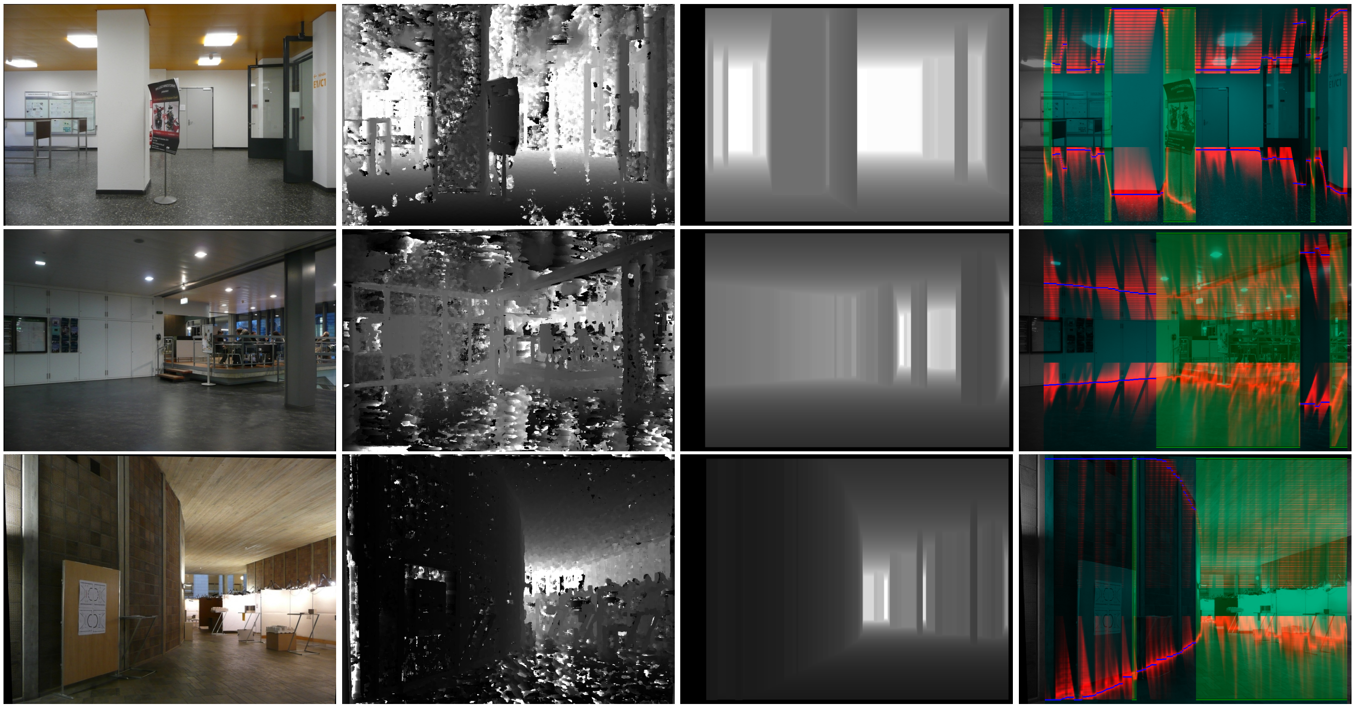
Acknowledgment

The authors would like to thank Georges Baatz for providing his implementation of the vanishing point detection algorithm.



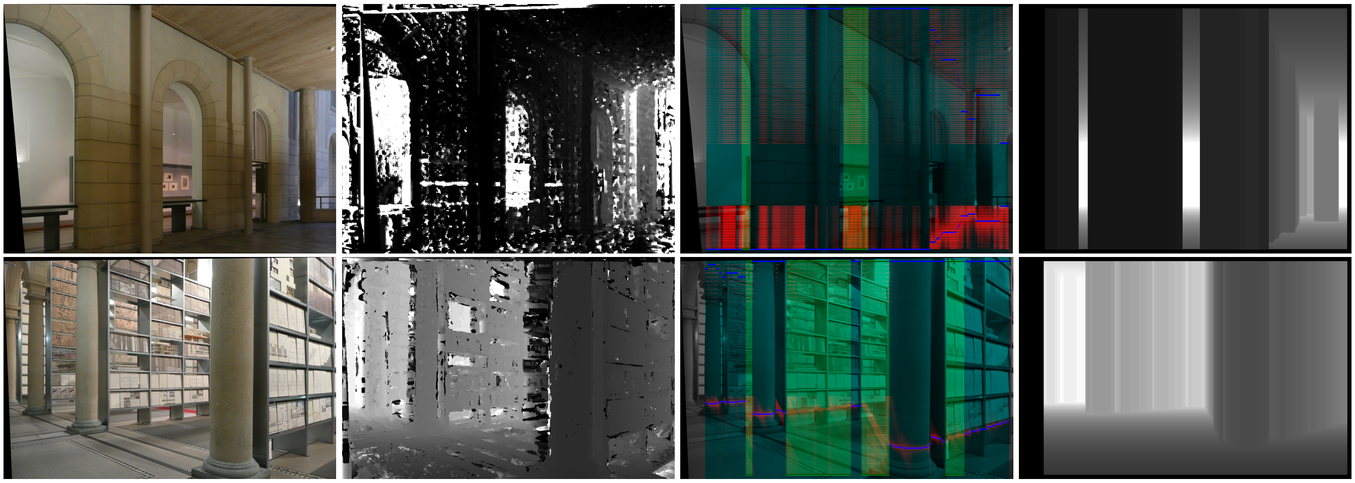
(a) Upright images (b) Best cost depth (c) Depth using vertical aggregation (d) Our result with DP

Figure 5. Depth maps for less textured indoor environments. Images also exhibit small non-vertical parts, e.g. ceiling, open doors and structured walls. Our depth maps in column (d) show a visually pleasing fit of vertical structure to the scenes.



(a) Upright images (b) Best cost depth (c) Best vertical depth after DP (d) Labeling of non-vertical structures

Figure 6. Depth calculation for scenes containing non-vertical, general structures. Column (d) illustrated detected non-vertical areas and occlusions with a green overlay. The blue line indicates the best sequence of indices (i, j) after DP. In red the likelihood (base on an exponential mapping of costs $D_k(d)$) for an index (i, j) is shown. (best viewed in color)



(a) Upright images (b) Best cost depth (c) Detected non-vertical structures (d) Depth, best vertical assumption

Figure 7. Failure cases. First row: Reflections on the glass and structure behind the arches violate the vertical structure assumption. Second row: The floor boundary of the vertical structures is set too high, since book shelves posses holes at their bottom resulting in less support for a vertical structure continuation. As a consequence DP wrongly estimates large parts of the scene as non-vertical. (best viewed in color)

REFERENCES

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1-3, pp. 7–42, 2002.

[2] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Proc. ACCV*, 2010.

[3] C. Zach, M. Niethammer, and J.-M. Frahm, "Continuous maximal flows and Wulff shapes: Application to MRFs," in *Proc. CVPR*, 2009, pp. 1911–1918.

[4] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *Proc. CVPR*, 2007.

[5] B. Micusik and J. Kosecka, "Multi-view superpixel stereo in urban environments," *IJCV*, vol. 89, pp. 106–119, 2010.

[6] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, "Manhattan-world stereo," in *Proc. CVPR*, 2009, pp. 1422–1429.

[7] S. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *Proc. ICCV*, 2009.

[8] N. Cornelis, B. Leibe, K. Cornelis, and L. V. Gool, "3D urban scene modeling integrating recognition and reconstruction," *IJCV*, vol. 78, no. 2–3, pp. 121–141, 2008.

[9] D. C. Lee, M. Hebert, and T. Kanade, "Geometric reasoning for single image structure recovery," in *Proc. CVPR*, 2009.

[10] A. Flint, C. Mei, D. Murray, and I. Reid, "A dynamic programming approach to reconstructing building interiors," in *Proc. ECCV*, 2010, pp. 394–407.

[11] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, "Reconstructing building interiors from images," in *Proc. ICCV*, 2009.

[12] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum, "Symmetric stereo matching for occlusion handling," in *Proc. CVPR*, 2005, pp. 399–406.

[13] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Proc. ICPR*, 2006, pp. 15–18.

[14] Z.-F. Wang and Z.-G. Zheng, "A region based stereo matching algorithm using cooperative optimization," in *Proc. CVPR*, 2008.

[15] D. Gallup, J. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *Proc. CVPR*. IEEE, 2010, pp. 1418–1425.

[16] J. Kosecka and W. Zhang, "Video compass," in *Proc. ECCV*. Springer-Verlag, 2002, pp. 476–490.

[17] G. Schindler, P. Krishnamurthy, and F. Dellaert, "Line-based structure from motion for urban environments," in *Proc. 3DPVT*. IEEE, 2006, pp. 846–853.

[18] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *IJCV*, vol. 70, no. 1, pp. 41–54, 2006.

[19] A. Amini, T. Weymouth, and R. Jain, "Using dynamic programming for solving variational problems in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 9, pp. 855–867, 1990.

[20] P. F. Felzenszwalb and O. Veksler, "Tiered scene labeling with dynamic programming," in *Proc. CVPR*, 2010.