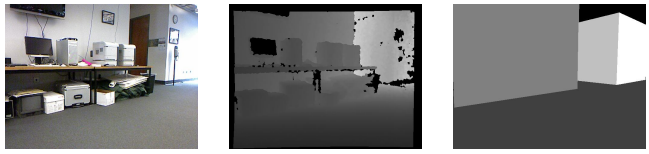
Parsing Indoor Scenes Using RGB-D Imagery

Camillo J. Taylor and Anthony Cowley
GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104
Email: [cjtaylor,acowley]@cis.upenn.edu

***Abstract*—This paper presents an approach to parsing the Manhattan structure of an indoor scene from a single RGB-D frame. The problem of recovering the floor plan is recast as an optimal labeling problem which can be solved efficiently using Dynamic Programming.**

I. Introduction

This paper considers the problem of parsing RGB-D images of indoor scenes, such as the one shown in Figure 1 to extract an underlying floor plan defined by the delimiting walls.



(a) (b) (c)

Fig. 1. a. RGB image b. Depth Image c. Inferred floor plan

It is important to note that this parsing problem involves more than simply identifying the wall surfaces in the scene. While this is a necessary first step, the extracted walls are effectively infinite planes supported by a finite set of depth measurements. In order to produce a floor plan one must reason about the extent of each plane and how the walls interact with each other to form corners, occlusion edges and other structures.

Furthermore, in a typical scene one must contend with the fact that wall segments are often occluded by furniture, pedestrians and general clutter and that the depth measurements may be inaccurate or missing entirely. An effective parsing procedure needs to be able to infer the extents of the wall surfaces even in the presence of ambiguous, uncertain and incorrect input measurements. The approach proposed in this paper deals with all of these problems by formulating the parsing process as an optimal labeling problem which can be solved exactly and efficiently using Dynamic Programming.

A number of researchers have addressed the related, but more challenging problem of inferring the 3D structure of a scene from monocular imagery. Gupta et al. [5] describe an interesting system for reasoning about scene structure using qualitative geometric and structural constraints. Hedau and his colleagues [6, 7] have explored algorithms for inferring the layout of indoor scenes based on vanishing points and other cues. Lee et al. [9] present an interesting approach to reasoning about scene structure using volumetric constraints. Saxena and Ng [12] recast the problem in terms of a Markov Random field and infer the scene structure based on learned models. Furukawa et. al [4] describe an impressive system for recovering the structure of indoor scenes that utilizes a sequence of monocular image and employs a sophisticated but expensive volumetric analysis procedure. In this paper we make use of the range data provided by the Kinect sensor which simplifies the interpretation problem and provides more accurate parsing results.

Recently Silberman and Fergus [13] have addressed the problem of scene analysis using RGB-D data. In this work the analysis problem is framed in terms of pixel labeling where the goal is to assign each pixel in the frame an appropriate class label. The goal in our work is to go beyond a labeling of the visible pixels and to instead propose a coherent floor plan that accurately extrapolates the underlying structure of the scene even in the face of clutter. In this paper we choose to focus on extracting the floors and walls since these represent the major structures which delimit the extent of the scene and provide the underlying context for other structures such as doors, walls, tables and chairs.

The approach to interpretation taken in this paper is most similar to the one given by Lee, Hebert and Kanade [10] who propose an approach to extracting the Manhattan structure of a frame based on vanishing points and an analysis of possible corners in the scene. Our approach makes use of the 2.5D structure of the image in the same way that they do but takes a different approach to formulating and solving the parsing problem. The proposed approach is also similar to the parsing scheme described by Flint et al.[3, 2] who also make use of Dynamic Programming to efficiently propose interpretations of indoor scenes from monocular imagery. The principal differences between this work and that of Flint et al.is that it takes as input an RGB-D frame and begins by explicitly extracting planar surfaces, further the subsequent dynamic programming optimization procedure is phrased in terms of the RGB-D measurements as opposed to monocular and stereo cues.

Several schemes have been proposed to address the problem of interpreting point cloud data. Rusu et al. [11] describe an impressive system for parsing range scans acquired from indoor kitchen scenes. Toshev et al. [14] describe a scheme that has been used to automatically parse range scans to produce building models. Anguelov et al. [1] and Lalonde et al. [8]

describe schemes for classifying regions in point cloud data sets to identify, buildings, trees and other structures.

Most of these schemes were designed to work offline in a batch fashion. In this context one can afford to make several passes over the data to identify nearest neighbors, or to fuse neighboring regions. The goal in this work is to develop a scheme that can ultimately be run in an online fashion so that it can be used to parse the data as it is being acquired. Another salient difference is the fact that this approach seeks to exploit the relationship between the 2D range image and the associated imagery to accelerate the interpretation process.

II. Technical Approach

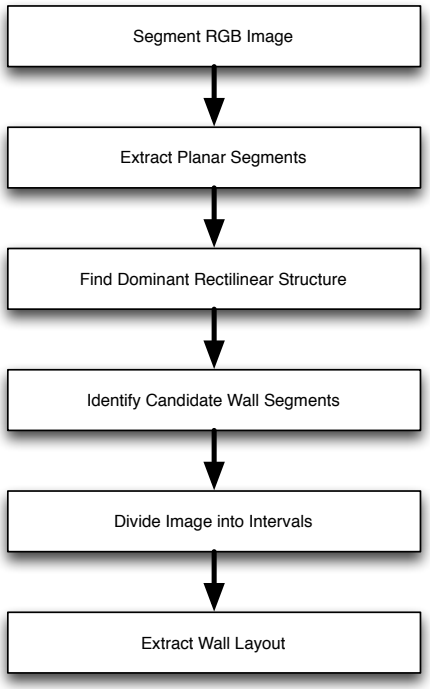
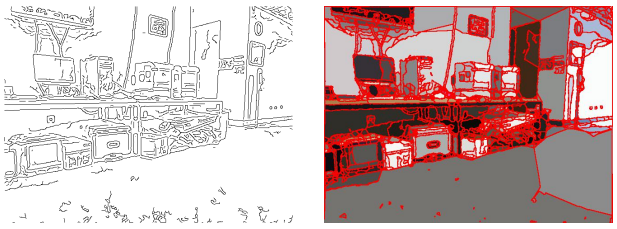


Fig. 2. Flowchart of overall parsing procedure.

The overall procedure for parsing the scene based on an RGB-D image is outlined in the flowchart given in Figure 2. The first stage in the pipeline segments the input RGB image into regions based on extracted image edges. The second stage of processing uses the image segmentation as a prior to search for planar surfaces in the scene. The third stage identifies the floor of the scene and estimates the dominant rectilinear orientation. The fourth stage considers the set of extracted planes and identifies segments that could serve as walls. The fifth stage breaks the image up into intervals based on the extracted wall segments. The final stage estimates the layout of the scene by labeling each interval with the index of the underlying wall.

Each of these stages is explained in more detail in the following subsections using the scene shown in Figure 1 as a running example.

A. Image Segmentation

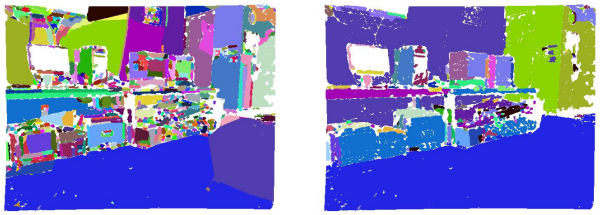


(a) (b)

Fig. 3. a. Extracted Intensity Edges b. Image Segmentation

The first step in our analysis is an edge accurate segmentation scheme that breaks the RGB imagery into coherent, disjoint regions. The image segmentation procedure begins with a standard Canny edge extraction step which finds significant discontinuities in the intensity image as shown in Figure 3a. The detected edgels are then passed to a Delaunay Triangulation procedure which produces a triangular tessalation of the image. The resulting triangular graph is then segmented using an agglomerative merging procedure that repeatedly merges the two regions with the lowest normalized boundary cost. These merging costs are computed by considering the average HSV color in each region. This procedure can be efficiently implemented using a heap data structure and the entire segmentation procedure can be carried out in 0.1 seconds on a typical image in Matlab.

B. Extracting Planar Surfaces



(a) (b)

Fig. 4. a. Extracted Planes b. Result after coplanar segments are merged

The regions derived from the image segmentation are used to suggest groupings of the depth samples from the range imagery. More specifically, the depth samples associated with each of the image regions are passed to a RANSAC routine which is used to recursively divide the point set into planar regions. A key advantage of the proposed approach is that the image segmentation procedure is very effective at suggesting useful groupings so very few RANSAC iterations are needed to discover the structures of interest. Effectively, the image segmentation serves to focus the computational effort of the procedure on groupings that are likely to yield fruitful interpretations so the procedure is able to discover relevant groupings quickly even in complex environments with several surfaces.

It is important to keep in mind that the depth measurements produced by the Kinect sensor are derived from structured light via triangulation as opposed to time of flight. As such the device is best thought of as measuring disparity which is inversely related to depth. One practical consequence is that the error in the depth estimates increases rapidly as one gets further away from the sensor which argues against standard approaches to fitting planes to points based on the residual error in 3D.

In the proposed scheme the planar surfaces are fit to the RGB-D measurements by exploiting the observation that planar surfaces in the scene will project to planar regions in the disparity image. This can be seen by taking the standard equation for a plane in the coordinate frame of the sensor:

$$n_x X + n_y Y + n_z Z = c$$

dividing through by scene depth, Z, to obtain:

$$n_x \frac{X}{Z} + n_y \frac{Y}{Z} + n_z = c \frac{1}{Z}$$

and noting that $u = \frac{X}{Z}$ and $v = \frac{Y}{Z}$ correspond to the normalized image coordinates while $w = \frac{1}{Z}$ denotes the measured disparity at that coordinate. This means that planar regions in the scene can be extracted by fitting affine models to the disparity in each image region.

Figure 4 shows the results of the planar interpretation procedure on the sample scene. Here the different colors correspond to different planar segments that were recovered. These planar segments are then passed to a greedy merging procedure which seeks to group coplanar segments into extended regions as shown in Figure 4b.

C. Finding Dominant Rectilinear Structure

The planar extraction procedure returns a set of segments which can then be analyzed to identify salient structures. The analysis procedure assumes that the vertical axis of the image is roughly aligned with the gravity vector. Given this assumption, the first step in the interpretation procedure is to identify the floor plane by searching for a large planar region near the bottom of the image whose normal is approximately vertical and which appears to underly most of the other 3D points. The normal to this plane defines the direction of the gravity vector in the RGB-D sensors frame of reference.

Candidate wall segments are identified by looking for extended planar segments whose normals are perpendicular to this gravity direction. Each candidate wall segment effectively defines an associated rectilinear orientation for the scene where the z-axis corresponds to the gravity direction, the x-axis corresponds to the normal to the wall segment and the y-axis is simply the cross product of these normal vectors. This rectilinear orientation can be compactly represented with a single rotation matrix $R_{cw} \in SO(3), R_{cw} = \begin{bmatrix} \hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}} \end{bmatrix}$ which captures the orientation of the RGB-D sensor with respect to the Manhattan structure of the scene.

The interpretation system cycles through each of the candidate wall segments and scores the associated rectilinear orientation by determining how many of the other planar surfaces are aligned with one of the cardinal axes. The candidate rotation matrix with the most support is chosen as the dominant rectilinear orientation. The fourth column of Figure 8 depicts the extracted rectilinear structure by showing how various segments are aligned with the dominant axes. Segments colored blue are aligned with the gravity direction or z-axis. Segments colored red or green are aligned with the x and y axes respectively.

In addition to the walls extracted by the fitting procedure, 4 additional walls are added to form an axis aligned bounding box that surrounds all of the recovered points. This bounding box serves as a 'backdrop' providing a default interpretation for every point in the scene.

D. Identify Candidate Walls and Wall Segments

Once the dominant rectilinear structure of the scene has been established, the system identifies extracted planar segments that may be walls in the scene. This is accomplished by finding planar structures in the scene whose normals are aligned with either the x or y axis and that have an appropriate horizontal and vertical extent. Note that in practice the walls are often partially occluded by furniture and other clutter so there is no requirement that the wall segment extend from the floor to the ceiling. The third column of Figure 8 shows the results of the analysis that identifies wall segments. Note that for the purposes of our experiments tall cabinets and office partitions can serve as walls since they have an appropriate extent and serve to delimit the floor plan of the scene.

Once the candidate walls have been identified in the image, the points that make up that segment are further divided into contiguous sections called wall segments. This partitioning accounts for the fact that a wall may stop and start in the scene as the dominant wall in Figure 1 does. These wall segments represent continuous stretches of wall surface observed in the RGB-D image. Figure 5 makes the distinction between walls, which are modeled as infinite planes, and wall segments which are thought of as finite sections of wall surface observed in the RGB-D image.

E. Divide Image into Intervals

Figure 6 shows an idealized image of an indoor scene where the vertical axis of the camera is aligned with the gravity direction. In this case vertical lines in the scene will project to vertical lines in the image. In particular the vertical lines corresponding to corners in the scene or to points where one wall segment occludes another would effectively subdivide the horizontal field of view into a sequence of disjoint intervals as shown. Each interval in the image would be associated with a wall. This structure was noted and exploited by Lee, Hebert and Kanade [10] in their work on parsing indoor scenes from single images.

The parsing scheme proposed in this paper proceeds by breaking the image into a sequence of intervals and then associating a wall with each span to produce an interpretation of the scene. The endpoints of the intervals are derived from

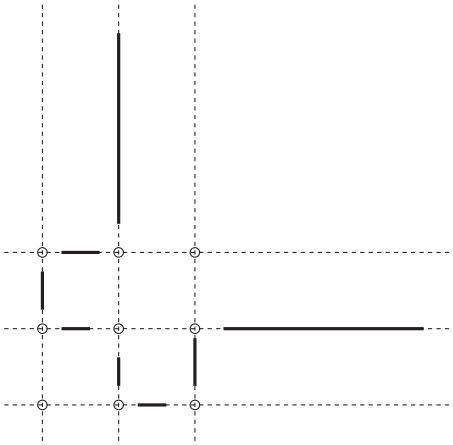


Fig. 5. An overhead view of a floor plan of the scene indicating the walls, which are considered to have infinite extent and are depicted by dashed lines, and the wall segments, which are depicted by the solid line segments. These wall segments correspond to contiguous planar regions in the RGB-D image. The interpretation system also considers all of the possible intersections between perpendicular walls, denoted by circles in the figure, since these may correspond to corners in the scene.

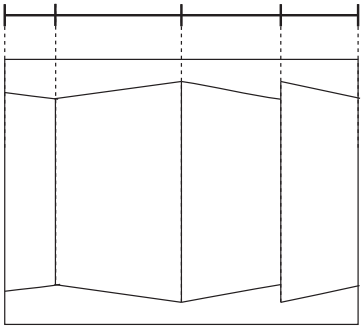


Fig. 6. Idealized figure of a frame showing how the horizontal field of view can be divided into non-overlapping intervals each of which will be associated with an underlying wall.

the extents of the extracted wall segments and from the inferred intersections between all pairs of perpendicular walls. These intersections are depicted by circles in Figure 5 . All of these putative endpoints are projected into the horizontal field of view of the sensor and then sorted from right to left to define image intervals as shown in Figure 6. Including more endpoints than needed is not a problem since the subsequent labeling stage can consolidate neighboring intervals as needed.

Note that we *do not* require our input images to be perfectly aligned with gravity as shown in this idealized view since the rotation matrix R_{cw} recovered in the rectilinear analysis stage allows us to effectively rectify the input image to account for the tilt and roll of the sensor.

F. Extract Wall Layout

As noted in the previous subsection, the scene interpretation procedure is rephrased as a labeling problem where the goal

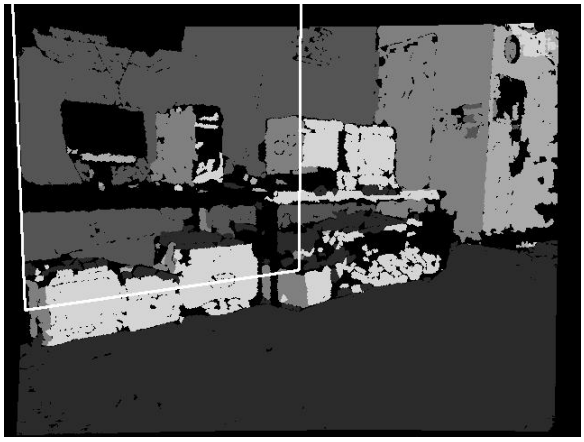


Fig. 7. For each pixel in the frame we can compute the index of the wall that best explains the observed disparity and orientation. We can then evaluate how well a particular wall explains a given image interval by projecting the wall into the frame at that interval and determining how many pixels in that region agree with the proposed label. The quadrilateral demarcated on the frame corresponds to the projection of one of the prospective walls into one of the image intervals.

is to label each interval in the horizontal field of view with an associated wall segment. The extents of the intervals would define the extents of the walls from the point of view of the observer.

This can be viewed as a graph labeling problem where the nodes in the graph are the intervals and neighboring intervals are connected by edges to form a simple 1D chain. We associate a cost with assigning wall labels to each of the intervals and a cost to assigning different wall labels to neighboring segments and then find a labeling which minimizes the overall cost.

In order to assign costs to each interval, the procedure first determines the optimal wall segment label for each pixel in the frame. This is accomplished by projecting each wall into the RGB-D frame and comparing the disparity and orientation predicted at each pixel with the observed disparity and orientation. The predicted depth or disparity at each pixel can easily be calculated based on the best fit normal to the plane. The system considers each of the extracted walls in turn and each pixel in the frame retains the index of the wall that comes closest to predicting the observed disparity and surface normal at that point.

Once this initial labeling step has been completed, we can evaluate the cost of assigning a particular image interval to a particular wall candidate by projecting a quadrilateral defined by the interval endpoints and the ceiling and floor heights into the frame as shown in Figure 7. The system considers all of the pixels that lie within that region and measures what fraction of these pixels do *not* have the wall in question as their first choice. This fraction indicates the cost of assigning the wall label to the interval. The lower the number, the better the match.

This score is also weighted by the apparent size of the interval in the image. More specifically a normalized interval

size is computed by taking the angle subtended by the interval in the image and dividing it by the effective horizontal field of view of the image to produce a number between 0 and 1. The product of this normalized interval size and the fractional score mentioned in the previous paragraph is used as the final label score for the interval.

Unfortunately it is typically not sufficient to simply assign each interval the index of the wall with the lowest cost. Missing depth measurements, occlusion and inherent ambiguities serve to introduce spurious labelings. The proposed scheme makes use of a global optimization procedure to enforce smoothness and produce a more coherent interpretation in the face of these uncertainties. The labeling procedure exploits the fact that the nodes in the graph we are labeling form a simple chain which allows us to find the globally optimal assignment efficiently via Dynamic Programming.

In order to encourage continuity in the labeling process a cost is associated with each of the edges in the chain. This transition cost is designed to capture the cost associated with assigning adjacent intervals to different wall segments. If the proposed labeling induces a significant change in depth at the junction between the two image intervals then a fixed transition penalty is applied, in our experiments this transition penalty was fixed at 0.03. Note that if two perpendicular walls meet to form a corner there is no depth discontinuity and the associated transition penalty would be zero.

The objective function that we are interested in minimizing takes the following form:

$$\mathcal{O}(\mathbf{l}) = \sum_{\mathbf{i=1}}^{\mathbf{n_i}} (\mathbf{f_i}(\mathbf{l_i}) + \mathbf{e_i}(\mathbf{l_i}, \mathbf{l_{i-1}})) \tag{1}$$

Where n_i denotes the number of intervals, $f_i(l_i)$ denotes the cost of associating interval i with wall label l_i and $e_i(l_i, l_{i-1})$ represents the transition cost associated with assigning interval i the label l_i while interval $i-1$ is assigned the label l_{i-1}.

The optimization problem can be tackled in stages where at each stage the system considers a series of optimization problem of the form:

$$D_{i,j} = \min_k [f_i(j) + e_i(j,k) + D_{i-1,k}] \tag{2}$$

Where $D_{i,j}$ represents the cumulative cost of assigning interval i the label j at this stage. The Dynamic Programming procedure systematically considers all of the legal labels that can be assigned to each interval. Note that at each stage this optimization problem uses the results from the previous optimization stage. The computational complexity of the optimization problem $O(n_i n_w^2)$. Where n_w denotes the number of wall labels. On a typical problem n_i is on the order of 20 while n_w is on the order of 10 so the computational cost of the optimization is usually quite small. The final output of the optimization procedure is a labeling of each of the image intervals. This labeling can be used to construct the 2.5D scene models shown in Figure 8.

The overall strategy of phrasing the interpretation process as an optimal labeling problem that can be solved with Dynamic Programming is similar to the approach proposed by Flint et al.[3, 2] however the schemes used to define the interpretation costs are quite different because this approach exploits RGB-D imagery.

III. Experimental Results

In order to evaluate the effectiveness of the proposed algorithm it was applied to a data set of 38 RGB-D images taken from various vantage points in a typical office environment. For each of the test images an interpretation score was manually generated by counting the number of image intervals where the wall assignment suggested by the automatic procedure differed from the humans assessment. A score of zero would indicate that the system had correctly labeled all of the wall surfaces in the scene while a score of 1 would indicate that one of the image intervals was labeled incorrectly.

On 20 of these images the procedure produced a completely correct interpretation extracting all wall surfaces even in the presence of significant clutter, on sixteen of the images one of the recovered wall segments was labeled incorrectly. In the vast majority of these cases the erroneous parse covers a small section of the frame and the error is caused by incomplete range data. On two of the frames the procedure failed to produce an intelligible result. The same parameter settings were used on all of the examples.

Figure 8 shows samples of the scenes that the system parsed correctly. Note that the scheme was able to handle situations with significant amounts of clutter such as the third and fifth case. It can also correctly handle cases with relatively complex occlusion structures as in the second example. Note that the system correctly recovers small features like the edges of the doorway on the second example and the structure of the water cooler alcove in the third example. It is also able to deal correctly with the clutter in the fourth and fifth examples.

Figure 9 shows examples of the cases where one of the planes is labeled incorrectly. Note that in all of these cases the errors are fairly subtle and the gross structure of the scene is actually recovered quite well. For example In the fifth case the person in the foreground is labeled as a wall, in the fourth example the end of the corridor is not parsed correctly because it is beyond the range of the sensor.

The entire segmentation and analysis procedure is implemented in Matlab and it takes approximately 6 seconds to run the complete analysis on a typical RGB-D image on a Macbook Pro laptop. No attempt has been made to optimize the code and we expect that a more efficient implementation would run significantly faster.

All of the code and datasets used in this paper are freely available online at the following website: http://www.cis.upenn.edu/~cjtaylor/RESEARCH/projects/RGBD/RGBD.html.

IV. Conclusions and Future Work

This paper has presented an approach to parsing the floor plan of an indoor scene from a single RGB-D frame by

finding a set of candidate walls and delimiting their extent in the image. The problem of parsing the scene is recast as an optimal labeling problem which can be solved efficiently using Dynamic Programming. In this sense, the method exploits the 2.5D structure of the image to simplify the scene interpretation problem.

The analysis provides a compact description of the overall structure of the scene in terms of a floor plane and wall segments. This representation can serve as a basis for further semantic analysis which would identify other structures such as doors, tables, chairs and windows. We note that while the Manhattan structure is a convenient cue, it is not essential to this approach. The same basic scheme could be employed to parse environments where some of the walls do not adhere to the rectilinear model.

Future work will seek to merge parse results from a sequence of RGB-D frames into larger floor plans. Here the Manhattan structure provides a convenient framework for accumulating information about recovered wall segments into a coherent map in an incremental fashion. Such an approach could be used to produce semantic decompositions of extended regions which could be useful in a variety of robotic applications.

ACKNOWLEDGMENTS

This research was partially sponsored by the Army Research Laboratory Cooperative Agreement Number W911NF-10-2-0016 and by the National Science Foundation through an I/UCRC grant on Safety, Security, and Rescue Robotics.

REFERENCES

[1] D. Anguelov, B. Taskarf, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of markov random fields for segmentation of 3d scan data. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 169 – 176, jun. 2005.

[2] A. Flint, D. Murray, and I. Reid. Manhattan scene understanding using monocular, stereo, and 3d features. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2228 –2235, nov. 2011.

[3] Alex Flint, Christopher Mei, David Murray, and Ian Reid. A dynamic programming approach to reconstructing building interiors. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, volume 6315 of *Lecture Notes in Computer Science*, pages 394–407. Springer Berlin / Heidelberg, 2010.

[4] Yasutaka Furukawa, Brian Curless, Steven M. Seitz, and Richard Szeliski. Reconstructing building interiors from images. In *International Conference on Computer Vision*, pages 80–87, Kyoto, October 2009.

[5] Abhinav Gupta, Alexei A. Efros, and Martial Hebert. Blocks world revisited: image understanding using qualitative geometry and mechanics. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 482–496, Berlin, Heidelberg, 2010. Springer-Verlag.

[6] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV '09)*, 2009.

[7] Varsha Hedau, Derek Hoiem, and David Forsyth. Thinking inside the box: using appearance models and context based on room geometry. In *Proceedings of the 11th European conference on Computer vision: Part VI*, ECCV'10, pages 224–237, Berlin, Heidelberg, 2010. Springer-Verlag.

[8] J. F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert. Natural terrain classification using three-dimensional ladar data for ground robot mobility. *Journal of Field Robotics*, 23(10):839–861, 2006.

[9] David Changsoo Lee, Abhinav Gupta, Martial Hebert, and Takeo Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. *Advances in Neural Information Processing Systems (NIPS)*, 24, November 2010.

[10] D.C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2136 –2143, june 2009.

[11] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3D Point Cloud Based Object Maps for Household Environments. *Robotics and Autonomous Systems Journal (Special Issue on Semantic Knowledge)*, 2008.

[12] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:824–840, 2009.

[13] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *Proceedings of the International Conference on Computer Vision - Workshop on 3D Representation and Recognition*, 2011.

[14] A. Toshev, P. Mordohai, and B. Taskar. Detecting and parsing architecture at city scale from range data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 398–405, 2010.

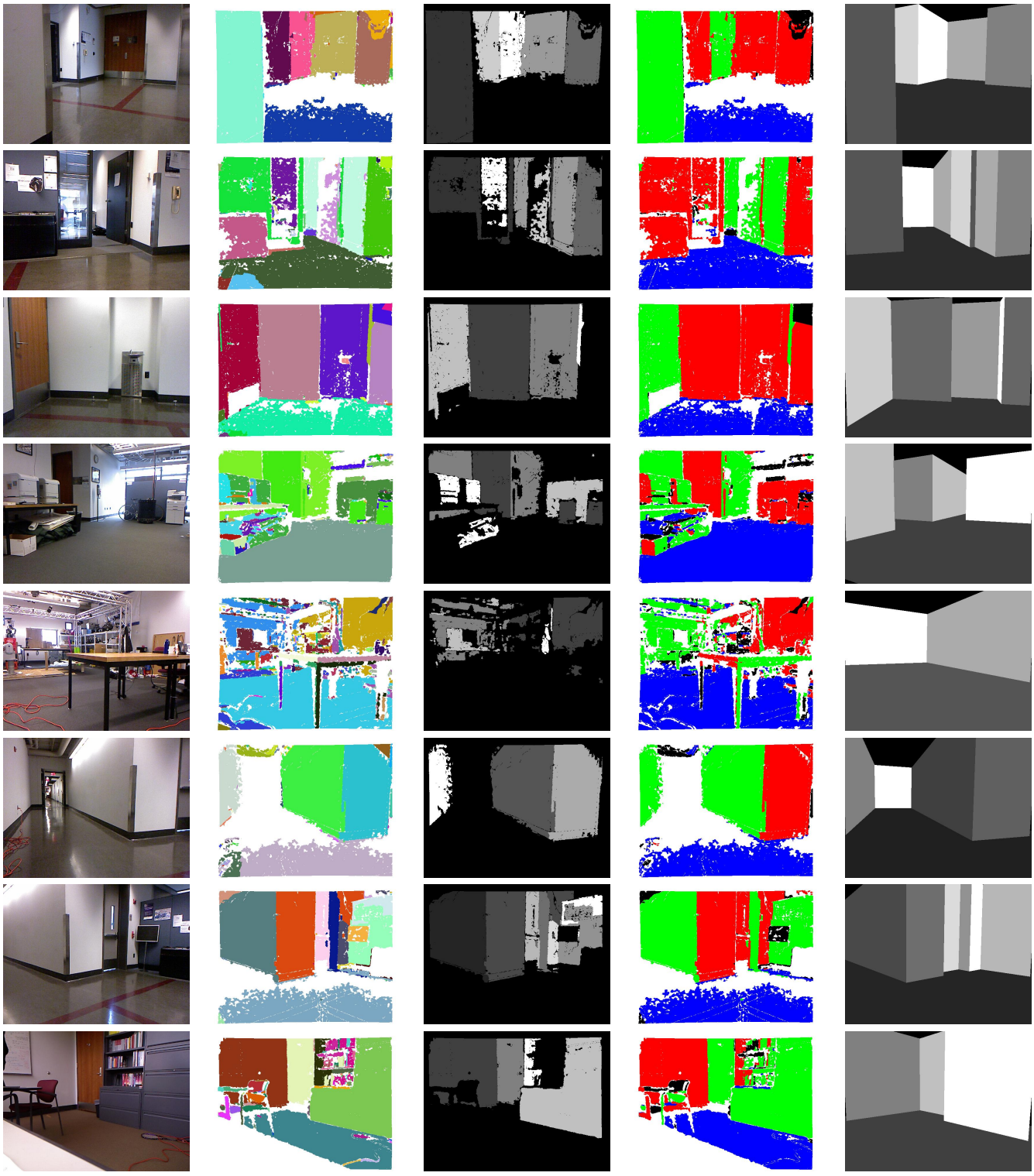


Fig. 8. Figure showing some successful scene parsing results. The second column shows the shows the planar segments extracted from the scene, different colors are used to label points belonging to different planes. The third column shows the points belonging to candidate wall segments, the fourth column shows shows the results of the analysis that finds the dominant rectilinear structure, Horizontal surfaces are blue and vertical surfaces are red or green depending on their orientation. The final column shows the results of parsing the image into wall segments, each pixel is assigned a label corresponding to the underlying wall.

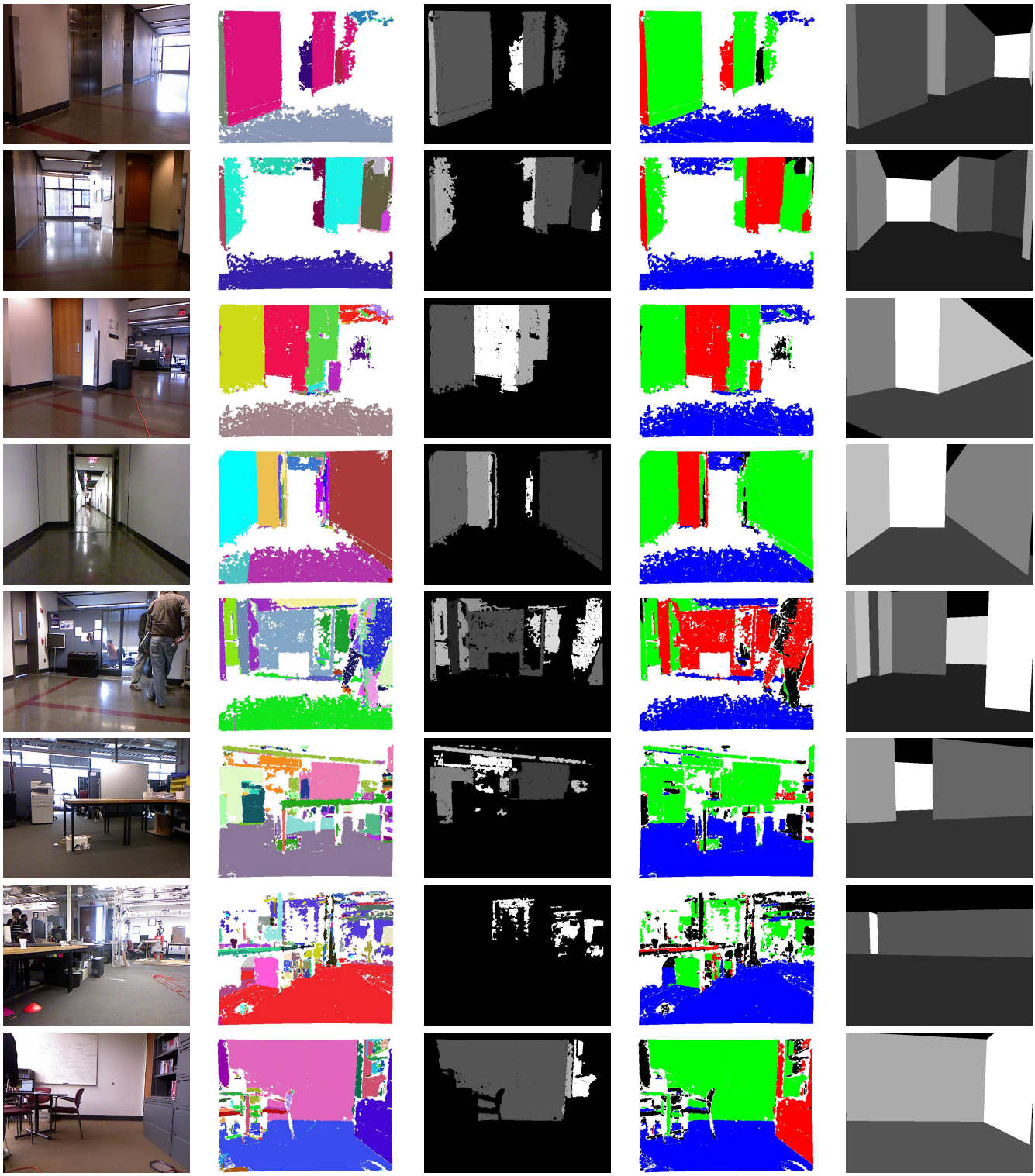


Fig. 9. Figure showing cases where one of the recovered planes has an error. The second column shows the shows the planar segments extracted from the scene, different colors are used to label points belonging to different planes. The third column shows the points belonging to candidate wall segments, the fourth column shows shows the results of the analysis that finds the dominant rectilinear structure, Horizontal surfaces are blue and vertical surfaces are red or green depending on their orientation. The final column shows the results of parsing the image into wall segments, each pixel is assigned a label corresponding to the underlying wall.