

Real-Time Geometric Scene Estimation for RGBD Images using a 3D Box Shape Grammar

Andrew R. Willis^a and Kevin M. Brink^b

^aUniversity of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223

^bAir Force Research Laboratory, Munitions Directorate, Eglin AFB, FL, 32542

ABSTRACT

This article describes a novel real-time algorithm for the purpose of extracting box-like structures from RGBD image data. In contrast to conventional approaches, the proposed algorithm includes two novel attributes: (1) it divides the geometric estimation procedure into subroutines having atomic incremental computational costs, and (2) it uses a generative “Block World” perceptual model that infers both concave and convex box elements from detection of primitive box substructures. The end result is an efficient geometry processing engine suitable for use in real-time embedded systems such as those on an UAVs where it is intended to be an integral component for robotic navigation and mapping applications.

1. INTRODUCTION

A fundamental challenge in robotics seeks to endow robotic agents with the capability to map and navigate geometrically complex environments. While high-altitude Unmanned Aerial Vehicles (UAVs) may use the Global Positioning System (GPS) and satellite maps for navigation, more detailed maps and more accurate robot position data are required for ground-level navigation, especially in GPS-denied scenarios. The navigation problem is further exacerbated by numerous sources of geometric variability such as weather, moving objects, and land development. Hence, robotic agents must construct maps of the *in-situ* 3D geometry of their environment and simultaneously track their position within this map to solve this problem; a process referred to as Simultaneous Localization and Mapping (SLAM). Here, robots track their position in geometric maps constructed from sensed 3D (X, Y, Z) surface data. Sensors are typically either passive imagery, e.g., 3D structure from multiple camera images, or active imagery, e.g., Light Distance And Ranging (LiDAR), laser triangulation or color+depth (RGBD) cameras.

The SLAM problem was introduced nearly 30 years ago¹ and continues to be a fundamental challenge in robotics today. Solutions to the SLAM problem require algorithms that solve three challenging problems: (1) how to efficiently recognize salient geometry, (2) how to associate and integrate map data, and (3) how to track the position and orientation of the robot in the map. The major stumbling block in current SLAM approaches is that the proposed solutions to these problems overburden the available computational resources. The computational bottleneck comes from three sources: (1) processing that filters and associates new data to the map, (2) processing that updates the map with new information and (3) processing that tracks uncertainties in the updated map geometry and robotic position. Compromising simplifications of these problems are typically used reduce the computational burden at the cost of accuracy which prevents current SLAM methods from building accurate maps over large regions.²

This article describes a computationally efficient approach for the first stage in the SLAM data processing pipeline: efficient recognition and modeling of geometric data. Our efforts in this category focus on real-time extraction of perceptual geometric patterns from RGBD depth images of man-made and indoor scenes. RGBD data, like most geometric measurement data, consists entirely of points. Hence, our algorithm detects and estimates parameters for higher level organizations of the geometric data. When accurate models are found that closely fit the measured data, one can replace large subsets of the measured data with a small number of model parameters. Working on these parameters directly reduces the space and time resources needed to analyze,

Further author information: (Send correspondence to A. Willis)

A. Willis: E-mail: arwillis@unc.edu, Telephone: 1 704 687 8420

match and integrate measured surface data. Since the SLAM problem is computationally bound, advances in these areas promise to mitigate limitations in existing SLAM approaches due to computational cost.

Our algorithm design applies generative shape models inspired by the “Block World” models for scene data and related shape grammar models.³⁻⁵ In contrast to prior approaches, which focus on convex block models and offline processing of 2D image data, we model 3D RGBD surface data in real-time and extract both convex and concave box models from detections of their sub-structures.

The perceptual rules applied by our grammatical estimator seeks to endow robots with capabilities to estimate spatial groupings of box-like structures commonly found in indoor scenes. Solving for the unknown models in real-time from generic RGBD scene data poses difficult challenges which include:

1. Reliable segmentation of 3D depth data into spatially-contiguous irregular planar regions.
2. Reliable estimation of the 3D box substructures which include: (1) unbound planar regions (irregular surface patches), (2) quadrangular planar regions (sides), (3) perpendicular pairwise planar intersections (edges), and (4) mutually perpendicular three-plane junctions (corners).
3. Reliable perception of complete 3D box models from collections of estimated box substructures, e.g., given estimates for a box corner and one box side one may estimate the entire 3D box structure and its scene pose. The same is true for other box substructures, e.g., estimates of the three different box sides.

In contrast to existing algorithms which estimate planes using a local-to-global approach our algorithm seeks performs cursory local tests to identify large-scale geometric structures within the image. This sparse set of semi-global structure is taken as the initial geometric representation of the measured scene and is the lowest resolution model for the measured image data available from our algorithm requiring the smallest amount of processing time.

While it is possible to attempt to detect planes, their adjacencies and higher order structures simultaneously. Algorithms that proceed in this way are more difficult to regulate from a computational cost perspective. Complexity originates from the fact that the scene data within each tile may exhibit a wide variety of geometric phenomenon. Algorithms that seek to identify and estimate multiple models, especially higher-order models, e.g., quadratic surfaces, require additional computation at the atomic, i.e., tile, level. The impact of this effect is that the time required for the algorithm to complete an initial model estimate for an entire depth image is highly variable. Our application seeks to perform segmentation in real-time, a primary performance metric is to create estimation procedures which impose atomic incremental computational costs whose variance is small for generic values of the depth image data. This allows real-time robotic systems to regulate the time allocated to the geometric estimation algorithm. When integrated as part of a robotic system, this algorithm design allows client applications, e.g., robotic navigation and control software, to more efficiently control their computational budget. For example, when most depth image data is out-of-range or from a small number of planes, a smaller computational budget for geometric analysis may be appropriate. Conversely, when the depth image data includes rich and informative geometric shape, a larger computational budget may be allocated to process the depth image data. Dynamic load-balancing in this way can produce machines that efficiently leverage their computational resources to maximize the amount of information that they extract from measured sensor data for each CPU clock cycle.

To summarize, there are three contributions which we attribute to the work in this article:

1. We propose a new algorithm that estimates and perceptually groups planar surfaces in RGBD images into higher-order box substructures with the goal of describing scene data in terms of a compact collection of parameters for boxes and their substructures.
2. A new statistical hypothesis testing framework is proposed that affords computational savings to solve the perceptual grouping problem. The approach fits planar models to point data only once. Subsequent work uses the model fitting results and error statistics to form increasingly complex/compound hypotheses from the initial data without recomputing model parameters.

3. Our algorithm design allows for explicit control of the computational cost. When integrated as part of a robotic system, this algorithm design allows client applications, e.g., robotic navigation and control software, to more efficiently control their computational budget.

2. PRIOR WORK

Work on segmentation and perceptual inference for geometric is found across the computer graphics, robotics and computer vision literature. Given the breadth of interest in this subject our review here selects a subset of the most relevant which are generally applicable to the topic of real-time geometric scene estimation or the topic of rule-based, i.e., grammatical, inference of scene structure.

In the general robotics community, researchers have sought methods to quickly extract planes from RGBD data.^{6–10} One popular approach^{11,12} applies pattern recognition techniques to associated measured points to a collection of planes. To do this, the planar surface is estimated at image locations and then a clustering algorithm groups points having similar coefficients to a single plane. While this approach can effectively cluster points to their respective planes, disjoint planar surfaces having the same plane equation cluster spatially-disjoint groups of points to a single segment. The lack of spatial coherence in segments is a significant problem for perceptual grouping algorithms such as ours which seeks to discover box edges faces and corners in the data. Problems arise when one considers adjacency hypotheses such as box edges and corners which require a known relative spatial arrangement for planes. As such, the generic point-to-plane classification via clustering approach is not applicable for our algorithm.

Other efforts from the SLAM literature use planar models to expedite the SLAM estimation, association and integration map-building tasks.^{13–16} Much of the effort in this research is to not only discover planar segments in the 3D data, but also to merge these segments into a global map. Hence, much of the focus of these articles is to define parameterizations of the detected planar data that facilitate efficient solutions to downstream SLAM map-building problems which require multiple depth images / surface estimates to be associated and merged. As such, their planar surface estimation approaches do not seek to extract perceptual information similar to that proposed in this article.

More sophisticated extensions to planar surface SLAM models add rule sets that enforce mutual orthogonality.^{17,18} Yet, these approaches are applied for sparse laser scan data and the mutually orthogonal ruleset applied is leveraged to extract hallway and room structures and does not generalize to arbitrary convex or concave rectilinear structures.

The computer vision community has primarily applied generative models to estimate structures in 2D images. A notable subset of this work^{4,5} focuses on “Block World” and seeks to estimate 3D containers for images of block-like objects, e.g., buildings. These approaches apply rules to infer the pose of the 3D block using convexity assumptions that key on intensity changes on the surface of building exteriors to detect fold edges for the convex box geometries. Application here is primarily to outdoor scenes and seeks to model man-made scene objects, e.g., buildings, as volumetric block-like objects.

3. METHODOLOGY

Our grammatical approach for depth image processing seeks to extract primitive shapes from the image. Each primitive is viewed as a symbol in the grammar and attributed a probabilistic distribution. Constellations of symbols in the RGBD data reveal an organization for these symbols and our algorithm must search for the shape “words” and their parameters that best for the observed data.

Processing seeks to quickly move from the primitives provided in the measured data, e.g., 3D surface points, to higher-dimensional primitives having increasing structural complexity and information. Geometries of interest for our box world and their associated dimension are: points (0D), lines (1D), planes (2D), pairwise-plane intersections (edges) (2.5D) and triplets of mutually perpendicular intersecting planes (3D). While the fundamental box model is a primitive geometric object, we envision this algorithm as a subcomponent to a higher-level geometric analysis engine that takes as input the parameters for estimated box primitives and provides potential explanations for specific spatial groupings of these boxes, e.g., desk, bookshelf, door, hallway, room etc.

Our algorithm for box estimation takes as input a scale parameter, W , a time budget, T , and a quality criteria, Q . The algorithm terminates when either the time budget, T , expires or the quality criteria, Q , is met. We express this criteria as a stopping function $Stop(T, Q)$ which is true under the previously mentioned criteria. The time budget is specified in *ms.* and the quality criteria is specified as a percentage. The quality criteria is satisfied when the ratio of labeled data to measured data exceeds Q . Hence, a value of $Q = 0.8$ allows the algorithm to terminate early when 80% of the image data has been labeled. Using these initial parameters, our algorithm proceeds in as described in the steps below:

1. If $Stop(T, Q) = false$, invoke the planar region detector algorithm at scale W as described in §3.1. Note that the initial invocation of this function detects planar regions at all image locations. Subsequent invocations only process locations that are not already labeled as members of a previously estimated plane. Planar region detection is a two-stage process: (1) grow existing planar regions and (2) detect new planar regions. As output, the detection algorithm returns a set of irregular planar surfaces, S . Otherwise, go to Step 5.
2. If $Stop(T, Q) = false$, invoke the box semantic analysis algorithm as described in §3.2. This algorithm takes as input detected planes, S , and provides as output parametric estimates for higher-order box substructures. These substructures include faces, F , face edges, E , and box corners, V , found in the depth image. Otherwise, go to Step 5.
3. If $Stop(T, Q) = false$, invoke the box estimation algorithm as described in §3.3. This algorithm takes as input the set of higher-order box substructures, $\{F, E, V\}$, and provides as output estimates for complete 3D boxes, B , found in the depth image. Otherwise, go to Step 5.
4. If $Stop(T, Q) = true$, go to Step 5. Otherwise, set $W = \frac{W}{2}$ and go to Step 1.
5. Return the estimated box parameters $\Theta = \{S, F, E, V, B\}$ and exit.

As evident in steps (1-5) our algorithm consists of three subroutines: (1) a detector/estimator for planar regions, (2) an estimator for box substructures, and (3) a box estimator. These subroutines are discussed in the following three sections.

3.1 Scale-Space Extraction of Planar Regions

Our scale space extraction algorithm proceeds as described in the steps below:

1. A quad-tree decomposition is imposed on the depth image, $Z(x, y)$, having an initial $W \times W$ tile size.
2. A visitation sequence is prescribed to traverse the quad-tree tiles. Our visitation sequences is raster-scan order starting in the top left corner of the image.
3. Apply the chosen planar hypothesis test, discussed later in this section, to data in each tile.

The hypothesis test fits a planar model to the surface points in the tile and then evaluates a test statistic to either reject or accept the null hypothesis. When the data supports the alternate hypothesis, a segment label, l , is attributed to this region and the quadrangular planar polygon used as a model for the tile data.

4. For tiles having a planar model, a search procedure visits adjacent tiles to find candidate adjacencies between planar segments. For those adjacent tiles found that contain valid planar hypotheses, a pairwise planar model equality hypothesis test, also discussed later in this section, is posed.

This hypothesis asks if the planar models (A, B) from adjacent tiles are equal. In contrast to the hypothesis of (a), no model must be fit and this hypothesis can be answered by evaluating the likelihood of data from tile B given the model from tile A and vice-versa. When more than one adjacent and compatible model is identified, the hypothesis having more power (less error) is accepted.

5. Hypothesis test results from steps (3) and (4) produce a set of planar models having distinct labels. Each planar model is attributed with a connected set of $W \times W$ tiles in the image whose surface data lie close to the plane.

Detection and Estimation of Planar Domains

The generic rationale for estimation of planar segments is to adopt a hypothesis testing procedure and then apply these hypothesis tests to a scale-space/quad-tree decomposition of the depth image. Hypothesis tests required by our algorithm are:

1. Planar model hypothesis:

Null Hypothesis: The data does not originate from a planar surface.

Alternate Hypothesis: The data originates from a planar surface.

When the alternate hypothesis is accepted, a quadrangle of depth data is assigned a planar model (4 coefficients). The value of the planar coefficient vector, α , and the test statistic, T_1 , is saved.

2. Planar model equality hypothesis:

Null Hypothesis: The two planar models approximate distinct 3D surfaces.

Alternate Hypothesis: The two planar models approximate the same 3D surface.

When the alternate hypothesis is accepted, the two planar models are merged into a single planar model. The value of the planar coefficient vector, α , and the test statistic, T_2 , is saved.

Our general representation for planes uses the four-parameter implicit representation as shown in equation (1).

$$n_X X + n_Y Y + n_Z Z + d = \alpha^t \mathbf{p} = 0 \quad (1)$$

Our plane parameter notation incorporates the 3D surface normal $\mathbf{n} = [n_X, n_Y, n_Z]^t$ into the coefficient vector and, in its vector form, $\alpha = [\mathbf{n}, d]^t$, uses the homogeneous form for 3D coordinates; $\mathbf{p} = [X, Y, Z, 1]^t$. Note that this representation in (1) is referred to as the Hessian normal form which allows the perpendicular signed distance between a point \mathbf{p} and the the plane, $d(\mathbf{p}|\alpha)$, to be easily computed as $d(\mathbf{p}|\alpha) = \alpha^t \mathbf{p}$. If necessary, we negate the plane coefficients to restrict the Z -component, n_Z , to have a negative value which, for depth image data, ensures that our plane equations consistently represents surface orientation using an outward-pointing surface normal.

Our results apply two popular models as examples of planar model hypothesis tests:

1. A three-point plane model,
2. An explicit least squares plane fit.

For (1) we compute the equation of the plane coefficients as follows: (1) select three (x, y) point locations in the quadrangle having valid depth measurements, (2) compute the 3D locations for the three points, $\{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\}$, using the RGBD reconstruction equations (3), (3) compute coefficients, $\hat{\alpha}$, of the plane passing through the chosen points using the standard approach: $a\mathbf{n} = (\mathbf{p}_2 - \mathbf{p}_1) \times (\mathbf{p}_3 - \mathbf{p}_1)$, $\hat{\alpha} = [a\mathbf{n}, \mathbf{p}_1 \cdot \mathbf{n}]^t$ where $a\mathbf{n}$ denotes the scaled surface normal. Note the computational cost of model fitting (step (3)) is fixed; $O(18)$.

For (2) we switch the form of our model from the implicit plane model of equation (1) to an explicit algebraic plane model or Monge patch surface which is generally faster to estimate as there are only three unknown variables rather than four. The model for our Monge patch surface is shown in equation (2).

$$Z(x, y) = n'_X X + n'_Y Y + d' = \beta^t \mathbf{p}' \quad (2)$$

where $\beta = \left[\beta_X = \frac{n_X}{n_Z}, \beta_Y = \frac{n_Y}{n_Z}, \beta_d = \frac{d}{n_Z} \right]^t$ and the Z -coordinate of our 3D points is omitted from the homogeneous 3D coordinate; $\mathbf{p}' = [X, Y, 1]$. For hypothesis testing, we compute the equation of the plane coefficients as follows: (1) select N uniformly distributed random (x, y) point locations in the quadrangle, (2) compute the (X, Y) coordinates of the N points, $\{\mathbf{p}'_1, \mathbf{p}'_2, \dots, \mathbf{p}'_N\}$, using the RGBD reconstruction equations (3), (3) compute the coefficients of the explicit plane, $\hat{\beta}$, that minimizes the function $\hat{\beta} = \min_{\beta} \sum_{i=1}^N \|\beta^t \mathbf{p}'_i - Z_i\|^2$. Note that the computational cost of model fitting (step (3)) requires computation of a 3x3 scatter matrix, it's pseudo-inverse,

and three matrix multiplications whose computational cost is $(N^3 + 3N^2 + 12N + 27)$ which we truncate to be simply $O(N^3)$.

A planar model equality hypothesis for the two planes, (α_i, α_j) , is evaluated using the test statistic $T(\alpha_i, \alpha_j) = \sum_{(i,j)=1}^N (\alpha_i \mathbf{p}_j)^2 + (\alpha_j \mathbf{p}_i)^2$. In this case, we sample N points from α_i denoted \mathbf{p}_i and N points from α_j denoted \mathbf{p}_j . We then symmetrically compute the squared orthogonal distance between the points from one plane to the other plane. In cases where the planes are equal, these distances will be zero. Otherwise, the squared distance is positive and indicates that the planes are distinct. We threshold this statistic using the square of the expected measurement noise. Since RGBD measurement noise is depth dependent, our threshold uses the average Z values of the planar regions it contains as Z . Using the RGBD noise model,¹⁹ our threshold is then computed as $\tau(\alpha_i, \alpha_j) = N(1.425e^{-3})Z^2$. If $T(\alpha_i, \alpha_j) < \tau(\alpha_i, \alpha_j)$, we reject the null hypothesis and merge the planar models. The merge process weights coefficients by the number of observed inliers for each plane to obtain a new coefficient vector for the merged plane model.

RGBD 3D Point Reconstruction

Measured 3D (X, Y, Z) positions of sensed surfaces can be directly computed from the intrinsic RGBD camera parameters and the measured depth image values. The Z coordinate is directly taken as the depth value and the (X, Y) coordinates are computed using the pinhole camera model. In a typical pinhole camera model 3D (X, Y, Z) points are projected to (x, y) image locations, e.g., for the image columns the x image coordinate is $x = f_x \frac{X}{Z} + c_x - \delta_x$. However, for a depth image, this equation is re-organized to “back-project” the depth into the 3D scene and recover the 3D (X, Y) coordinates as shown by equation (3)

$$\begin{aligned} X &= (x + \delta_x - c_x)Z/f_x \\ Y &= (y + \delta_y - c_y)Z/f_y \\ Z &= Z \end{aligned} \tag{3}$$

where Z denotes the sensed depth at image position $D(x, y)$, (f_x, f_y) denotes the camera focal length (in pixels), (c_x, c_y) denotes the pixel coordinate of the image center, i.e., the principal point, and (δ_x, δ_y) denote adjustments of the projected pixel coordinate to correct for camera lens distortion.

3.2 Box Substructure Estimation Algorithms

Our algorithm seeks to identify substructures in the following order: (1) estimate plane-plane intersections (edges) as a 3D line lying on two mutually perpendicular surfaces, (2) estimate faces (as four mutually perpendicular 3D edges that adjoin a planar surface to other planar surfaces, and (3) vertices as locations where three mutually perpendicular edges intersect.

Note that all box junction types create instances of 3D lines. Let (α_i, α_j) denote a pair of planes. The parametric form of the line that traces their intersection is given by $\mathbf{l} = \lambda \mathbf{v} + \mathbf{p}$ where \mathbf{p} is a point on the line and \mathbf{v} is a vector in the direction of the line. The parameters of the line can be computed from the coefficients of a plane pair as follows: (1) compute the direction of the line, $\mathbf{v} = \mathbf{n}_i \times \mathbf{n}_j$, (2) solve the equations $\alpha_i \mathbf{p} = \mathbf{0} = \alpha_j \mathbf{p}$ for some 3D point \mathbf{p} . Solutions to these equations lie in a 1-dimensional subspace since the specific location of the point \mathbf{p} is unconstrained. A unique solution to this under-determined system is obtained by assigning \mathbf{p} 's z -component to 0 which fixes the point \mathbf{p} to lie in the $Z = 0$ plane. Doing so yields the following equations for the x and y components of $\mathbf{p} = \left[\frac{-n_{y_i}d_j + n_{y_j}d_i}{v_z}, \frac{-n_{x_i}d_j + n_{x_j}d_i}{v_z}, 0 \right]^t$.

Our search for box sub-structures seeks to detect and estimate parameters for the plane-plane intersections which are critical components to every non-trivial box substructure. To do so, we consider the 3D lines generated by pairwise intersections of our estimated planar segments. The equation for each 3D line of intersection is computed as described above. An algorithm then seeks to detect measurements from this line in the RGBD image to re-estimate a more accurate set of line parameters and to infer the box substructures of interest. The steps of this algorithm are below:

1. The equation of each 3D edge line is projected into the RGBD depth image using the projection equations of §3.1 and the line segment of the 3D line visible in the RGBD image is computed.
2. Using the parametric equation of the projected 2D line, we then traverse the visible line segment in the depth image and, for each point along the line, we compute the depth difference between our 3D line and the measured RGBD image depth.
3. At locations where the difference in depth is small, we hypothesize that the RGBD depth image contains measurements from the estimated line model as a model estimation procedure re-computes the 3D line model parameters from the depth image data.
4. When the hypothesis suggests that the line lies along a plane-plane intersection, the resulting line segment is listed as a plane-plane edge substructure.

Reliable estimation of plane-plane edges greatly simplifies extraction of higher order structures. Extraction of these structures is detected as part of the plane-plane edge extraction process. Specifically, once a plane-plane edge has been detected, we search the list of existing plane-plane edges and compare endpoints of the previously found 2D line segments to detect higher-order plane edge substructures. Face substructures are detected by their a-priori known shape constraints which require neighboring edge directions to be perpendicular and opposite edge directions to be parallel. Additionally, we restrict their spatial organization to ensure opposite edges have similar lengths and perpendicular edges share endpoints. Corner substructures are found when three mutually perpendicular edges are found that share a common endpoint.

Re-Fitting 3D Line Models to RGBD Data

When a 3D line describing a pairwise plane intersection is found to pass close to measured RGBD data, we hypothesize that the RGBD depth image contains measurements from the estimated line model. However, noise in the plane estimation procedure is amplified in our line estimates which depend on stability in pairwise planar cross product ($\mathbf{v} = \mathbf{n}_i \times \mathbf{n}_j$). Hence, when planar lines of intersection do exist in the RGBD data, our estimate for these lines rarely align with the true intersection line apparent in the RGBD image data.

Our algorithm to re-estimate the line parameters selects two positions along the line and traverses the RGBD image data in direction perpendicular to the line to locate the closest local extrema in the depth data along that line. The choice of maximum or minimum is determined by the curvature of the planar pair. Specifically, if $\mathbf{n}_i - \mathbf{n}_j < 0$, the surface is concave with respect to the view direction (+z-axis) and we seek a local minima. Otherwise, the surface is convex and we seek a local maxima. Once the two positions of the extrema are located, $(\mathbf{p}_A, \mathbf{p}_B)$, the new line parameters are computed by taking one of the extrema locations as the location parameter, $\mathbf{p} = \mathbf{p}_A$, and the difference between the extrema point pair as the line direction, $\mathbf{v} = \mathbf{p}_B - \mathbf{p}_A$.

3.3 Box Estimation

While plane detection and box-substructure estimation procedures can require model fitting. These estimates serve to provide semantic level information for inference of box structures. Hence, estimation of the box structure seeks to compute the parameters of a box from the box substructures detected in the scene data. Our approach considers the following two substructure sets for estimation and how the box parameters are computed from the substructure parameters:

1. A box estimate from an estimate of a box corner, C , and two more box faces, F . Here we use the directions of the line segments at C to define the orientation of the box. The three lengths of the sides aligned with the corner directions determine the (X, Y, Z) dimensions for the box.
2. A box estimate from an estimate of three distinct box faces, F . Here, we extract the three distinct, mutually orthogonal of the face edges to define the orientation of the box. The three lengths of the sides aligned with the corner directions determine the (X, Y, Z) dimensions for the box.

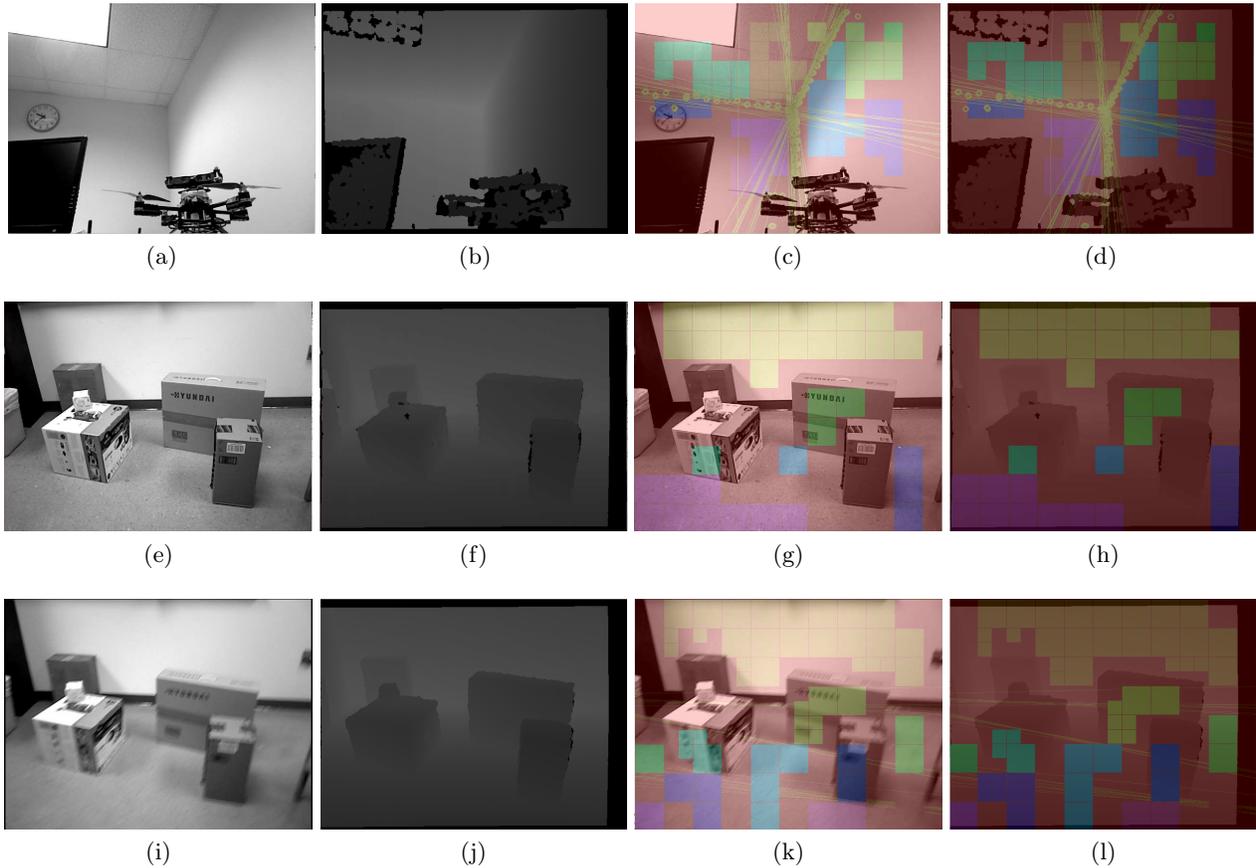


Figure 1: (left two columns) show the gray scale and depth images respectively. (right two columns) superimpose results from the three stages of the box estimation algorithm on the grayscale and depth images. The outcome of the plane detection and estimation stage is depicted as colored tiles where each color indicates a distinct plane. The lines shown in result image depict the initial data for Stage 2, box substructure estimation, which processes the collection of lines defined as the 3D intersection of the detected planes of Stage 1. The circles shown in (c,d) depict the positions on the estimated box edges. As shown in (c,d) the three detected box edge lines converge on the concave corner of the office where a box corner is located.

4. RESULTS

Figure 1 depicts results of the proposed algorithm for three images recorded from an Asus XTion Pro Live RGBD sensor. The algorithm was implemented as a node in the Robot Operating System development platform. Experimental results were generated by providing the algorithm pre-recorded RGBD data from file captured from the XTion sensor. The algorithm was run on a Lenovo Yoga 3 Pro laptop which has an Intel Core M-5Y71 CPU with a 1.20GHz clock rate and no GPU or other hardware acceleration was used in the algorithm implementation. Recorded data depicts indoor scenes from an office and include images of the office ceiling (top row) and images of a collection of boxes placed on the office floor (middle, bottom rows).

The first two columns of figure 1 show excerpts from the sensed gray scale and depth image data. The second two columns of figure 1 superimpose outcomes the algorithm stages onto the RGB and depth image data. The outcome of Stage 1, the plane detection and estimation stage, is depicted as colored tiles where each color indicates a distinct plane. Lines shown lie at the intersection of the estimated 3D planes. These lines serve as data for Stage 2, box substructure estimation stage, which processes these lines to detect and estimate perpendicular plane adjacencies visible in the image referred to as box edges. As mentioned earlier, box edges are then used to estimate box corners and entire boxes.

Images from the top row of figure 1, depict the upper corner of an office (top right) with some geometric clutter including a monitor (bottom left) and a quadrotor vehicle in the foreground (bottom right). The depth image contains significant noise due to the bright light fixture in the image (top left), the average depth of the plane adjacencies ($\sim 4\text{m}$) and the orientations of the right wall (right) and ceiling (top) planes which are nearly parallel to the camera view direction. Noise is evidenced by a number of distinct plane detections for both the right wall and ceiling planar segments shown as tiles in the images having distinct colors while the back wall, which is nearly perpendicular to the view direction, is covered by two planar segments. Intersection lines of the detected planar segments are shown which localize candidate locations for perpendicular plane adjacencies, i.e., box edges. As mentioned previously, these lines are noisy estimates of the locations where the back wall, ceiling, and right wall touch. The box substructure algorithm searches these locations for local maxima of the curvature to detect instances of visible box edges in the depth image shown as circles in the results image. The results for the office image show the three edge lines detected in the image that converge on the concave corner of the office where a box corner is located. Note that the numerous noisy plane estimates generate numerous candidate box-edge lines. Convergence of these lines to the same local extrema of the depth curvature indicates that these planar segments share the same box edges which allows these segments to be merged after box edges have been computed. Parameters of the algorithm for this image were $W = 40$ and only one iteration of the algorithm was allowed requiring an average of 5ms. Images of Figure 1 in the middle and bottom rows show two distinct views of a collection of boxes laid out on the office floor taken while moving the camera through the office. In both cases the initial quad-tree resolution defined tiles having size $W = 60$ pixels. However, for the middle image row, only one level of the quad-tree model was expanded while two levels of the quad-tree were expanded in the bottom row. Here, the algorithm required an average of 25ms. to process the two level quad-tree (bottom row).

Figure 2 depicts a labeling of the box image data provided by the algorithm. Here, the planes extracted from the image were used to classify the image data. In this image the subset of points lying within 3 standard deviations to some detected plane are labeled and each point is assigned the label of the closest plane. Coherence in the spatial labels are apparent for the box faces shown for the three largest boxes in the foreground; shown as green, blue and cyan segments. Other planar segments detected in the image include the office floor (purple) and back wall (yellow). The visible portions of touching planar surfaces are identified for a number of the scene objects. This includes the floor-wall adjacency and the adjacency locations between the floor and each of the three box faces in the foreground. These are important semantic relationships that indicate box edges for the foreground objects and, in combination with the wall-floor adjacency indicates the global relative pose of these boxes with respect to the orientation of the office itself. Extraction of this information shows promise for robotic SLAM applications where it can have multiple impacts. For RGBD odometry, the extracted plane and line features can be used to track changes in pose over time to track the motion of a robot within indoor contexts in real-time. For localization, the extract box substructures and the associated appearance information from the RGB image can provide important data for solving difficult loop closure problems where robot sense measurements from the

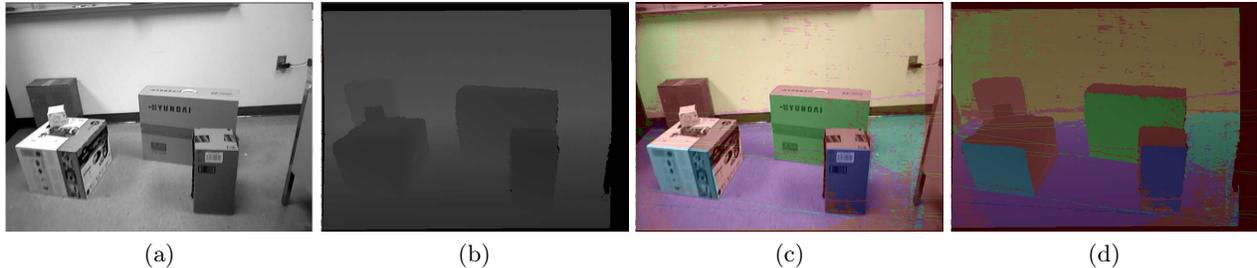


Figure 2: (a-d) show a segmentation of a 3D scene into box primitives using the proposed algorithm. (a,b) show the gray scale and depth images respectively. (c,d) superimpose colors on the images (a,b) showing distinct planar segments as separate colors. Due to their apparent size in the image and their relatively low noise content, planar segments having orientations that are nearly perpendicular to the camera view are extracted from the image. Three of these planar segments include entire box faces (green, blue, cyan) which originate from convex (small) boxes completely in the view. Two planar segments (purple, yellow) are subsets of a complete box face that are associated with the office wall (yellow) and the office floor (purple). Colored lines that traverse through the image depict edge line initial locations.

same box-like object at temporally and/or spatially distant locations. For mapping, detection of the orientation of the environment shows promise for defining coordinate systems for map computation whose grid aligns with the spatial organization of the scene.

5. ACKNOWLEDGMENT

This research is sponsored by an AFRL/National Research Council fellowship and results are made possible by resources made available from AFRL's Autonomous Vehicles Laboratory at the University of Florida Research Engineering Education Facility (REEF) in Shalimar, FL.

6. CONCLUSION

This article describes a real-time algorithm that detects and estimates boxes and their substructures in RGBD images. The algorithm includes a novel framework for estimation that efficiently extracts complex semantic groups of planar structures from the measured data using a hierarchical hypothesis testing framework. Algorithm design is sensitive to resource bound systems and includes termination criteria to limit the time consumed by the algorithm within a larger computational budget. As such, we envision the algorithm provides useful information for embedded systems such as UAVs where it is intended to be used for robotic navigation and mapping applications.

REFERENCES

1. R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Rob. Res.* **5**, pp. 56–68, Dec. 1986.
2. J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó, "The SLAM problem: A survey," in *Artificial Intelligence Research and Development, Proceedings of the 2008 Conference on*, pp. 363–371, IOS Press, (Amsterdam, The Netherlands, The Netherlands), 2008.
3. G. Stiny and J. Gips, "Shape grammars and the generative specification of painting and sculpture," in *Information Processing '71*, C. V. Friedman, ed., pp. 1460–1465, (Amsterdam), 1972.
4. A. Gupta, A. A. Efros, and M. Hebert, "Blocks world revisited: Image understanding using qualitative geometry and mechanics," in *European Conference on Computer Vision (ECCV)*, 2010.
5. H. Kim and A. Hilton, "Block world reconstruction from spherical stereo image pairs," *Computer Vision and Image Understanding* **139**, pp. 104–121, Oct. 2015.
6. A. Trevor, J. Rogers, and H. Christensen, "Planar surface SLAM with 3D and 2D sensors," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 3041–3048, May 2012.

7. Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 5182–5189, May 2013.
8. N. Srinivasan and F. Dellaert, "A Rao-Blackwellized MCMC algorithm for recovering piecewise planar 3D models from multiple view rgb-d images," in *International Conference on Image Processing*, IEEE, (Paris), October 2014.
9. E. Ataer-Cansizoglu, Y. Taguchi, S. Ramalingam, and T. Garaas, "Tracking an rgb-d camera using points and planes," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pp. 51–58, Dec. 2013.
10. Y. Lu and D. Song, "Robust rgb-d odometry using point and line features," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
11. D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, *RoboCup 2011: Robot Soccer World Cup XV*, ch. Real-Time Plane Segmentation Using RGB-D Cameras, pp. 306–317. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
12. J. Xiao, B. Adler, and H. Zhang, "3d point cloud registration based on planar surfaces," in *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pp. 40–45, Sept 2012.
13. P. Henry, D. Fox, A. Bhowmik, and R. Mongia, "Patch Volumes: Segmentation-based Consistent Mapping with RGB-D Cameras," in *International Conference on 3D Vision (3DV)*, 2013.
14. E. Herbst, P. Henry, and D. Fox, "Toward Online 3-D Object Segmentation and Mapping," in *International Conference on Robotics and Automation (ICRA)*, 2014.
15. H. Barron-Gonzalez and T. Dodd, "RBPF-SLAM based on probabilistic geometric planar constraints," in *Intelligent Systems (IS), 2010 5th IEEE International Conference on*, pp. 260–265, July 2010.
16. P. Ozog and R. M. Eustice, "Real-time SLAM with piecewise-planar surface models and sparse 3D point clouds," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1042–1049, (Tokyo, Japan), November 2013.
17. F. Barrera, F. Lumbreras, and A. D. Sappa, "Multispectral piecewise planar stereo using manhattan-world assumption," *Pattern Recognition Letters* **34**(1), pp. 52–61, 2013. Extracting Semantics from Multi-Spectrum Video.
18. V. Nguyen, A. Harati, A. Martinelli, N. Tomatis, and B. Sa, "Orthogonal SLAM: a step toward lightweight indoor autonomous navigation," in *In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2006.
19. K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors* **12**(2), p. 1437, 2012.