Probabilistic Primitive Refinement Algorithm for Colored Point Cloud Data

Johan Ekekrantz[1], Akshaya Thippur[1], John Folkesson[1] and Patric Jensfelt[1]

***Abstract*—In this work we present the Probabilistic Primitive Refinement (PPR) algorithm, an iterative method for accurately determining the inliers of an estimated primitive (such as planes and spheres) parametrization in an unorganized, noisy point cloud. The measurement noise of the points belonging to the proposed primitive surface are modelled using a Gaussian distribution and the measurements of extraneous points to the proposed surface are modelled as a histogram. Given these models, the probability that a measurement originated from the proposed surface model can be computed. Our novel technique to model the noisy surface from the measurement data does not require a priori given parameters for the sensor noise model. The absence of sensitive parameters selection is a strength of our method. Using the geometric information obtained from such an estimate the algorithm then builds a color-based model for the surface, further boosting the accuracy of the segmentation. If used iteratively the PPR algorithm can be seen as a variation of the popular mean-shift algorithm with an adaptive stochastic kernel function.**

I. INTRODUCTION

Robots can capture 3D point cloud data using several types of sensors such as laser range finders, stereo cameras and structured light sensors. When combined with a calibrated camera, these 3D sensors can produce colored 3D point clouds which provide an informative visual snapshot of the environment. This combination of sensors is often called a RGB-D sensor. Popularized by the release of the Microsoft Kinect sensor back in late 2010, RGB-D sensors have received a large interest by the computer vision and robotics communities.

Man made environments contain large amounts of information in the form of easily parametrized surfaces such as planes, spheres and cylinders. Detection, estimation and segmentation of these surfaces can be used to solve many robotic problems, such as object segmentation, simultaneous localization and mapping (SLAM), navigation and planning. Hence, despite the noisy observations obtained from the sensors, we persistently try to autonomously extract such surface information from our environments. For vision based tasks such as object recognition and scene characterizations , the capacity of the robot to differentiate between such geometric primitives and other objects is of high utility. Hence, when dealing with 3D images or maps in the form of point clouds, it becomes beneficial to determine which points of the point cloud belong to the *surface inliers* and differentiate them from *surface outliers*, see Fig.(1).

This assignment is limited by the sensor noise. As a direct result, efforts to improve point cloud segmentation have focused on accurately modelling the measurement noise. Most of these methods have built a priori models of the sensor noise.

[1]The authors are with the Centre for Autonomous System at KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden {ekz,akshaya,johnf,patric}@csc.kth.se

978-1-4673-9163-4/15/$31.00 ©

Fig. 1: An example point cloud with a desktop scene. All the points detected by the PPR technique as desktop surface inliers are highlighted in pink, outliers have their original color. Notice how the object points are avoided from being misclassified as inliers. The typical result from plane extraction without our refinement on this scene is to include the bottom few centimetres of objects as table points.

For parametric surface estimation from point clouds the two widely accepted algorithm categories are the RANdom Sample Consensus (RANSAC) algorithm [1] and the Hough transform [2], [3]. Both these techniques typically require well calibrated, accurate camera sensors and finely hand tuned parameters to provide good surface and inlier estimates. The Mean-shift algorithm [4] is a greedy iterative technique which can be used for surface refinement given an initial estimate. It finds an accurate model by maximizing the number of points close to the surface. It works by iteratively estimating the parameters: performing a weighted re-estimation of the surface parameters, using all the 3D measurements weighted by a characteristic kernel function ω operating on distances of the points from the surface; Inliers can consequently be detected as measurements where ω is larger than some predefined threshold.

In this paper we present the *Probabilistic Primitive Refinement* (PPR) algorithm for accurately estimating the surface parameters and inliers, given a rough initial estimate as provided by e.g., RANSAC or the Hough transform. Our algorithm can be seen as a variation of the Mean-shift algorithm. Assuming that a set of measurements consist of noisy data with surface inliers and outliers, we propose a kernel function ω in which the weight of ω is proportional to the *probability* of a measurement being an inlier. The contributions of this paper lie in the formulation of a novel ω. Rather than using a priori built models of the sensor noise we take another approach in which we look at the evidence from the measured scene to directly build noise models from the measurement data, by leveraging

the insight that observed things in a scene can exist in front of surfaces but not behind them as a simple consequence of occlusion. This means that observable things will be on one side of the underlying surface primitive. We hypothesize that our technique could outperform the above mentioned three categories of techniques by being more robust to sensor noise, less reliant on hand tuned parameters as well as better being able to handle non uniform background noise.

To quantify the probability of a point being an inlier it is important to study the measurement noise of the sensor. The measurement noise depends on many environment specific aspects such as the proximity and angle of the sensor to the surface, lighting conditions, surface material, reflectivity, and texture. Given that all of these properties vary in environments, it is impossible to, a priori, build a precise measurement noise model with optimally tuned parameters. The PPR algorithm solves this problem by dynamically modeling the measurement noise from available data using a modified least squares formulation. By integrating over the measurement noise distribution, PPR can accurately estimate the number of inliers expected for the surface, given a required accuracy. This removes the normal requirement of a user defined thresholding parameter on ω for inlier calculation.

Color information in RGB-D sensor data is a discriminating feature that can be exploited for surface characterization. Assuming accurate calibration, Color information can especially be used at boundary regions where the sharp edges could be drowned in depth measurement noise but can still be classified based on color. The PPR algorithm builds a color-based histogram in order to learn which colors are correlated to likely surface inliers. This is then used to boost the accuracy of the PPR algorithm.

The algorithm is evaluated in three experiments, two using public datasets ([5], [6]), with favourable results. See Sections V, VI and VII for more details.

II. RELATED WORK

Probably the most popular technique for surface estimation from point clouds today is the RANSAC algorithm [1]. The RANSAC algorithm is an iterative, robust estimation technique where a subset of measurements is randomly selected and used to form a hypothesis which is then validated by calculating the support for the hypothesis using the rest of the data. The typical formulation using RANSAC can give the surface with most support by the data of a point cloud.

While flexible, RANSAC hinges on the assumption that it is possible to accurately find the inlier measurements given a proposed solution. As discussed in the introductory section this is far from simple due to measurement noise.

In [7] an efficient RANSAC based algorithm is presented and used for detecting planes, spheres, cylinders, cones and tori. In that paper the performance of RANSAC is increased by using a sampling strategy which samples points using a pyramid, controlling the spatial locality of the samples picked.

In [8] superquadrics were fit to everyday objects such as bowls and cups using RANSAC. Compound objects are created using connectivity. Using compound models the system is able to model complex objects such as cups with handles.

The Hough transform [2] is a voting based algorithm where the surface parameter space is discretized into bins. The data points add support to a bin if the bin's parametrization could generate the data point. The Hough transform is often used in the context of detecting lines but can also be used to detect other shapes. In [9] the Hough transform was extended to work for arbitrary objects described by some predetermined model.

As a result of the abundance of planar surfaces in indoor environments, a large body of work has been spent on detecting and estimating the parameters and inliers of planar surfaces from point clouds.

In [10] multiple variations of the Hough transform were evaluated for plane detection in 3D laser scan data. The Randomized Hough transform [11] was found to provide exceptional results with regards to execution time.

In the work of [12] an efficient plane detector was introduced that relied on an efficient approach to normal clustering through the use of voting. When a cluster of normals was detected the measurements in the cluster could be separated into planes by detecting the clusters of points along the normal axis.

In [13] an efficient plane segmentation was proposed which utilizes the connected component algorithm using a distance metric which can use color, normals and depth information in order to group similar points. The segments are then checked to see if they are planar and if so, refined by growing the segment to nearby points which fit in the plane and are characterized by the correct color, giving good results on organized data.

In [14] a plane detection algorithm based on the principle of region growing is presented. For a found plane a histogram of points along the normal axis of the plane is calculated and the peak values of the histogram are used in order to detect planes. This technique bear a resemblance to our algorithm in that both rely on the assumption that primitives are detectable as peaks in histograms of point to surface distances.

In [15] a primitive segmentation algorithm for organized pointclouds based on region growing is presented. By utilizing the organized structure the authors are able to efficiently calculate surface normals and curvature for the measurement data. Filtering the data using an edge-preserving multilateral filter taking into account color intensity, normal direction and depth measurement.

In [16] it was shown that detection of elipses in organized pointcloud data could efficiently be levareged to detect cylinders, spheres and cones.

Using the generalized Swendsen-Wang sampling MCMC method, [17] fits planar surfaces using color and disparity images by fusing super pixels. This work is well tailored to segmenting RGB-D images because it explicitly models the noise in the disparity map rather than in a generated point cloud. The authors propose that this method could be suitable for other types of primitives as well.

The authors of [18] address a region segmentation problem in video stream data and utilize an *adaptive* Mean-shift Algorithm. They utilize different kernel functions (spherical, product-based, Gaussian etc.) to iteratively estimate a proposed multi modal underlying probability function, seeded by

the histogram of features extracted on the input data. They calculate the classification probability for every pixel in the video frame, based on such probability functions over space (connected regions) and color. These probability functions are built using kernels with adaptive parametrizations which are optimized to suit the instance data the best. Our PPR algorithm is similar to this contribution in a way that we are interested in a 2-class classification problem (inliers or outliers) and utilize a similar concept of finding an underlying inlier probability function based on geometry and color. In addition to processing different types of data for different purposes, we rely on a unimodal Gaussian kernel and employ more sophisticated signal processing engineering to achieve inlier and outlier classification.

III. THE PPR ALGORITHM

The Probabilistic Primitive Refinement (PPR) algorithm considers an initial estimate of a surface parametrization p_{in} of a geometric primitive surface S, a set of measurements (i.e. points in the point cloud) $M = \{m_0, m_1, \ldots, m_n\}$. PPR outputs a refined parametrization p_{out} and a set $K = \{k_0, k_1, \ldots, k_j\} \subset M$ where K are the measurements classified to belong to the surface S in M.

PPR assumes that given K, a parametrization p for S can be accurately estimated. The parametrization p contains valuable information when constructing K. The main principle of this paper is therefore to iteratively refine K and p as described in Algorithm 1. This lies within the bounds of a family of algorithms rooted in the Mean-shift [4] algorithm.

Algorithm 1 Mean-shift-based Surface Estimation

while not convergence **do**
1: Compute surface inliers K given surface parametrization p_{in}
2: Estimate new surface parametrization p_{out} given inliers K
3: Update $p_{in} \leftarrow p_{out}$
end while

Algorithm 2 Probabilistic Primitive Refinement (PPR)

while not convergence **do**
1: Build distance to surface histogram H given surface parametrization p_{in}
2: Fit a scaled Gaussian $\alpha * G$ to approximate H
3: Find color-based models for surface inliers and outliers
4: Calculate the probability $P(inlier)$ of belonging to the surface for each measurement
5: Compute a estimate of surface inliers K from $P(inlier)$
7: Estimate new surface parameters p_{out} using $P(inlier)$
8: Update $p_{in} \leftarrow p_{out}$
end while

While estimation of p given K is straightforward using, for example, least squares optimization, the determination of K given p will be flawed if measurement noise is imprecisely known. The PPR algorithm addresses this aspect of the problem in particular, by calculating the probability $P(m_i = inlier)$ that a measurement m_i is sampled from the surface S i.e. the probability that a particular point in the point cloud is an inlier of the surface.

In order to find $P(m_i = inlier)$, PPR (see Algorithm 2) assumes that the measurement noise is an identically distributed random variable, unbiased and symmetric. In the rest of this paper we use a Gaussian to represent the measurement noise distribution.

An inherent issue of fitting a measurement noise distribution to sensor data for a surface are outlier points that lie near the surface. These points will be asymmetrically distributed, with nearly all being on the side between the sensor and the surface. Due to occlusion it is usually not possible to see two sides of the same surface.

A. Inlier Probability Function

The measurement noise for measurement m_i is assumed to be a zero mean Gaussian as a function of the distance $d(m_i, S)$, where $d(m_i, S)$ is the shortest signed distance from m_i to the surface S i.e. the distance has a sign $(+/-)$ depending on which side of the surface the point is located at. For notational convenience we abbreviate $d(m_i, S)$ as $d(m_i)$.

By constructing a smoothed histogram H of the distances for all for the measurements close ($-\beta \leq d(m_i) < \beta$, where β is a neighbourhood threshold) to the surface S, we can capture the distribution of points which are likely to be classified into K. An example of a histogram H given a typical desktop scene (Fig. 2a) is shown in Fig. 2b.

Given the Gaussian distribution $N(x-\mu, \sigma)$ with σ capturing the noise in the measurements of K and a scaling constant α, controlling the number of inliers in the surface S, we can calculate the inlier probability $P(m_i = inlier|d(m_i))$ as the ratio of expected inliers to number of actual measurements, see Eq. (1). For convenience we abbreviate $P(m_i = inlier|d(m_i))$ as $P(inlier|d(m_i))$.

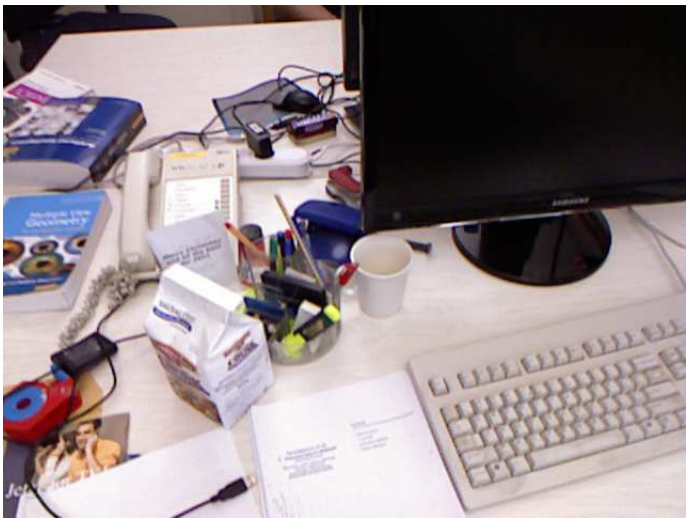
$$P(m_i = inlier|d(m_i)) = \frac{\alpha G(d(m_i), \mu, \sigma)}{H(d(m_i))} \tag{1}$$

B. Estimating Parameters For Geometrical Fit

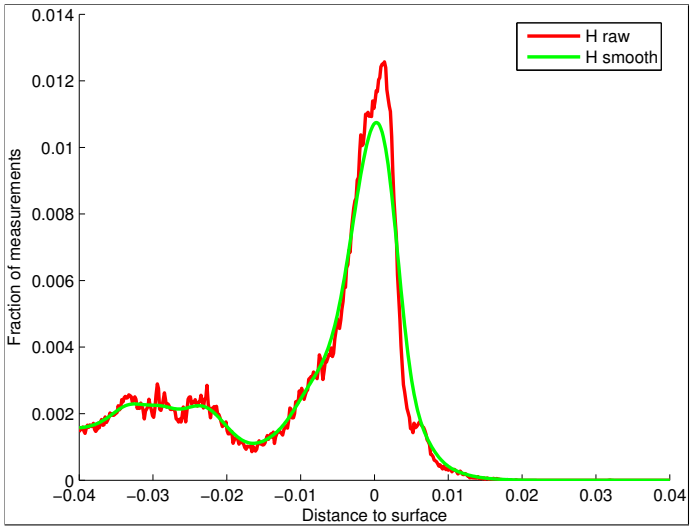
As mentioned in Section I the measurement noise σ for a surface S depends on several properties particular to the environment, sensor, surface etc. This makes it impossible to know a priori with high certainty the σ and α parameter values. PPR solves this by estimating these parameters (viz. μ, σ, α) from data.

The estimate of μ is done by a search for a large peak in the histogram H where $\mu \approx 0$ (i.e. in the neighbourhood of 0). This means that PPR assumes that the surface S is clearly distinguishable in the histogram but that the parametrization p is possibly inaccurate, but not by a large amount. We note that simply taking the mean of H would not be appropriate as the data is skewed by the asymmetric outlier distribution.

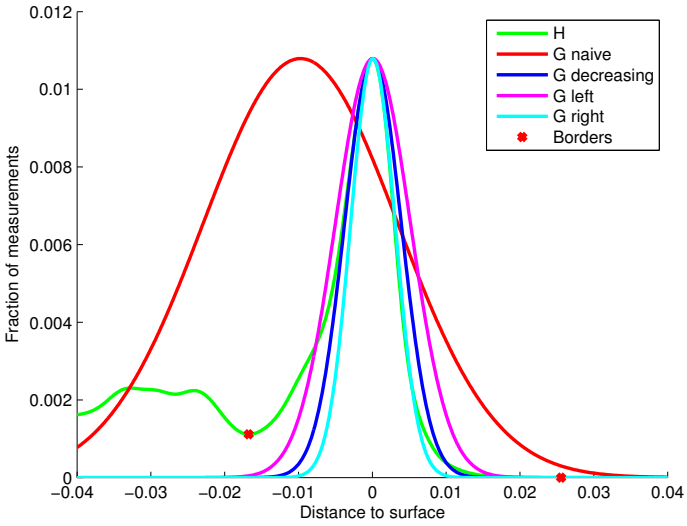
When estimating α, we rely on the assumption that given a perfect parametrization p the inlier measurements of S, dominate the total distribution of measurements in $H(\mu)$. We therefore have $P(inlier|d(m_i) = \mu) = \frac{\alpha G(\mu,\mu,\sigma)}{H(\mu)} = 1$ and can use the value of $\alpha = H(\mu)$. This means that we select α to be such that the probability of a measurement found exactly



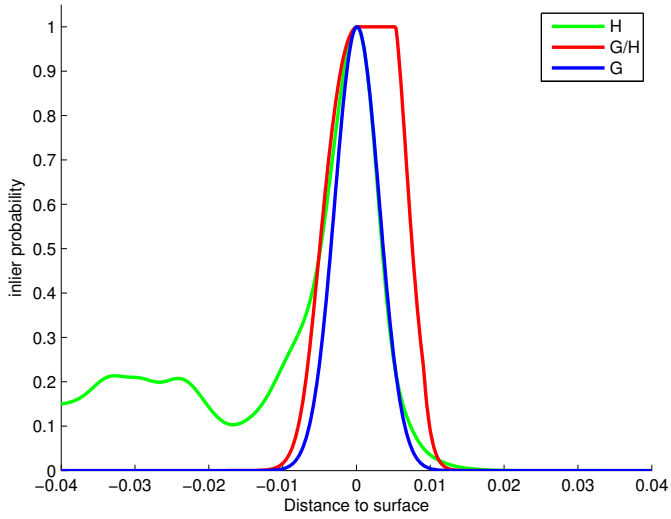
(a)



(b)



(c)



(d)

Fig. 2:
(a) Typical desktop with lots of clutter.
(b) Histogram of distances from estimated planar desk surface in (a). Negative distances are above the desk surface.
(c) Differently fitted Gaussian functions overlaid on the smoothed histogram H of distances from (b) (in green). The red boundary circles represent the first local minima on either side of the histogram peak. *G naive* (red) is calculated using standard mean and variance estimation of H. *G decreasing* (blue) is calculated using least squares fitting on the data between the aforementioned boundaries. *G left* is calculated using least squares fitting on the data between the peak and the left boundary point. *G right* is calculated using least squares fitting on the data between the peak and the right boundary point. Notice how *G right* fits very well to the histogram on the bottom side of the desk where there is no clutter.
(d) Histogram H of distances from estimated planar desk surface in Fig. 2a and found surface measurement noise estimate G normalized so that the peak values equals one. *G/H* shows the probability function for $P(inlier|d(m_i, p))$ where $-0.04m \leq d(m_i, p) < 0.04m$ for a parametrization p found after 15 iterations of PPR.

at the surface to be part of the surface is 1, see Fig. 2c for an example.

The inlier distribution uncertainty σ is due to measurement noise. This noise dominates over the clutter near the surface. We limit the estimation of σ to fit the data about the proposed peak (μ) in the section that is monotonically decreasing from the peak, since the increase after that section is presumably due to clutter becoming dominant. In order to reduce the risk of unwanted local minima caused by sample variance, we slightly smooth the histogram H (as seen in Fig. 2b).

Using least squares, σ can be estimated as $\underset{x}{\operatorname{argmin}} \int_a^b (H(x) - \alpha G(x, \mu, \sigma))^2 \mathrm{d}x$ where the interval $[a, b]$ defines the partitioned region. As can be seen in Fig. 2c, the proposed method leads to considerably better fit of the peak in H than the naive approach of estimating μ and σ as the mean and standard deviation of H.

We know that if the data used to estimate σ contains outliers, the size of σ will be over estimated. We also know that the outliers due to objects on the surface will only be on the side facing the sensor because of occlusion (normal sensors can only see the side of a surface facing the sensor).

In order to increase the accuracy of the estimated model we can therefore estimate σ individually for each side of μ in H for the partitioned region (i.e. for the intervals $[a, \mu]$ and $[\mu, b]$) and finally pick the smaller value of the estimated σ, as that estimate is likely to be influenced by less outliers.

As can be seen in Fig. 2c the measurement noise $Gright$ fits better to the data of H than the other estimations of the measurement noise G as it uses a section of the histogram that contains less outliers for estimating σ. Fig. 2d shows the probability, as defined in Eq. (1), that a measurement is an inlier to the planar desk surface in Fig. 2a. We can easily modify the least squares formulation to prefer solutions where $P(inlier|d(m_i)) \leq 1$

C. Estimating Color Probabilities

In many environments color contains strong descriptive information. Our idea is to use correlation of colors and the geometrical fit, to learn a set of colors that are more likely found among the inliers of the surface than among the outliers. To learn a mapping $P(inlier|color(m_i))$, we can use $P(inlier|d(m_i))$ for the measurements in M where $-\beta \leq d(m_i) < \beta$ as labelled training data in order to train a regressive color model. It is important that the learning algorithm used is robust to incorrectly labelled training data. In this work we use a model in the form of a discrete grid of colors in the HSV color space. For every cell in the grid we calculate the average probability for the measurement data to be inliers. This model is easy to learn and flexible enough to learn multi-modal color models. See Fig. 3b for an example of a learned color model applied to an image.

D. Inlier Inference

For measurements m_i we assume that $P(inlier|m_i)$ depends on two independent probability distributions

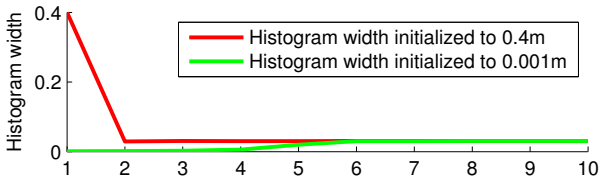


Fig. 4: Experiment showing that given different initial values for the width of the histogram the solution converges to the same value even if the initial estimate is far from the convergence value.

$P(inlier|color(m_i))$ and $P(inlier|d(m_i))$, resulting in Eq. (2).

$$P(inlier|m_i) \propto P(inlier|color(m_i)) \times P(inlier|d(m_i)) \tag{2}$$

The α is related to the total number of predicted inliers, γ, by Eq. (3), where w is the bin width of H. Using γ, K can be constructed by selecting the γ most likely measurements as inliers.

$$\gamma = \int_{-\infty}^{\infty} \alpha G(d(m_i), \mu, \sigma) \mathrm{d}x = \frac{\alpha\sqrt{2\pi\sigma^2}}{w} \tag{3}$$

IV. EXPERIMENTAL SETTINGS

In the experiments that follow, we measure and compare the performances of the PPR algorithm w.r.t. the two types of output produced: the surface parameters p and the set of surface inlier measurements K. On a single core using of an Intel Core i7 2600K processor, normal runtimes for one iteration of the PPR algorithm are 4-5 ms without calculating the color probabilities and 6-7 ms when calculating color probabilities.

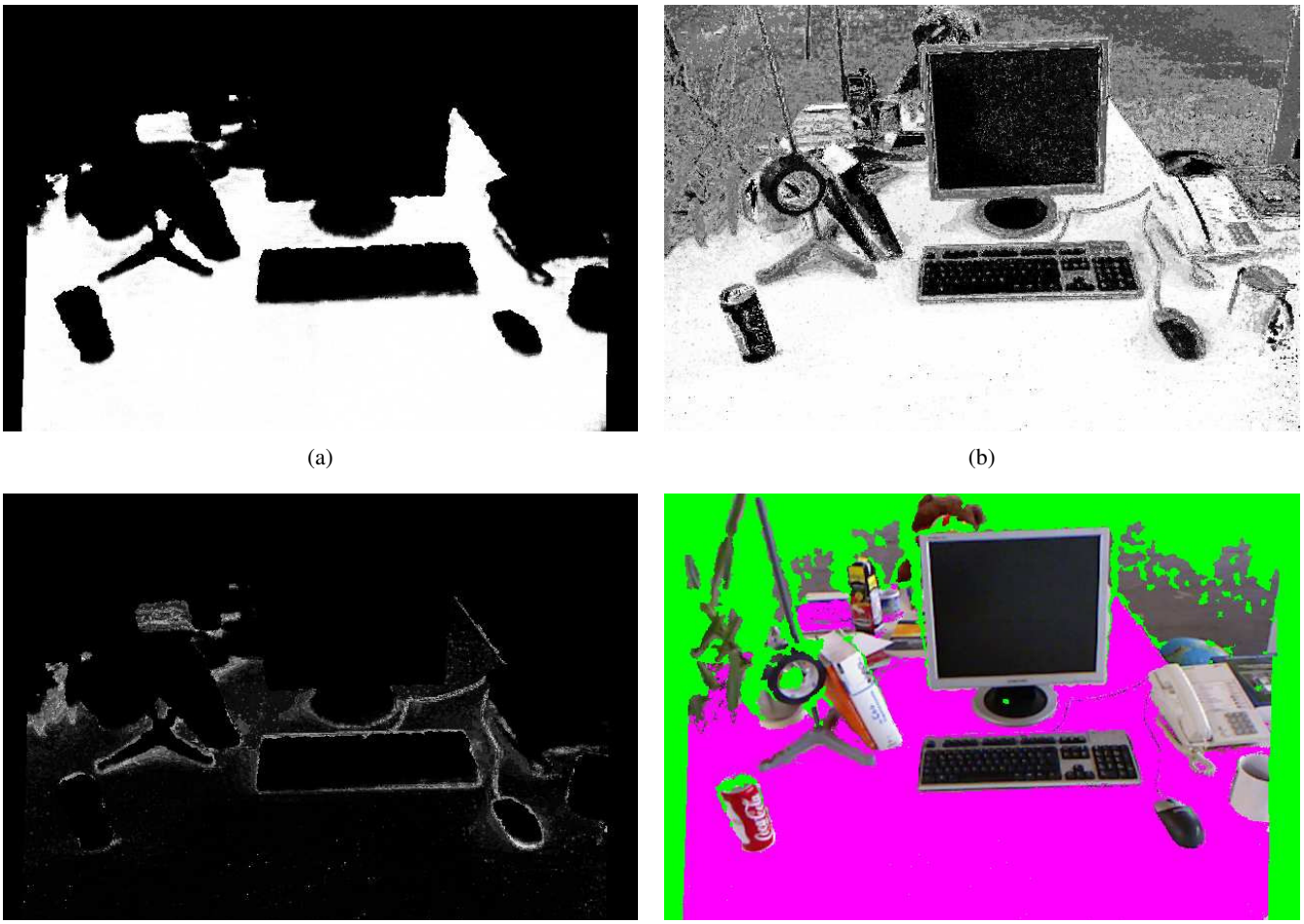
The following subsections provide brief details about the different techniques that are compared and their parameters. Given that some of the experiments will be run on unorganized point clouds and using both spheres and planes we focus our comparisons to RANSAC and Mean-shift which are both able to handle such cases and are widely used today.

A. RANSAC Algorithm

RANSAC proposes a solution by randomly selecting a set of unique measurements from M and calculating the primitive spanned by the selected measurements. The obtained solution is scored based on the number of points within a distance margin δ around the estimated primitive (inlier count). RANSAC is then run multiple times and the solution with the highest score is picked. Using the inliers of the best solution found, the primitive is re-estimated. From experiments we found that $\delta = 0.005m$ provided good results on the benchmarks. RANSAC will be used both for comparison and initialization to PPR.

B. Mean-Shift Algorithm

The Mean-shift algorithm is run for 30 iterations. We define inliers to be measurements where $\omega > 0.5$. A common kernel function is the Gaussian or normal distribution. We found that $\sigma = 0.005m$ provided good results on the benchmarks. Notice how this is equivalent to setting $max\left(H(d(m_i))\right) = 1$,



(a) (b) (c) (d)

Fig. 3: (a) Geometrical probability $P(inlier|geometry)$. (b) Color probability $P(inlier|color)$. Notice how the color model can clearly determine that the red color of the coca cola can does not fit in the gray desktop plane. (c) Difference of the geometrical fit $P(inlier|geometry)$ and joint probability $P(inlier) = P(inlier|geometry) \times P(inlier|color)$. Notice how the difference is largest at the boundaries of objects. (d) Selected inliers using automatic estimation of the number of inliers.

$\alpha = 1$, $\mu = 0$ and a user defined σ set to $0.005m$ when using the PPR algorithm. Mean-shift with a Gaussian kernel can therefore be seen as a subset of the PPR algorithm with no background modelling, no noise estimation and no automatic inlier inference. Mean-shift will be used for comparison to PPR.

C. PPR Algorithm

Using PPR[1] as defined in Section III, the histogram range β is initialized to $[-0.04m, 0.04m]$ and is adaptively updated as $\beta = k \times \sigma$. Here σ is initialized by statistics over the histogram and updated in subsequent iterations according to the least squares formulation. We use $k = 6$ which corresponds to the (6σ) bound. As can be seen in Fig. 4, β quickly converges to the same value even if initialized far from the optimal value. The number of bins in the histogram ϵ is selected as the number of measurements in H divided by 500. The result is constrained to have no less than 10 bins in order to be able to accurately capture the distribution and no more than 500 bins in order to avoid over representing the distribution. This removes the need to supply PPR with finely tuned user defined parameters for β and ϵ. Similar to the Mean-shift algorithm, PPR was run for 30 iterations.

[1]Open source implementation of the PPR algorithm available at: https://github.com/jekekrantz/PPR

V. EXPERIMENT: INLIER ACCURACY

In order to assess the accuracy of the PPR algorithm as compared to RANSAC and Mean-shift we devised an experiment where the task is to segment a set of desktop tables from the objects on top of it.

For the evaluation we use the *KTH-3D-TOTAL* dataset [5]. It contains over 400 desktop scenes in the form of large unorganized, colored point clouds generated by registering multiple frames and fusing the data into one point cloud. The data is captured in office rooms at a robotics research lab and contain real-world desktop scenes with lots of clutter. Given that the scenes are registered automatically the data contain properties

such as slight misalignment and missing data, as would be the case in most real-world robotics scenarios. Also, the final point clouds in the dataset are *unorganized*, i.e. the point clouds have no fixed density, size, resolution or point ordering w.r.t. the sensor. We consider this dataset with its "real-world attributes" to provide a good setting for benchmarking the performances of various plane estimation/refinement techniques.

The objects in the dataset are annotated to an accuracy of approximately 0.001 m. The dataset also contains an initial plane guess given by three points manually selected from the desktop. In the case of RANSAC point sampling was done only on points that were less than 0.05 m from the desktop in order to ensure that the desktop was selected, to increase the fairness of the comparison.

A. Performance Measure

We define the $Error$ for a scene i as the ratio of desktop inliers $||K_{plane}||$ selected from the annotated ground truth object inlier points K_{object} to the total number of object points in $||K_{object}||$ as defined in Eq. (4). In other words: $Error$ is the fraction of object points being miss classified as being part of the desk in a scene. Using this criterion a conservative technique that selects very few points as inliers would perform very well. It is therefore informative to see the average number of selected inliers for the different algorithms in order to ensure that all of the algorithms are approximately equally conservative in their inlier estimation. Given that the objects in the dataset are annotated using bounding boxes we use only the book and keyboard object categories which have cuboidal structures resulting in accurate annotations.

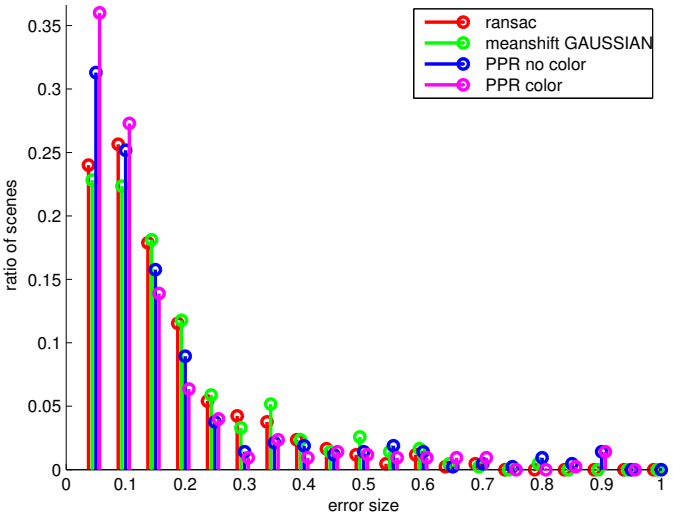
$$Error(i) = \frac{|K_{plane} \cap K_{object}|}{|K_{object}|} \tag{4}$$

B. Experimental Results

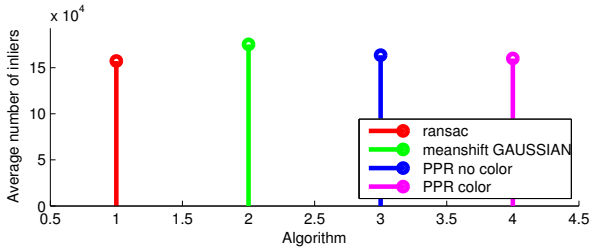
In Fig. 5a the distribution of errors as defined by Eq. (4) for the joint set of keyboards and books in the *KTH-3D-TOTAL* dataset can be seen. The experiments show that the PPR algorithm has a significantly higher number of cases with a very small error (see the leftmost bin of the histogram shown in Fig. 5a) as compared to RANSAC and Mean-shift. Similarly it is also clear that the use of a learned color model has a significant impact on the results. Looking at the average number of plane inliers (as shown in Fig. 5b) we conclude that all of the algorithms select approximately the same number of inliers. From the experiments we conclude that PPR, especially using color information, is better at picking correct surface inliers. We also notice that RANSAC slightly outperforms Mean-shift, this makes sense because RANSAC is less likely to get stuck in local minima, while optimizing on a similar criteria.

VI. EXPERIMENT: SURFACE ACCURACY 1

A key property for any surface segmentation algorithms is the ability to precisely estimate the same surface parameters, independently of where the surface is viewed from. Using five RGB-D video sequences from the dataset [6] we investigate the performance of the PPR algorithm by running a single iteration of refinement to post process the parameters for a set of desktop planes found using RANSAC.



(a) Distribution of errors as defined in Eq. (4) for the joint set of keyboards and books in the dataset.



(b) Number of average selected inliers for the benchmarked algorithms in the dataset. Notice how all four benchmarked algorithms extract approximately the same number of inliers.

Fig. 5: A summary of the results on the KTH-3D-TOTAL dataset.

A. Performance Measure

Using ground truth sensor poses, acquired using a motion capture system, it is possible to transform the plane parameters found in each RGB-D frame into a global coordinate system. For a frame i with normal N_i defined in the local coordinate system and a global camera rotation R_i, the angle between two planes can be found as in Eq. (5). Calculating the average angle between all pair of planes can be used to measure the stability of the estimations.

$$angle(i,j) = cos(R_i^{-1}N_i \cdot R_j^{-1}N_j)^{-1} \tag{5}$$

B. Experimental Results

From table I we can see that even a single iteration of PPR consistently improves the results for the RANSAC algorithm. In sequences *fr1/xyz* and *fr1/rpy* the desktop contains a lot of occluding objects (see Fig. 2a) and fast motions. Accurate measurement noise estimation and background modeling is therefore of key importance for these sequences. Sequences *fr2/xyz*, *fr2/rpy* and *fr2/desk* contain a much cleaner desk (see Fig. 3d) and involves slower camera motions. The improvement using PPR is greater in hard cases whereas in simple cases a finely tuned RANSAC performs adequately.

Algorithm		*fr1/xyz*	*fr1/rpy*	*fr2/xyz*	*fr2/rpy*	*fr2/desk*
RANSAC+PPR	color	**1.35**	6.53	**0.68**	**1.23**	**3.14**
	no color	**1.35**	**6.32**	**0.68**	1.24	3.17
RANSAC	$\sigma = 0.005m$	1.43	6.99	**0.68**	1.24	3.18

TABLE I: Mean angular difference for a set of estimated planes on five image sequences from the dataset presented in [6]. Notice how the application of PPR on the solution found by RANSAC consistently leads to more consistent estimation of plane parameters.

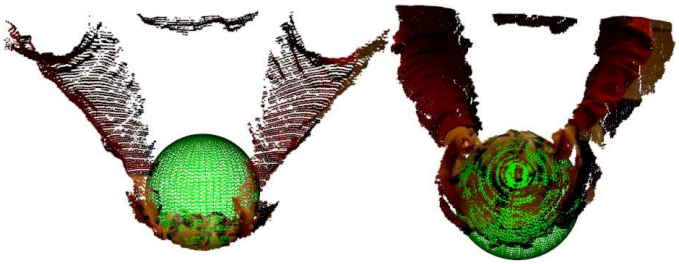


Fig. 6: Two views of a sphere fitted to a plastic ball using PPR. Estimated ball diameter was accurate up to 0.002 m even for noisy data such as in this pointcloud.

VII. EXPERIMENT: SURFACE ACCURACY 2

To test on non-planar primatives, we devised an experiment where the parameters of a sphere is to be estimated. The diameter of the sphere, a plastic soccer ball, was manually measured to 0.175 m. RANSAC is used to get an initial estimate which is then improved by PPR. Results can be seen in Fig. 6. Using 100 images of the soccer ball, the average diameter as estimated by PPR was 0.177 m (average error 0.002 m) with a standard deviation of 0.002 m. For RANSAC the average diameter estimated was 0.197 m (average error 0.022 m, 11 times that of PPR) with a standard deviation of 0.003 m. Studying the data, we notice that this is caused by RANSAC being unable to accurately filter out the hands holding the ball.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper we present a probabilistic primitive refinement algorithm that given an initial geometric primitive estimate and a point cloud can refine the parameters of the primitive and calculate a set of likely surface inliers. Using a novel approach to estimate the measurement noise we can accurately estimate the probability of a measured point as being sampled from the surface. By utilizing inherent properties of the problem and self-tuning, the proposed algorithm is able to avoid using finely tuned sensor specific parameters. Based on three separate experiments we conclude that the proposed algorithm outperforms both RANSAC and a Mean-shift formulation using a Gaussian kernel. By learning a multi modal color model for the surface the accuracy of the inlier estimation can be boosted further. In future work we envision that the use of localized color and noise models could improve accuracy further. We also believe that enforcing locally coherent segmentation could improve the solutions.

ACKNOWLEDGMENT

This work was funded by SSF through its Centre for Autonomous Systems and the EU FP7 project STRANDS (600623).

REFERENCES

[1] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[2] P. V. Hough, "Method and means for recognizing complex patterns," Dec. 18 1962. US Patent 3,069,654.

[3] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Commun. ACM*, vol. 15, pp. 11–15, Jan. 1972.

[4] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *Information Theory, IEEE Transactions on*, vol. 21, no. 1, pp. 32–40, 1975.

[5] A. Thippur, R. Ambrus, G. Agrawal, A. G. del Burgo, J. H. Ramesh, M. K. Jha, M. B. S. S. Akhil, N. B. Shetty, J. Folkesson, and P. Jensfelt, "KTH-3D-TOTAL: A 3D Dataset for Discovering Spatial Structures for Long-Term Autonomous Learning," in *Control Automation Robotics and Vision (ICARCV), IEEE International Conference on*, Dec 2014.

[6] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[7] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for Point-Cloud Shape Detection," in *Computer graphics forum*, vol. 26, pp. 214–226, Wiley Online Library, 2007.

[8] G. Biegelbauer and M. Vincze, "Efficient 3D object detection by fitting superquadrics to range image data for robot's object manipulation," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1086–1091, IEEE, 2007.

[9] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern recognition*, vol. 13, no. 2, pp. 111–122, 1981.

[10] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter, "The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design," *3D Research*, vol. 2, no. 2, pp. 1–13, 2011.

[11] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: randomized Hough transform (RHT)," *Pattern recognition letters*, vol. 11, no. 5, pp. 331–338, 1990.

[12] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using RGB-D cameras," in *RoboCup 2011: Robot Soccer World Cup XV*, pp. 306–317, Springer, 2012.

[13] A. Trevor, S. Gedikli, R. Rusu, and H. Christensen, "Efficient organized point cloud segmentation with connected components," *Semantic Perception Mapping and Exploration (SPME)*, 2013.

[14] D. Hähnel, W. Burgard, and S. Thrun, "Learning compact 3d models of indoor and outdoor environments with a mobile robot," *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 15–27, 2003.

[15] D. Holz and S. Behnke, "Approximate triangulation and region growing for efficient segmentation and smoothing of range images," *Robotics and Autonomous Systems*, vol. 62, no. 9, pp. 1282–1293, 2014.

[16] R. L. Kristiyan Georgiev, Motaz Al-Hami, "Real-time 3D scene description using Spheres, Cones and Cylinders," Nov 2013.

[17] C. Erdogan, M. Paluri, and F. Dellaert, "Planar segmentation of RGBD images using fast linear fitting and markov chain monte carlo," in *Computer and Robot Vision (CRV), 2012 Ninth Conference on*, pp. 32–39, IEEE, 2012.

[18] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, pp. 438–445 vol.1, 2001.