2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

The Reflectance Field Map: Mapping Glass and Specular Surfaces in Dynamic Environments

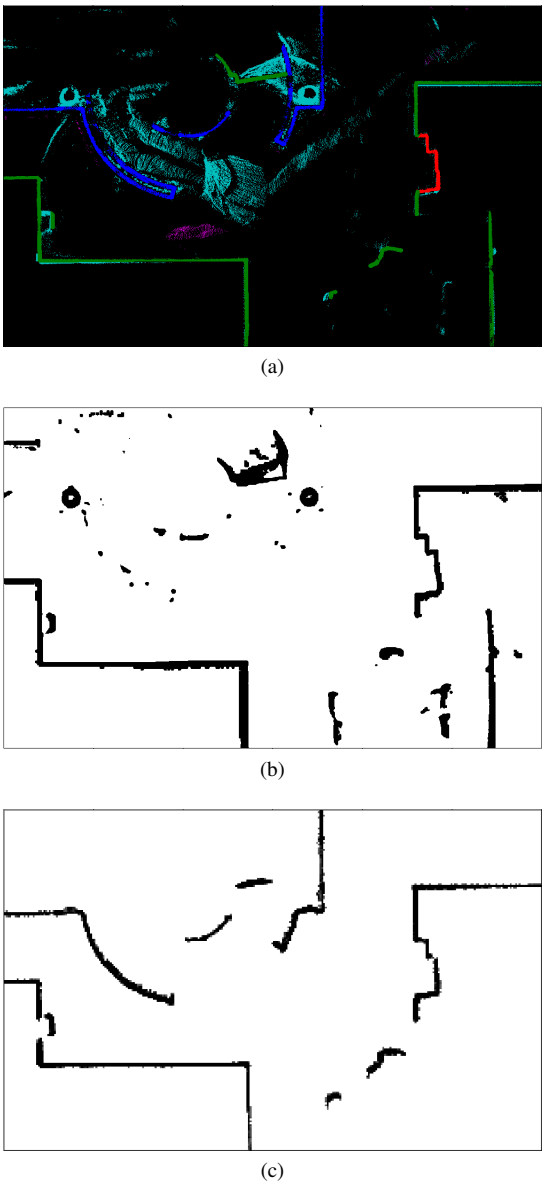
Paul Foster[1], Collin Johnson[2], and Benjamin Kuipers[3]

***Abstract*— We present the Reflectance Field Map, a reliable real-time method for detecting shiny surfaces, like glass, metal, and mirrors, with lidar. The Reflectance Field Map combines the theory developed for Light Field Mapping, common in computer graphics, with occupancy grid mapping. Like early methods for sonar-based robot mapping, we show how the addition of angular viewpoint information to a standard 2D grid map enables robust mapping in the presence of specular reflections. However unlike previous approaches, our method works in dynamic environments. Additionally, unlike recent approaches for lidar-based mapping of specular surfaces, our approach is sensor-agnostic and has no reliance on either intensity or multi-return measurements. We demonstrate the ability of the Reflectance Field Map to accurately map a campus environment containing numerous pedestrians and significant plate glass, both straight and curved. The algorithm runs in real-time (75+Hz) on a single core of a standard desktop processor. An open source implementation of the algorithm is available at `https://github.com/collinej/reflectance_field_map`.**

I. Introduction

Modern man-made environments present a significant difficulty for safe navigation of mobile robots: important barriers that are seemingly invisible to most sensors, including lidar. Plate glass, both flat and curved, is a common architectural element, and mirrors and reflective metal surfaces like elevator doors raise similar issues. Mapping these specular surfaces would be intractable, except for one situation: when the incidence angle of the laser ray is very close to the surface normal, sufficient light is reflected back to the sensor to produce a reliable range measurement. However, almost all current mapping approaches treat lidar observations as independent of observation direction. Since lidar light bouncing off of glass only returns to the sensor from a small number of directions, observations detecting the glass are vastly outnumbered by those missing it.

A small but growing number of techniques have been proposed for robotic mapping of glass and mirrors, but so far none have proved reliable for realtime use by a mobile robot across the variety of common environments. We argue that this is because current techniques are overfitted to ideal models, expecting static environments, bright and consistent specular reflections, and/or flat surfaces. Real surfaces may be clean or covered in dirt. They may be flat, curved or cornered. They may have coatings from IR absorbers to glitter to retroreflectors. A successful mapping system should be able to represent this diversity of appearance, and then make inferences about it.



(a) (b) (c)

Fig. 1. (a) A ground truth map of our scene is shown with walls of diffuse paint (green), specular glass (blue), and shiny metal (red); and distrators such as moving people (magenta), and transmitted/reflected/scattered rays ("reflections", in cyan). (b) A standard occupancy grid is unable to infer the location of glass walls surrounding an atrium, making it dangerous and unusable for navigating this environment. (c) Our method seamlessly recovers the walls while removing the distractors by considering the appearance from all angles using our proposed Reflectance Field Mapping.

[1]Paul Foster is with Cruise Automation, San Francisco, CA, USA, pfost@umich.edu
[2]Collin Johnson is with May Mobility, Ann Arbor, MI, USA, collinej@umich.edu
[3]Benjamin Kuipers is with Computer Science and Engineering, University of Michigan, Ann Arbor, MI, USA, kuipers@umich.edu

Despite a relative lack of attention in robotics, the computer graphics and view synthesis communities have long recognized that the appearance of glass and other specular (shiny) surfaces are fundamentally related to the view direction. We present a new algorithm that detects glass and other shiny surfaces without fitting a particular physical model, treating the robotic mapping problem as a version of view synthesis. Our system's core representation is what we call a Reflectance Field Map (RFM), in analogy to light fields (also referred to as radiance fields), which have recently become popular in computer rendering [1], [2].

The RFM is a recording of the appearance of every point in space from every direction it has ever been viewed from, stored as a simple lookup table. This representation allows us to consider the appearance of a point from all known directions when determining whether it is an obstacle. Critically, since inferences about obstacles are separated from inferences about appearance, new information allows us to quickly change our classification of obstacles (e.g. from static to dynamic) without invalidating our appearance model. We gracefully handle mapping the entire continuum of specular and diffuse surfaces, allowing us to treat such diverse materials as cloth, metal, mirrors, glass, and paint using the same algorithm. Unlike the analogous radiance fields in camera view synthesis, a reflectance field for lidar is relatively low dimensional and fast to compute. Additionally, as lidars are actively illuminated sensors, the representation is stable over time and invariant to changing lighting conditions.

The contributions of this paper are:

- a new approach for lidar-based robotic mapping, the Reflectance Field Map, that allows for robust mapping in the presence of specular surfaces.
- an algorithm for distinguishing specular surfaces from dynamic obstacles in the Reflectance Field Map to support real-time operation in both highly dynamic and highly specular environments.

II. Related Work

Recent work has focused on modeling how a lidar measures specular surfaces to attempt to either filter reflections from the point cloud or to identify specular surfaces so they are not eliminated from the map by the preponderance of evidence pointing to no object existing at that location. Koch et al. use multi-echo returns [3] and intensity data [4] to develop filters that identify planar specular surfaces, including the ability to distinguish transparent from reflective surfaces. In addition to pre-filtering, they include a post-filtering step to reprocess all scans to identify reflections off specular surfaces that were detected after passing by the surface normal. Zhao et al. [5] use intensity data and dual return for a 3D lidar to find planar glass surfaces. Additional processing allows them to project reflections into the map, providing the ability to see behind objects.

Wang et al. [6] use a threshold-based intensity classifier to identify likely glass cells in the map. Their approach simply adds these measurements back into the final map. Jiang et al. [7] use a more probabilistic approach to classification and train a neural net classifier to compute a probability of glass as a function of distance, intensity, and incidence angle. If a cell is more likely to be glass ($p_{glass} > 0.5$)), then a different hit/miss threshold is used to estimate if a cell is occupied to increase the likelihood that glass cells are correctly identified in the map.

Our prior algorithm for mapping glass, VisAGGE [8], can be viewed as a specialization of the more general RFM (Sec. III). VisAGGE estimates only the largest mode of the RFM at each location. However, the reliance on a single reflection peak makes the method unable to represent a multimodal reflectance peak. Additionally, VisAGGE required 360° lidar coverage for best performance. The RFM naturally handles arbitrary numbers of sensors and partial scans, while also being simpler to implement.

Concurrent with the robot mapping community working to interpret sensor data, the computer graphics community has been working to generate increasingly realistic images of scenes. Our method is most closely related to the idea of *light fields* from computer rendering [1], recently made popular in the view synthesis and visual SLAM community as *radiance fields*. Debevec et al. [2] introduce the concept of a reflectance field, which is defined as a function mapping incoming light on a surface to its reflected (outgoing) light. Though developed independently, this definition is analogous to our definition of the reflectance field for lidar.

Recently, methods such as Block-NeRF [9] build a map that represents an encoding of the appearance of every point in a scene from every possible direction. These methods are quite slow and compute heavy, taking several seconds per image, and also must be recomputed for different lighting, but create an unparalleled representation of the scene. The *reflectance field*, our lidar analog of a light field, is much simpler than the camera equivalent due to the self-illuminated nature of the sensor. Inspired by occupancy grids, we further simplify the representation by estimating the reflectance field as a binary field. These changes make our representation far faster and simpler to operate on than their visual equivalents.

Historically, some of the most challenging specular situations were encountered by sonar sensors. Sonar was the most widely used sensor during the development of many standard robotic mapping algorithms, foremost the occupancy grid [10]. However, sonar has a long wavelength, so nearly all surfaces act like mirrors, creating significant issues for reliable mapping.

To address specular reflections, Konolige's MURIEL algorithm [11] introduced a key insight: the probability of a specular reflection is highly dependent on the local surface properties for a grid cell. Pose buckets were introduced that stored the viewing angle and range for the measurements incorporated in the map. Though used for avoiding redundant measurements rather than for understanding specularity, the pose buckets approach can be viewed as a low-cost approximation of our reflectance field. However, by discarding measurements, MURIEL was limited to static

environments.

In later work using short-range lidar, Bennewitz et al. [12] use the lidar's viewing angle information to estimate a probability of reflectance for grid cells already identified as occupied by a standard occupancy grid mapping algorithm. However, a key assumption of their method fails to hold in general. They assume that a failure to observe the surface at a cell results in a max range result, which allows for easily distinguishing between a ray passing through a cell and a ray providing no information. For long-range lidar though, specular reflections often result in multipath returns. Consequently, the initial occupancy test in [12] will erase some specular surfaces.

In contrast to prior work with enhanced map representations, the RFM computes a log-odds of reflectance for every (x, y, θ) cell. Computing log-odds, rather than discarding additional measurements, is required for dynamic environments because the obstacle that was detected at a previous time may no longer exist, leading to permanent false positives in the map. Furthermore, the RFM allows computing an initial probability of occupancy that includes the viewing angle, so a specular surface with a very narrow but highly repeatable angle of incidence for detection is accurately detected as an obstacle.

III. The Reflectance Field Map

Light only changes direction when it hits an object's surface, at which point it will typically be reflected and transmitted in many directions, as well as partially absorbed. Every material surface can be characterized by a function $f(w_i, w_o)$ expressing how much light is emitted in any direction w_o for a given direction of incoming light w_i. This function is called the Bidirectional Scattering Distribution Function [13]. Often, the transmitted and absorbed parts are ignored, giving the Bidirectional Reflectance Distribution Function (BRDF) [14].

Since any point in space may have a surface at it, we need to consider the set of BRDFs throughout space, which, following [2], we define as the *reflectance field*, R.

$$R = R(x, y, z, w_i, w_o) = R(x, y, z, \theta_i, \phi_i, \theta_o, \phi_o) \quad (1)$$

In general, the field R is a 7D function, specified for every 2D input direction $w_i = (\theta_i, \phi_i)$, 2D output direction $w_o = (\theta_o, \phi_o)$, and 3D position (x, y, z). However, the following simplifications and assumptions allow us to use a simpler representation.

First, we only consider 2D position (x, y) because a lidar for a ground robot moves in a 2D plane. Second, we only consider light traveling in the horizontal plane ($\phi_i = 0, \phi_o = 0$) reducing the directions to 1D. Last, since a lidar only sees light that returns back along the same direction as the ray, we only consider the case where the incoming and outgoing directions are in exactly the opposite direction $\theta_i = \theta_o + \pi$.

Thus, we can approximate the full 7D function using a simpler 3D reflectance field:

$$\hat{R} = R(x, y, \theta_i) \quad (2)$$

A. Constructing the Reflectance Field Map

Intuitively, we might consider measuring $\hat{R}$ in units of percent of the light reflected ("normalized intensity"), but it turns out that this is a very unstable metric for specular surfaces. A lidar beam impinging on a specular surface has a very bright central spot, surrounded by much dimmer scatter. A tiny out-of-plane rotation (tilt) of the lidar or surface often causes the reflected central spot to miss the sensor, but a much larger tilt is needed before the scatter region misses the sensor entirely. Thus, a more repeatable measure than asking "What is the percent reflectance?" is to ask "Is there any detectable reflectance at all?" In other words, we should model $\hat{R}$ as a collection of binary random variables.

We approximate the continuous field $\hat{R}$ as a 3D discretized grid $\hat{\mathbf{R}}_i = \{\hat{R}_i\}$ with uniform cells of size Δ_{xy} in the x-y plane and Δ_θ in the range $\theta = (0, 2\pi]$, which we call the *reflectance field map* (RFM) Given a trajectory $\mathbf{X}^T$ and a sequence of measurements $\mathbf{Z}^T$, we estimate the distribution $p(\hat{\mathbf{R}}_i|\mathbf{Z}^T, \mathbf{X}^T)$ by treating each cell R_i as independent and estimating the state $p(R_i = 1|\mathbf{Z}^T, \mathbf{X}^T)$ using a binary Bayes filter [15]. The remainder of the paper reduces $p(R_i = 1)$ to $p(R_i)$ for clarity.

$$\begin{aligned} p(\hat{R}|\mathbf{Z}^T, \mathbf{X}^T) &= \eta p(\mathbf{Z}^T|\hat{R}, \mathbf{X}^T) p(\hat{R}|\mathbf{X}^T) \\ &\approx \eta \prod_i p(\mathbf{Z}^T|\hat{R}_i, \mathbf{X}^T) p(\hat{R}_i|\mathbf{X}^T) \end{aligned} \quad (3)$$

Our implementation uses a uniform prior $p(\hat{R}_i|\mathbf{X}^T) = 0.5$. We note that (3) is identical to the standard 2D occupancy grid [15], except in a 3D (x, y, θ) space, grid rather than 2D Cartesian space.

Notice that by this definition, an occupancy grid map can be interpreted as a reflectance field map by setting $\Delta_\theta = 2\pi$ and treating occupancy as synonymous with reflectivity. In an occupancy grid map, every point in space is considered either occupied and reflective from all directions, or free and non-reflective. A cell of the map is assumed to not reflect light with high probability if and only if it is empty.

We accumulate evidence for the RFM using the raycasting approach described in Alg. 1. The key is that each ray updates a (potentially) different slice of the x-y plane along the measurement direction θ_r. By estimating independent reflectances in every direction, we avoid conflating the concepts of reflectivity and occupancy. Instead, we proceed by first collecting evidence of reflectance, from which we later infer occupancy.

The reflectance field map represents the probability of a lidar ray to be reflected at a given location and direction in the environment. Because lidar reflections occur where there are objects, locations with a high probability of reflectance are highly likely to correspond to obstacles in the environment. Thus, we can construct an occupancy grid from an RFM by applying a function to each (x, y) that computes a probability of occupancy, given the probability of reflectance:

$$p(c_{x,y}) = f_{\hat{R}}(\hat{R}(x, y, \theta = 0), \ldots, \hat{R}(x, y, 2\pi - \Delta_\theta)) \quad (4)$$

Algorithm 1 Accumulating RFM Evidence

Input: R_{hit} : a 3D grid storing number of rays impinging on each cell

Input: R_{miss} : a 3D grid storing number of rays passing through each cell

// All x, y, θ are in units of the cell size $(\Delta_x, \Delta_y, \Delta_\theta)$

for all $r \in rays$ **do**

  // r has origin (x_r, y_r), distance D_r, and heading θ_r

  $[u_x, u_y] \leftarrow [\cos(\theta_r \Delta_\theta), \sin(\theta_r \Delta_\theta)]$

  for $d = 0$ **to** $D_r - 1$ **do**

    $[x_d, y_d] \leftarrow [x_r + du_x, y_r + du_y]$

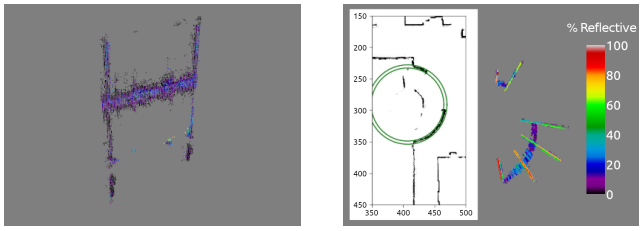
    $R_{miss}(x_d, y_d, \theta) \leftarrow R_{miss}(x_d, y_d, \theta) + 1$

  end for

  $[x_D, y_D] \leftarrow [x_r + D_r u_x, y_r + D_r u_y]$

  $R_{hit}(x_D, y_D, \theta) \leftarrow R_{hit}(x_D, y_D, \theta) + 1$

end for



(a) (b)

Fig. 2. (a) The Reflectance Field Map from a freestanding glass wall. Notice the distinctive H structure of reflective cells along the θ (vertical in image) axis. The edges of the glass are scattering and retroreflective, and thus are visible from nearly all directions, creating the sides of the H. *Unlike intensity-based features, this core H structure is highly repeatable, and is robust to typical out-of-plane rotations caused by bumps and suspension loading in mobile robot applications.* (b) The H figure reliably appears at the edges of curved glass walls. Shown is a cutout of the RFM along a spiral staircase (left) (see inset), seen in extreme perspective to display the H pattern in different directions (right). The gap in the pattern is the entrance to the stairwell, and is correctly identified as freespace.

In practice, a simple counting approach is sufficient for (4). If the count of directions where a cell is probable to reflect is greater than some threshold, we consider the cell an obstacle:

$$f_{\hat{R}}(x,y) = \sum_{\Theta=0}^{2\pi} [\Theta \in p(\hat{R}(x,y,\Theta)|\mathbf{Z}^T, \mathbf{X}^T) > 0.5]\Delta_\theta > \alpha \quad (5)$$

α is in units of radians, so we only consider obstacles visible from a large enough range. We set $\alpha = 1°$ for our results.

The RFM creates a good representation of the reflectance field for static environments. Following a similar result for light fields [1], the observed part of the reflectance field is independent of the sensor path as long as it is outside the convex hull of the region of interest. In practice, this means that driving topologically similar paths through the environment produces identical RFMs, making them good, stable input features for environment modeling.

IV. Solving the Motion Problem

A. The Motion Problem

The RFM is capable of mapping any environment. Over repeated observations, the RFM will converge to the static reflectance field. However, in the presence of motion, temporary obstructions can appear in the map, that would not appear in a standard 2D occupancy grid. Such obstructions persist because only a very small amount of time is generally spent observing each cell, much less than in an equivalent occupancy grid.[1].

In the common scenario where a lidar sensor passes a point in a straight line, it will only observe that point from any given angle once. The intensity may also be similar to a diffuse wall. Consequently, there is no robust way to tell if a brief set of observations at that point are due to a moving object, or just a highly specular one. In fact we assert that, for a single sensor travelling at constant velocity, it is impossible to distinguish glass and motion reliably by looking at the lidar measurements of a single point in space. We call this the *local ambiguity problem.*

[1]We are measuring a 3D grid of cells from the same data that would measure a 2D occupancy grid, so naturally much less data affects each cell.

B. Resolving Local Motion Ambiguities

Our solution to the local ambiguity problem finds a non-local feature that is: (a) highly discriminative between motion and glass, (b) robust to varying levels of curvature and transparency, and (c) robust to out-of-plane tilt caused by bumps or suspension loading. To understand our solution, it is first helpful to consider the appearance of a moving object in the RFM compared to a true pane of glass.

As shown in Fig. 2, if we plot the reflective cells from an RFM taken near a pane of glass in 3D(x, y, θ), we see a distinctive H shape, with the sides of the H aligned to the θ axis. In contrast, as shown in Fig. 3, a moving object appears as a single relatively thin stroke in 3D. The stroke has no intersections with other strokes, and does not travel parallel to the θ axis unless the object temporarily stops.

The H feature always has two common components when plotted with the θ axis vertical:

1) Strong vertical lines caused by the scattering at the edge of the glass, or its frame if mounted in another material. We refer to these as *widely visible locations*.
2) An approximately horizontal stroke corresponding to the surface normal of the glass face. The thickness of this line varies depending on if the glass is dirty or doesn't have much behind it. Crucially, the surface normal cells are connected in (x, y, θ) space, even for curved glass (see Fig. 2b).

The distinctive H shape is still observed even if a glass pane or mirror has a frame or is inset into another wall because, instead of the edges of the glass, the frame or surrounding wall provides the sides of the H. These widely visible locations are the critical distinguishing feature that is present at the edge of *every* piece of glass, but not on the path of a moving object.

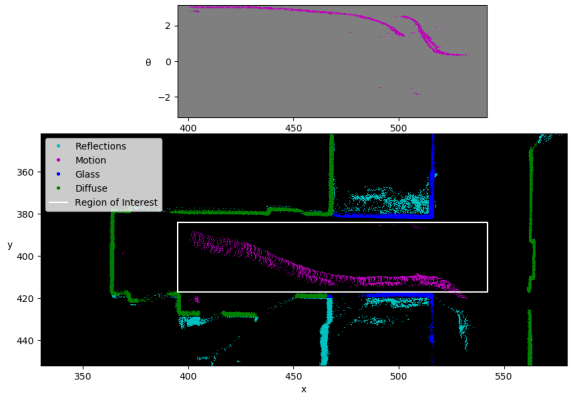


Fig. 3. A moving object in the RFM appears as a thin stroke in θ space in the RFM that is readily distinguishable from the H shape in θ formed by a static specular surface (Fig. 2). In particular, the sides of the H are aligned with the θ axis. However, an object can only create a similar shape by not moving. As soon as the object moves, it will begin creating a stroke perpendicular to the θ axis. In this example we see the RFM near a pedestrian moving along a hall, while our robot passes by. Since we have two lidars, the pedestrian actually splits into two thin strokes, as the lidars do not observe the person from the same viewing angle.

We define a widely visible cell as one with high odds of being reflective from any direction. To compute the odds, we marginalize out θ from the odds-likelihood distribution [11] for location (x, y):

$$O(\hat{R} = 1|\mathbf{Z}^T, \mathbf{X}^T) = \frac{p(\mathbf{Z}^T|\hat{R}_i = 1, \mathbf{X}^T)}{p(\mathbf{Z}^T|\hat{R}_i = 0, \mathbf{X}^T)} O(\hat{R}_i = 1|\mathbf{X}^T) \quad (6)$$

$$O(\hat{R}_i(x, y) = 1|\mathbf{Z}^T, \mathbf{X}^T) = \sum_{\Theta=0}^{2\pi} [O(\hat{R}(x, y, \theta = \Theta) = 1|\mathbf{Z}^T, \mathbf{X}^T)\Delta_\theta)] \quad (7)$$

where $O(\hat{R}_i = 1|\mathbf{X}^T)$ is a prior that an unobserved cell (x, y, θ) is reflective.

Using (7), we can define a widely visible cell $v_{x,y}$ as a cell with high odds of being reflective, $O(\hat{R}_i(x, y)|\mathbf{Z}^T, \mathbf{X}^T) > \beta$.

These observations immediately suggest an algorithm to distinguish glass and motion:

Algorithm 2 Motion Removal

1) Find the set of widely visible (x, y) locations in the map $\{v_{x,y} \in V\}$ per (7).
2) Find the connected components $\mathbf{C}$ of $\hat{R}$ in (x, y, θ) space.
3) For each $C \in \mathbf{C}$:
   a) If $C \cap V \neq \emptyset$, mark cells in $C - V$ as glassy.
   b) If $C \cap V = \emptyset$, mark cells in C as motion.

V. Removing Reflections

Specular reflections can create persistent virtual images in the RFM. While reflections out windows only create visual clutter, reflections off interior glass can create false obstacles that block otherwise navigable portions of the environment.

Koch et al. [3] provide an algorithm to trigger reprocessing old scans for reflection removal after the robot has passed a glass surface. They use the endpoints of lines found by RANSAC, but their method works for any algorithm that can identify glass surfaces, which are found during our motion removal algorithm.

Algorithm 3 Reflection Removal

1) Identify glassy surfaces using Alg. 2.
2) Reprocess all scan rays to build a new reflection-removed RFM $\hat{R}'$:
   a) If a ray passes through glass in $\hat{R}$, discard evidence beyond the glass.
   b) Otherwise, add to $\hat{R}'$ per Alg. 1.

As shown in Sec. VI, reflections are reliably identified and removed by Alg. 3. However, the output is a new RFM, $\hat{R}'$. Since reflections are identified based on the contents of the RFM, Alg. 3 can be run iteratively until it converges and no additional reflections are found. In practice, we haven't found meaningful differences beyond the first iteration. In at least one rare instance, the reflection removal failed to converge in an environment with two parallel glass panes that would cause rays impinging on the other surface to be removed, so each iteration would remove one wall and add the other.

VI. Evaluation

We evaluate the RFM in a challenging environment at the University of Michigan. The environment contains a wide variety of glass, including flat, curved, and stained, as well as shiny metal, and many pedestrians. As shown in Table I, approximately half of the rays in the dataset go through or reflect off a shiny surface. We collected the data using our intelligent wheelchair, Vulcan, equipped with two Hokuyo-UTM-30LX lidars, an IMU, and wheel encoders. The wheelchair was operating autonomously during data collection.

A. Map Accuracy

To evaluate the accuracy of the RFM, we built and hand-labelled a high-resolution ($\Delta_{xy} = 1cm, \Delta_\theta = 0.25°$) map from our data and measured the percentage of glass, metal, and traditional surfaces kept by the RFM, as well as the amount of motion and reflection removed. We baseline our method against a typical occupancy grid, and perform an ablation study to determine the importance of each step of our system. Fig. 1 shows a portion of the evaluation environment, including the labeled ground truth and RFM and occupancy grid maps built from the data.

We treat the problem as a multiclass classification problem, with the goal to remove reflections and motion while keeping static obstacles. We wish to eliminate unreliable observations through glass, but often observations land in the same cell both through glass and directly. To distinguish this situation, we classify *lidar rays*. For the

TABLE I

CONFUSION MATRIX FOR RFM CLASSIFICATION OF LASER RAYS

		Predicted	
	Total	**Kept(%)**	**Removed(%)**
Diffuse	1705612	98.5	1.5
Glass	104478	98.7	1.3
Metal	7049	99.4	0.6
Motion	22439	0.3	99.7
Reflection	2102216	1.8	98.2

TABLE II

CONFUSION MATRIX FOR OCCUPANCY GRID CLASSIFICATION OF LASER RAYS

		Predicted	
	Total	**Kept(%)**	**Removed(%)**
Diffuse	2253828	98.3	1.7
Glass	176619	51.6	48.4
Metal	13470	96.8	3.2
Motion	22439	0	100
Reflection	2102216	24.6	75.4

purposes of unambiguous metrics, we treat both transmitted and reflected rays as "reflections", and define "motion" as any non-reflected rays landing on a transient object. For our results, we treat the different classifications for rays as mutually exclusive. In the case of reflections, a ray can be transmitted through glass and then return an object on the other side of the glass. We report such a ray as a reflection because all objects observed through glass are treated as untrustworthy. For classification of kept surfaces, we only consider rays that are not rejected as reflections. Since the RFM successfully rejects many more reflections, the total number of surface rays considered is less. As a result, there are different total numbers of rays between Table I and Table II.

As shown, in Table III, the simple RFM already increases glass recall capability, but at the expense of some unwanted motion appearing in the map. The addition of motion removal and reflection removal largely eliminates those sources of error, while maintaining high recall on glass and normal diffuse surfaces. Not represented in the table is that many of the rays lost from motion and reflection removal are in the same cells as other rays. For example, motion and reflection filtering decrease ray recall for glass by 1.1%, but only 0.4% of actual glass cells are discarded.

B. Performance

A Python implementation of the RFM algorithm was used for the presented results. Scans of the environment were batch processed. Adding a new scan, consisting of 1081 measurements, takes 1.25ms, removing reflections takes 1.28ms/scan, and motion filtering is run approximately once every 20 seconds and takes 2500ms.

For online SLAM, we use an optimized C++ implementation that maintains an active region centered on the robot where motion filtering and map updates occur. Running on an AMD Ryzen7 3800X processor,

TABLE III

ABLATION STUDY OF RFM OPERATIONS

Method	**Recall Diffuse (%)**	**Recall Glass (%)**	**Motion Removed (%)**	**Reflection Removed (%)**
Occupancy Grid	98.3	51.6	100	75.4
Basic RFM	99.8	99.8	91.8	9.0
+Motion Removal	99.2	99.9	96.5	10.7
+Reflection Removal	98.5	98.7	99.7	98.2

using an active region with 10m radius, and cells with $(\Delta_{xy} = 5cm, \Delta_\theta = 2^\circ)$, adding a new scan takes an average of 1.8ms, motion removal an average of 6.4ms, and the entire RFM update an average of 10.3ms.

VII. CONCLUSION

The RFM makes no attempt to model the complex interactions of a lidar ray with a surface, but instead simply records the information the sensor receives, and then does inference on that information as a second step. There are no planarity assumptions, requirements for multi-echo sensors, or models of intensity data. However, these filtering and classification methods can be useful for the inference step of the mapping process for aiding in motion removal. In particular, classification-based techniques have a very low false positive rate, but a relatively high false negative rate [16]. We can therefore use them as additional information to seed known glass points in our occupancy calculation and inform our motion removal step.

The evaluation of this work used 2D lidar. However, an approximation of the RFM can easily created for 3D scenes. To build the RFM, a virtual 2D scan can be created from the full 3D point cloud by selecting rays from the laser sweep nearest to parallel to the ground plane. This virtual scan can then be used to construct a 2D RFM of the environment, though only structures that are approximately vertical and intersect the virtual 2D scan will be included.

Converting raw sensor data to an RFM creates a feature that is able to express the entirety of a scene's static appearance, is repeatable between multiple runs through the environment, and is computationally cheap, making it a suitable input feature for a variety of perception problems. We believe that future researchers will see the usefulness of the RFM in other applications, including material identification, thin object detection, or as an input to various machine learning techniques.

We believe that Reflectance Field Mapping is a significant advance for robotic mapping and perception in common environments, representing the lidar equivalent of light field techniques. We have shown performance and robustness on the challenging specular surface detection problem sufficient to allow every day operation of autonomous robots in a wide variety of environments.

REFERENCES

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 31–42. [Online]. Available: https://doi.org/10.1145/237170.237199

[2] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar, "Acquiring the reflectance field of a human face," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 145–156.

[3] R. Koch, S. May, P. Murmann, and A. Nüchter, "Identification of transparent and specular reflective material in laser scans to discriminate affected measurements for faultless robotic SLAM," *Robotics and Autonomous Systems*, vol. 87, pp. 296–312, 2017.

[4] R. Koch, S. May, and A. Nüchter, "Effective distinction of transparent and specular reflective objects in point clouds of a multi-echo laser scanner," in *2017 18th International Conference on Advanced Robotics (ICAR)*. IEEE, 2017, pp. 566–571.

[5] X. Zhao, Z. Yang, and S. Schwertfeger, "Mapping with reflection-detection and utilization of reflection in 3d lidar scans," in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 2020, pp. 27–33.

[6] X. Wang and J. Wang, "Detecting glass in simultaneous localisation and mapping," *Robotics and Autonomous Systems*, vol. 88, pp. 97–103, 2017.

[7] J. Jiang, R. Miyagusuku, A. Yamashita, and H. Asama, "Online glass confidence map building using laser rangefinder for mobile robots," *Advanced Robotics*, vol. 34, no. 23, pp. 1506–1521, 2020.

[8] P. Foster, Z. Sun, J. J. Park, and B. Kuipers, "VisAGGE: Visible angle grid for glass environments," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2213–2220.

[9] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-NeRF: Scalable large scene neural view synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 8248–8258.

[10] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[11] K. Konolige, "Improved occupancy grids for map building," *Autonomous Robots*, vol. 4, no. 4, pp. 351–367, 1997.

[12] M. Bennewitz, C. Stachniss, S. Behnke, and W. Burgard, "Utilizing reflection properties of surfaces to improve mobile robot localization," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 4287–4292.

[13] F. O. Bartell, E. L. Dereniak, and W. L. Wolfe, "The theory and measurement of bidirectional reflectance distribution function BRDF and bidirectional transmittance distribution function BTDF," in *Radiation scattering in optical systems*, vol. 257. SPIE, 1981, pp. 154–160.

[14] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis, "Geometrical considerations and nomenclature for reflectance," 1977.

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[16] J. Jiang, R. Miyagusuku, A. Yamashita, and H. Asama, "Glass confidence maps building based on neural networks using laser range-finders for mobile robots," in *2017 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2017, pp. 405–410.