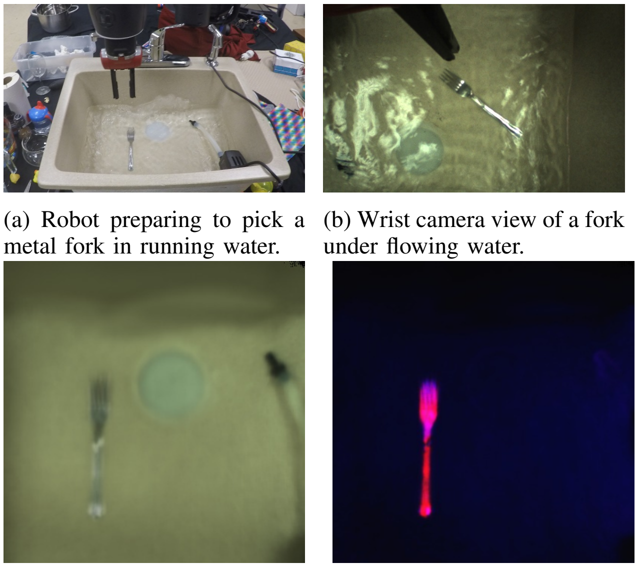
Time-Lapse Light Field Photography for Perceiving Transparent and Reflective Objects

John Oberlin and Stefanie Tellex

***Abstract*—Robust robotic perception and manipulation of household objects requires the ability to detect, localize and manipulate a wide variety of objects, which may be mirror reflective like polished metal, glossy like smooth plastic, or transparent like glass; for example, picking a metal fork out of a sink full of running water or screwing a metal nut onto a bolt. Existing perceptual approaches based on photographs only take into account the average intensity of light arriving at each pixel from one direction, which limits their ability to account for these non-Lambertian scenes. To address this problem, we demonstrate time-lapse light field photography with an eye-in-hand camera of a manipulator robot. An eye-in-hand robot can capture both the intensity of rays, as in a conventional photograph, as well as the direction of the rays. We present a formal model for robotic light-field photography that fits into a probabilistic robotics framework. Using this model, we can synthesize orthographic photographs, remove specular highlights from those photographs, and perform 3D reconstruction with a monocular camera by finding approximate maximum-likelihood estimates. This information can be used to detect, localize and manipulate non-Lambertian objects in non-Lambertian scenes: our approach enables the Baxter robot to pick a shiny metal fork out of a sink filled with running water 24/25 times, as well as to localize objects well enough to screw a nut onto a quarter inch bolt. The techniques in this paper point the way toward new approaches to robotic perception that leverage a robot's ability to move its camera to infer the state of the external world.**



(a) Robot preparing to pick a metal fork in running water.

(b) Wrist camera view of a fork under flowing water.

(c) Rendered orthographic image of the fork; reflections from water and metal are reduced.

(d) Discrepancy view, showing the robot can accurately segment the fork.

Fig. 1: Our approach allows a robot to detect, localize and manipulate under challenging non-Lambertian conditions.

I. Introduction

Many tasks require a robot to detect and manipulate shiny objects, such as washing silverware in a sink full of running water, or assisting a surgeon by picking a metal tool from a metal tray. However existing approaches to object detection struggle with these non-Lambertian objects [6, 21, 3] because the shiny reflections create false images and sharp gradients that change dramatically with camera position, fooling methods that are based on only a single camera image.

Because it can move its camera, a robot can obtain new views of the object, increasing robustness and avoiding difficulties in any single view. To benefit from this technique, the robot must integrate information across multiple observations. One approach is to use feature-based methods on individual images, as in the winning team for the Amazon Picking Challenge [5], but this approach does not incorporate information about the viewing angle and can still struggle with non-Lambertian objects. Other approaches make strong assumptions, such as that an object is rotationally symmetric [21].

An important area of releated work is multi-view stereo, which fuses information from multiple calibrated camera images in order to create a 3D reconstruction of the scene [6]. Multi-view stereo typically uses correspondence of features across frames as the main cue for 3D reconstruction. Our approach, in contrast, uses defocus from a very dense set of images. A fully general approach would combine both cues and probably lead to higher performance. It is worth noting that imaging non-Lambertian objects is still considered an open problem in a recent review paper on multi-view stereo [6].

In this paper, we demonstrate that light field photography [14], or plenoptic photography, is a powerful medium of inference for robotic perception of non-Lambertian objects because it incorporates information from the intensity as well as the angle of the light rays, information which is readily available from a calibrated camera that can collect multiple views of a scene. Light fields naturally capture phenomena such as parallax, specular reflections, and refraction by scene elements, enabling the robot to perceive and manipulate glass or metal objects with these properties. We present a probabilistic model and associated algorithms for turning a calibrated eye-in-hand camera into a time-lapse light field camera that can be used for robotic perception. The contributions of this paper are 1) a probabilistic model of light field photography for a calibrated eye-in-hand camera 2) a framework for calibrating the camera using light field techniques 3) demonstration of using a Baxter robot to produce orthographic synthetic photographs of a scene, extract 3D structure from multiple RGB wrist camera images, and localize and pick both Lambertian and non-Lambertian objects. Notably, our approach enables Baxter to pick a metal fork out of a sink filled with running water 24/25 times, using its wrist camera. Additionally, Baxter can use these approaches to place and tighten a nut on a $0.25''$ bolt. Portions of this work previously appeared in a workshop paper [19].

II. Related Work

Time lapse light field photography has precedent [32, 14], but the movement is typically constrained to a few dimensions. Fixed camera [31] and microlens [7, 18] arrays are stable once calibrated and can capture angular information from many directions simultaneously, but camera arrays are not very portable and microlens arrays do not have a very large baseline. Baxter's arm allows us to densely collect images (in sub millimeter proximity to each other) across large scales (about a meter) over 6 DoF of pose in a 3D volume. The wide baseline enables us to compute an orthographic projection of the scene, where every pixel is rendered as if viewed from directly above. Other datasets using camera gantries have been released, which enable wider baselines [30], but not a widely available piece of equipment such as Baxter. Existing approaches perform depth estimation [29], shape estimation [27] and other computer vision tasks. However we are unaware of a robot being used as a light field capture device for perception and manipulation. Using a 7 DoF arm as a camera gantry allows the robot to dynamically acquire more views and integrate information from multiple views in a probabilistic setting. This approach for object detection and picking is especially useful for non-Lambertian objects.

Phillips et al. [21] described an approach for detecting transparent objects that assumes objects are rotationally symmetric. Smith et al. [25] used a camera array to perform video stabilization using light fields but did not provide a framework for non-Lambertian objects. Rodrigues et al. [23] used a multi-light system to detect and localize transparent objects; our light field approach can benefit from controlling lighting, and opens the door to lighting models for further reducing artifacts and predicting next best views.

Herbst et al. [8] defines a probabilistic surface-based measurement model for an RGB-D camera and uses it to segment objects from the environment. A number of approaches use robots to acquire 3D structure of objects either by moving the camera or moving the object [12, 1, 9, 28, 24, 10, 16, 15, 11, 1]. Our approach, instead, uses a model based around light fields, incorporating both intensity and direction of light rays. This approach can be generalized to IR cameras as well, augmenting approaches such as KinectFusion [17] to handle non-Lambertian surfaces and exploiting complementary information from the IR and RGB channels. Correll et al. [3] review entries to the Amazon Robotics Challenge, a variety of state-of-the-art systems and state that the winning entry had problems due to the reflective metal shelves [5].

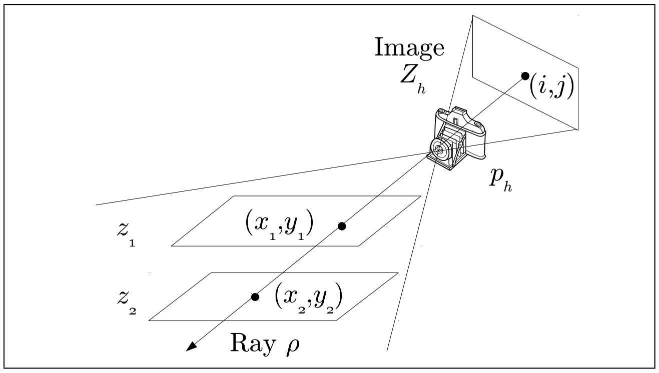


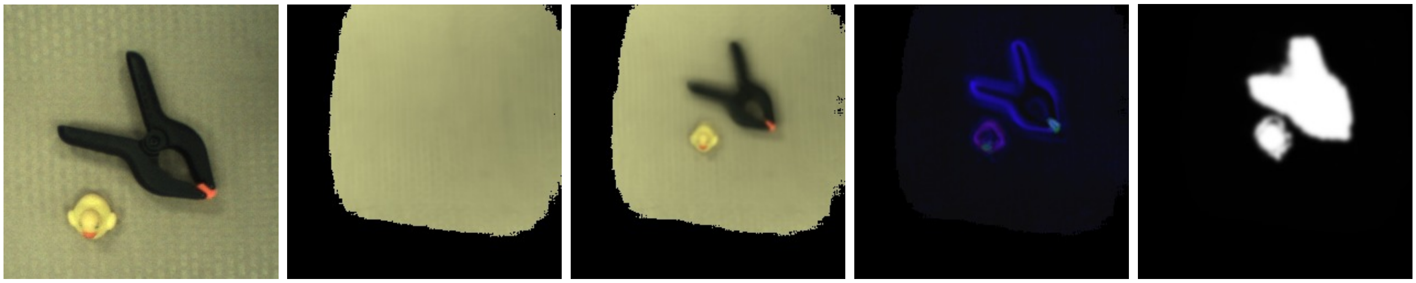
Fig. 2: Diagram of a scene captured by a camera showing our variables. For z_1, ray $\rho \in R_{l_1,w_1}$; for z_2, ray $\rho \in R_{l_2,w_2}$.

A separate body of work automatically acquires 3D structure of the environment from RGB or RGB-D cameras. Notably, LSD Slam [4] uses a pixel-based approach and achieves efficiency using key frames and semi-dense depth maps. Our approach, instead, uses all pixel values from observed images to render a synthetic photograph, which could then be processed with gradients or other steps. When pose is already available, as on Baxter's wrist camera, our approach enables efficient use of all information from the camera, integrating information across many images.

Kutulakos and Seitz [13] presented a theory of space carving. Both their work and ours use dense ray information from multiple perspectives and the standard deviation of intersecting rays to triangulate points in space, but their primary goal is view consistent 3D reconstruction of Lambertian objects and ours is 2D synthetic photography that is robust to non-Lambertian phenomena. It is possible that both works would ultimately develop into probabilistic ray tracers, and in fact, our work may benefit from incorporating space carving potentials into the graphical model to model occlusion.

III. Probabilistic Model for Light Fields

We present a probabilistic model for light field photography using a calibrated eye-in-hand camera. Inference in this model corresponds to finding a model of the light emitted from a scene as well as its 3D structure. We show how to use this model to reduce specular highlights in photographs, as well as demonstrate its ability to localize objects very accurately. We assume the robot is observing the scene from multiple perspectives using a camera, receiving a sequence of images, $Z_0 \dots Z_H$. The robot also receives a pose for each image, $p_0 \dots p_H$, containing camera position and orientation; for example a robotic arm can obtain this information from its forward kinematics. We assume access to a calibration function that defines a ray for each pixel in an image: $\mathcal{C}(i, j, p_h, z) \rightarrow \{(x, y)\}$ which converts pixel i and j coordinates to an x and y in the robot's base coordinate system given a z along the ray, along with its inverse: $\mathcal{C}^{-1}(x, y, p_h, z) \rightarrow (i, j)$. Section III-B describes this function in detail, including how we estimate its parameters for the Baxter robot.



(a) Scene.

(b) Initial map.

(c) Map with objects.

(d) Standard deviation of map with objects.

(e) Discrepancy.

Fig. 3: The mean values of a map m for a scene without objects, and the map after two objects have been added. Blue regions indicate discrepancy between the two maps. The synthetic photographs were made from 590 images taken in a spiral pattern around the objects.

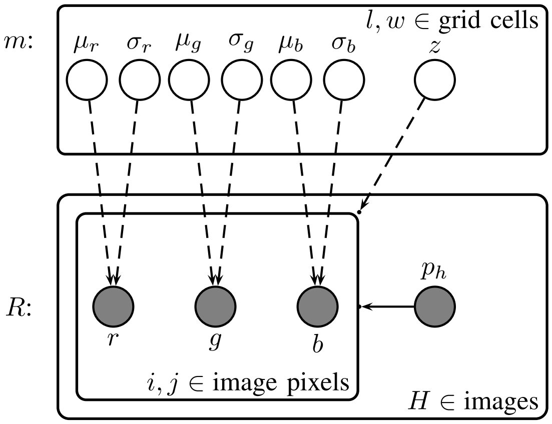


Fig. 4: Bayesian graphical model for our proposed approach. Dashed lines indicate that those edges are only present for a subset of the variables, for the bundle of rays $R_{l,w}$ that correspond to a particular grid cell, defined formally in Equation 4.

The calibration function enables us to define a set of light rays, R, where each ray contains the direction and intensity information from each pixel of each calibrated image. Formally, each $\rho \in R$ consists of an intensity value, (r, g, b)[1], as well as the pose of the camera when the image was taken, p_h, and its pixel coordinate, (i, j). This information enables us to use the ray components to compute an (x, y) coordinate for any height z. Figure 2 shows a diagram of a scene, where light ray ρ intersects different planes.

Next, we define a distribution over the light rays emitted from a scene. Using this generative model, we can infer information about the underlying scene, conditioned on observed light rays, R. We define a synthetic photograph, m, as an $L \times W$ array of cells in a plane in space. Each cell $(l, w) \in m$ has a height z and scatters light at its (x, y, z) location. For convenience, we write that the calibration function C can return either (x, y) coordinates in real space or (l, w) indexes into map cells. We assume each observed light ray arose from a particular cell (l, w), so that the parameters associated with each cell include its height z and a model of the intensity of light emitted from that cell. In order to model the scene, we wish to estimate m given observed light rays R. This estimate can be used to localize objects, extract 3D structure, and other tasks. Formally, we wish to find a maximum likelihood estimate for m given the observed light rays R:

$$\underset{m}{\operatorname{argmax}}\, P(R|m) \tag{1}$$

We assume that each bundle of rays is conditionally independent given the cell parameters. This assumption is violated when a cell actively illuminates neighboring cells (e.g., a lit candle), but enables us to factor the distribution over cells:

$$P(R|m) = \prod_{l,w} P(R_{l,w}|m) \tag{2}$$

Here $R_{l,w} \subset R$ denotes the bundle of rays that arose from map cell (l, w), which can be determined finding all rays that intersect the cell using the calibration function. Formally:

$$R_{l,w} \equiv \{\rho \in R | C(i, j, p_h, z) = (l, w)\} \tag{3}$$

We assume each cell emits light on each channel as a Gaussian over (r, g, b) with mean $\mu_{l,w} = (\mu_r, \mu_g, \mu_b)$ and variance $\sigma^2_{l,w} = \left(\sigma^2_r, \sigma^2_g, \sigma^2_b\right)$ so that:

$$P(R|m) = \prod_{l,w} P(R_{l,w}|\mu_{l,w}, \sigma^2_{l,w}, z_{l,w}). \tag{4}$$

We rewrite the distribution over the bundle of rays, $R_{l,w}$, as a product over individual rays $\rho \in R_{l,w}$:

$$P(R_{l,w}|\mu_{l,w}, \sigma^2_{l,w}, z_{l,w}) = \prod_{\rho \in R_{l,w}} P(\rho|\mu_{l,w}, \sigma^2_{l,w}, z_{l,w}). \tag{5}$$

Next we assume each color channel c in ρ is independent. We use ρ_c to denote the intensity value of ρ for channel c.

$$P(R_{l,w}|\mu_{l,w}, \sigma^2_{l,w}, z_{l,w}) = \prod_{\rho \in R_{l,w}} \prod_{c \in \{r,g,b\}} P(\rho_c|\mu_c, \sigma^2_c, z_{l,w}). \tag{6}$$

[1] We use (r, g, b) in our notation because it is more intuitive; our implementation uses the YCbCr color space.

As a Gaussian:

$$P(\rho_c|\mu_c, \sigma_c^2, z_{l,w}) = \mathcal{N}(\rho_c, \mu_c, \sigma_c^2). \tag{7}$$

Figure 4 shows a Baysian graphical model for our approach. Using this factorization, we can compute the μ_c and σ^2 by computing the sample mean and variance of the rays that intersect each cell. We iterate over z using grid search. We render m as an image by showing the values for $\mu_{l,w}$ as the pixel color; however variance information $\sigma_{l,w}^2$ is also stored. Figure 3 shows a sample wrist camera image, paired with synthetic photographs rendered using this model at fixed z (table height), as well as standard deviation. Edges have higher standard deviation because these images are focused at table height, and some rays strike the table, while others strike the object.

A. Detecting Changes

Once the robot has found an estimate, m for a scene, for example to create a background model, it might want to detect changes in the model after observing the scene again and detecting a ray, ρ' at (l, w). At each cell, (l, w), we define a binary random variable $d_{l,w}$ that is false if the light for that cell arose from background model m, and true if it arose from some other light emitter. Then for each cell we estimate:

$$P(d_{l,w}|m, \rho') = P(d_{l,w}|\mu_{l,w}, \sigma_{l,w}^2, \rho') \tag{8}$$

We rewrite using Bayes' rule, using the joint in the denominator:

$$= \frac{P(\rho'|d_{l,w}, \mu_{l,w}, \sigma_{l,w}^2) \times P(d_{l,w}|\mu_{l,w}, \sigma_{l,w}^2)}{\sum_{d_{l,w} \in \{0,1\}} P(\rho'|d_{l,w}, \mu_{l,w}, \sigma_{l,w}^2) \times P(d_{l,w}|\mu_{l,w}, \sigma_{l,w}^2)} \tag{9}$$

We initially tried a Naive Bayes model, where we assume each color channel is conditionally independent given $d_{l,w}$:

$$= \frac{\prod_{c \in \{r,g,b\}} P(\rho'_c|d_{l,w}, \mu_c, \sigma_c^2) \times P(d_{l,w}|\mu_c, \sigma_c^2)}{\sum_{d_{l,w}} \prod_{c \in \{r,g,b\}} P(\rho'_c|d_{l,w}, \mu_c, \sigma_c^2) \times P(d_{l,w}|\mu_c, \sigma_c^2)} \tag{10}$$

If $d_{l,w}$ is false, we use $P(\rho_c|d_{l,w} = 0, \mu_c, \sigma_c^2) = \frac{1}{255}$; otherwise we use the value from Equation 7. We only use one ray, which is the mean, $\mu'_{l,w}$ of the rays in $R'_{l,w}$. We use a uniform prior so that $P(d_{l,w}|\mu_c, \sigma_c^2) = 0.5$. However this model assumes each color channel is independent and tends to under-estimate the probabilities, as is well-known with Naive Bayes [2]. In particular, this model tends to ignore discrepancy in any single channel, instead requiring at least two channels to be substantially different before triggering. For a more sensitive test, we use a Noisy Or model. First we define variables for each channel, $d_{l,w,c}$, where each variable is a binary indicator based on the single channel c. We rewrite our distribution as a marginal over these indicator variables:

$$\begin{aligned} &P(d_{l,w}|m, \rho') = \\ &\sum_{d_{l,w,r}} \sum_{d_{l,w,g}} \sum_{d_{l,w,b}} \begin{array}{l} P(d_{l,w}|d_{l,w,r}, d_{l,w,g}, d_{l,w,b}) \times \\ P(d_{l,w,r}, d_{l,w,g}, d_{l,w,b}|m, \rho'). \end{array} \end{aligned} \tag{11}$$

We use a Noisy Or model [20] for the inner term:

$$\begin{aligned} &P(d_{l,w}|d_{l,w,r}, d_{l,w,g}, d_{l,w,b}, m, \rho') = 1- \\ &\prod_{c \in \{r,g,b\}} [1 - P(d_{l,w}|d_{l,w,c} = 1 \wedge d_{l,w,c' \neq c} = 0, m, \rho')]^{d_{l,w,c}} \end{aligned} \tag{12}$$

We define the inner term as the single channel estimator:

$$P(d_{l,w} = 1|d_{l,w,c} = 1 \wedge d_{l,w,c' \neq c} = 0, m, \rho') \equiv P(d_{l,w,c}|m, \rho') \tag{13}$$

We define it with an analogous version of the model from Equation 10 with only one channel:

$$\frac{P(\rho'_c|d_{l,w,c}, \mu_c, \sigma_c^2) \times P(d_{l,w,c}|\mu_c, \sigma_c^2)}{\sum_{d_{l,w} \in \{0,1\}} P(\rho'_c|d_{l,w}, \mu_c, \sigma_c^2) \times P(d_{l,w}|\mu_c, \sigma_c^2)} \tag{14}$$

Figure 3e shows a scene segmented with the noisy Or model. Note the accurate segmentation of the scene obtained by the robot's ability to move objects into the scene and compare the information using an orthographic projection. This segmentation allows us to make an appearance model of an object as viewed from above by using the discrepancy to segment it from the background. We can use this segmented model to detect and localize objects. Additionally, depending on where m is placed in space, we can render different photographs in different directions. For tabletop manipulation, we place m above and parallel to the table in order to obtain an orthographic projection of the camera. This synthetic photograph corresponds to an image taken by a virtual 35 cm lens that captures the entire region as viewed from directly above. In contrast, we can place m vertically to view outward as shown in Figure 6. We can sweep the focus value, z across many values in order to focus our virtual camera at different points in space. (See our video[2].)

B. Camera Calibration

In order to accurately focus rays in software, we must be able to determine the path of a ray of light corresponding to a pixel in a camera image given the camera pose for that image, p_h, and the pixel coordinate generating that ray, (i, j). Typical least squares calibration tracks the position of features across multiple views of a scene, and then uses least squares to find the camera parameters [26]. We can use synthetic photography to calibrate the camera by imaging a known plane and choosing camera parameters to bring the synthetic photograph into focus.

[2] https://youtu.be/ZHn2OQ3Yj7I

We define a calibration function of the form $\mathcal{C}(i, j, p_h, z) \to \{(x, y)\}$. To perform calibration, we first define a model for mapping between pixel coordinates and world coordinates. Then we find maximum likelihood model parameters using Equation 4.

We use (x^p, y^p) to denote the pixel coordinate $(i = y^p, j = x^p)$, to simplify the notation of the matrix math. Suppose the image is w pixels wide and h pixels tall so that $a = (x^p, y^p, 1, 1)$ is a pixel in the image located at row y and column x, x can span from 0 to $w - 1$, and y can span from 0 to $h - 1$. We specify x^p and y^p as integer pixels, assign a constant value of 1 to the z coordinate, and augment with a fourth dimension so we can apply full affine transformations with translations. Assume that the principle point c_p of the image is $c_p = (c_{px}, c_{py}) = (\frac{w}{2}, \frac{h}{2})$. That is to say, the aperture of the camera is modeled as being at the origin of the physical coordinate frame, facing the positive z half space, collecting light that travels from that half space towards the negative z half space, and the z axis intersects c_p. In the pinhole model, only light which passes through the origin is collected, and in a real camera some finite aperture width is used. We define a matrix T to correspond to the affine transform from p_h to the coordinate system centered on the camera, and S_z to correspond to the camera parameters. If we want to determine points that a ray passes through, we must specify the point on the ray we are interested in, and we do so by designating a query plane parallel to the xy-axis by its z coordinate. We can then define the calibration function as a matrix operation:

$$TS_z \begin{bmatrix} (x^p - c_{px}) \\ (y^p - c_{py}) \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \\ 1 \end{bmatrix} \tag{15}$$

To determine the constants that describe pixel width and height in meters, we obtained the relative pose of the stock wrist camera from the factory measurements. Here M_x and M_y are the camera magnification parameters:

$$S_z = \begin{bmatrix} M_x \cdot z & 0 & 0 & 0 \\ 0 & M_y \cdot z & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{16}$$

We model radial distortion in the typical way by making S_z a function not only of z but also quadratic in $(x^p - c_{px})$ and $(y^p - c_{py})$. We omit the details due to space.

Calibrating the camera involves finding the magnification terms M_x and M_y (though the principle points and the radial quadratic terms can be found by the same means). To find the magnification coefficients, we set a printed paper calibration target on a table of known height in front of Baxter. We collect camera images from a known distance above the table and estimate the model for the collected rays, forming a synthetic image of the calibration target. The values for M_x and M_y which maximize the likelihood of the observed rays under the estimated model in Equation 4 are the values which yield the correct pixel to global transform, which incidentally are also the values which form the sharpest synthetic image. We find M_x and M_y with grid search. Determining such parameters by iteratively rendering is a form of bundle adjustment, and also allows us to perform 3D reconstruction. It is repeatable and precise, and the cameras are consistent enough that we can provide a default calibration that works across different Baxter robots.

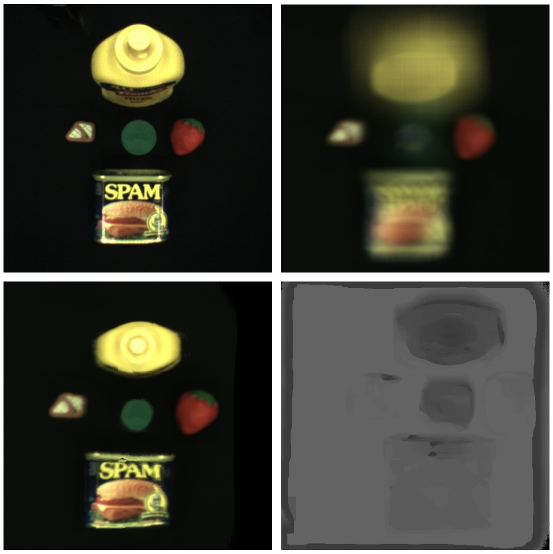


Fig. 5: A tabletop scene. Top Left: A single image from the wrist camera, showing perspective. Top Right: Refocused image converged at table height, showing defocus on tall objects. Bottom Left: Maximum likelihood RGB images, showing all objects in focus, specular reflection reduction, and perspective rectification. Bottom Right: Depth estimates for maximum likelihood images.

IV. Evaluation

To evaluate our framework, we demonstrate its capabilities to infer 3D structure, to detect and suppress reflections, and to pick reflective objects in non-Lambertian scenes.

A. Data Collection Parameters

For all work in this paper, we used the Baxter robot with its built-in wrist camera. The camera is a 1280×800 RGB sensor with a fixed focal length of $1.2\,\mathrm{mm}$. The pixel size is $3\,\mu\mathrm{m} \times 3\,\mu\mathrm{m}$. It is a $1.2\,\mathrm{mmF}/2.4$ lens. The depth of field is $9.4\,\mathrm{cm}$ to infinity, with the lens set for optimum focus at $19\,\mathrm{cm}$. (Some of these figures are available on the Rethink SDK website; others come from communications with Rethink Robotics.) We set it to a resolution of 400×640. We manually adjusted white balance and gain to ambient lighting conditions. We turned off automatic gain control and white balance adjustment because they introduce variance in the observed intensity values from different positions.

When collecting data, we move the arm slowly in a plane at approximately $5.5\,\mathrm{cm/s}$. We turn off all other processing to collect images and poses at the maximum frame rate, which leads to end effector pose returns at approximately 100 Hz and images at approximately 25 Hz. We use a stream buffer and linear interpolation to assign a pose to each image based on its time stamp, and treat this pose as ground truth. The resolution of the Baxter joint sensors is 0.023 degrees per tick resolution, with a typical accuracy on the order of ± 0.10 degrees and worst case ± 0.25 degrees accuracy when approaching joint limits [22]. In the worst case, these joint errors lead to up to 5 mm of error in the estimated position of the end effector. However in practice, after the arm has assumed a position and come to a stop, its position error is less than 1 mm. Rethink Robotics states that their specified repeatability is ± 2.5 mm (personal communication).

B. Inferring 3D Structure

To infer 3D structure, we estimate m at each height z using grid search. For each z, we compute the μ_c and σ^2 as the sample mean and variance of rays that intersect each cell. Then we can assign the maximum likelihood height for each cell as the one that has the minimum variance. Most generally, each ray should be associated with exactly one cell; however this model requires a new factorization for each setting of z at inference time, as rays must be reassociated with cells when the height changes. In particular, the rays associated with one cell, $R_{l,w}$, might change due to the height of a neighboring cell, requiring a joint inference over all the heights, z. If a cell's neighbor is very tall, it may occlude rays from reaching it; if its neighbor is short, that occlusion will go away.

As an approximation, instead of labeling each ray with a cell, we optimize each cell separately, over counting rays and leading to some issues with occlusions. This approximation is substantially faster to compute because we can analytically compute $\mu_{l,w}$ and $\sigma^2_{l,w}$ for a particular z as the sample mean and variance of the rays at each cell. This approximation works well when there are small variations in distances and little occlusion because it over counts each cell by approximately the same amount. We expect it to have more issues when it estimates scenes with significant variation in depth and occlusion, such as that shown in Figure 6. Under this approximation, occluded regions will contain more rays that should have been assigned to other cells, leading to overestimates of the variance of these cells. In the future we plan to explore EM approaches to perform ray labeling so that each ray is assigned to a particular cell at inference time.

Figure 5 shows the depth estimate for a tabletop scene computed using this method. The images were taken with the camera 38 cm from the table. The RGB map is composed from metrically calibrated images that give an orthographic top down view that appears in focus at every depth. Such a map greatly facilitates object detection, segmentation, and other image operations. Note that fine detail such as the letters on the SPAM is visible, as well as the shape of the top of the mustard bottle in the depth map. The orthographic projection is useful for detection and localization because it provides a canonical view of the object, enabling direct image-matching localization approaches to be extremely successful. Our approach can not only render a top-down orthographic view, but other canonical views as desired (if data is available from the camera).

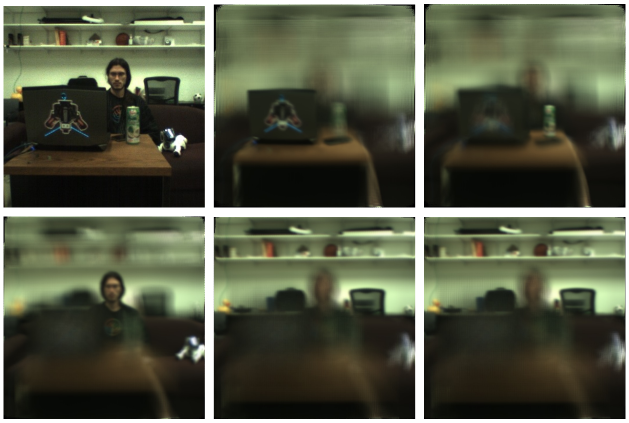


Fig. 6: A room scene. Top Left: A single image from the wrist camera. Remaining: Refocused photographs computed with approximately 4000 wrist images and focused at 0.91, 1.11, 1.86, 3.16, and 3.36 meters.

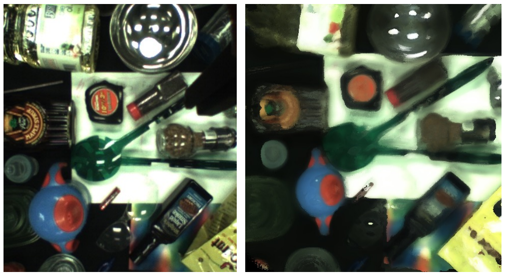


Fig. 7: Left: An image from Baxter's wrist camera, which contains many reflections from the overhead light. Right: an orthographic photograph of the same scene synthesized from 1000 wrist camera images with reflections suppressed.

C. Detecting and Suppressing Reflections

Overhead lights induce specular highlights and well formed images on shiny surfaces, as well as broader and more diffuse pseudo-images on textured or wet surfaces. We can use information contained in the light field to remove some of the reflections in an estimated map, as long as affected portions were seen without reflections from some angles.

Specular reflections of the surface of an object tend to form virtual images which are in focus behind the object. When a human looks at a shiny object or the surface of still water, they might first focus on a specular reflection formed by the object, realize it is bright and therefore not the true surface, and then look for a different color while focusing closer. To construct a synthetic photograph with reduced reflections, we perform a focus sweep to identify rays that are part of the object, in focus at one depth, and separately, rays part of a highlight, in focus at a different (deeper) depth. Specifically, we first estimate z values for the map by approximate maximum likelihood as in the previous section. Then, we re-estimate the z values by re-rendering at all heights while throwing out rays that are too similar to the first map and only considering heights which are closer to the observer than the original estimate. That is, the second estimate looks to form a different image that is closer to the observer than the first. The final image is formed by taking either the first or second value for each cell, whichever has the smallest variance, which is a measure of the average likelihood of the data considered by that cell. If the second estimate considered too few samples, we discard it and choose the first. Figure 7 shows an example image from Baxter's wrist camera showing highly reflective objects, and the same scene with reflections suppressed. This view trades off resolution but also removes significant artifacts that would vary as the camera obtained different views.

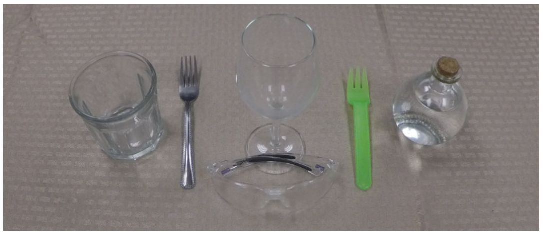
Identifying reflections using optical cues instead of colors allows us to remove spots and streaks of multiple light sources in a previously unencountered scene without destroying brightly colored Lambertian objects. The requirement that the second image form closer than the first dramatically reduces artifacts, but when considering z values over a large range, some image translocation can occur, causing portions of tall objects to bleed into shorter ones. When considering objects of similar height, this algorithm suppresses reflections substantially without admitting many false positives. Especially on sharply curved objects there will sometimes be a region that was covered by a reflection from all angles. If such a reflection occurs near a boundary it may aid in localizing the object. If it occurs on the inside of an object region, it will often be surrounded by a region of suppressed reflections, which are detected by the algorithm. Concave reflective regions can form reflections that are impossible to remove with this algorithm since they form complex distorted images which can project in front of the object, as in the metal bowl in our example.

D. Picking Objects

By forming orthographic projections we can use sliding window detectors to reliably localized Lambertian objects.

1) Picking Accuracy: For each trial, we moved the object to a random location on the table within approximately 25 cm of the arm's starting location. Then we localized the object using the wrist camera and picked it. We used two different modes when localizing. For the Point Scan, the wrist camera used an average of 40 images taken at the arm's starting location. We verified that the object was always in view of the wrist camera when this image was taken, although part of it may have been occluded by the gripper. In the Line Scan, we moved the arm 28 cm back and forth over the workspace to make a synthetic photograph using about 140 images. Next we estimated the object's position using image matching in the synthetic photographs and grasped it. Results appear in Table 8b.

Notably, the Lambertian green fork was picked every time with both scans; even the Point Scan was able to successfully localize the object. However, when using the Point Scan to localize a similar reflective object, the robot frequently missed due to different appearances of the object depending on its location on the table and reflections from the overhead lights. However when the robot used our model, performance returns to $10/10$, comparable to the Lambertian object. These results demonstrate that our approach enables the robot to average away the reflections by taking into account the light field information.

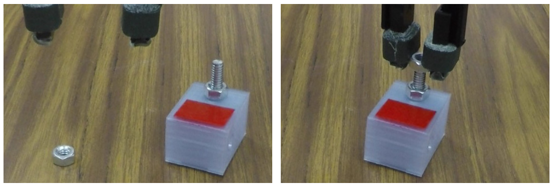


(a) Evaluation objects from left to right: glass tumbler, metal fork, wine glass, safety glasses, green fork, and glass flask.

Object	Point Scan	Line Scan
Green Fork	10/10	10/10
Metal Fork	5/10	10/10
Glass Tumbler	7/10	10/10
Wine Glass	3/10	10/10
Safety Glasses	8/10	9/10
Glass Flask	3/10	8/10

(b) Picking performance.

Fig. 8: Evaluation objects and picking performance.



(a) Initial scene. (b) Screwing the nut.

Fig. 9: Our approach allows us to an RGB camera to localize objects well enough to screw a nut onto a $0.25''$ bolt.

Additionally our approach is able to pick many transparent objects. For our evaluation we focused at the table height (without doing the z inference), enabling the robot to pick out the base of the object reliably, leading to more successful picks.

2) Screwing a Nut on a Bolt: As a test of our system's accuracy, we programmed Baxter to screw a nut onto a $0.25''$ bolt. The robot first localized a nut and the bolt on the table. Next it used an open-loop program to pick up the nut and place it on the bolt, given these inferred locations. The robot was able to perform this task several times, showing that the very precise localization enabled it to robustly and accurate pick up the nut and place it in the bolt, even though both objects are very small and the accuracy required is near the limits of the Baxter robot. Figure 9 shows the scene as the robot completes the task. See our video attachment [3] for a demonstration.

[3] https://youtu.be/ZHn2OQ3Yj7I

3) Picking from Running Water: To test a highly non-Lambertian scene, we filled a sink with several inches of water and used a pump to induce an appreciable current. The current maintained turbulence across the water surface, which generated a substantial number of shadows and reflections in the moving water and surrounding surfaces, shown in Figure 1a. Next we placed a metal fork in the sink under the flowing water. We used synthetic photographs focused at the bottom of the sink to successfully localize and pick the fork 24 times out of 25 attempts with Baxter's 4 cm gripper. There were no 180 degree confusions, and the single miss was pinched but not lifted. Our video attachment[4] shows the robot performing this task.

The constant presence of reflections and shadows make it challenging to pick the fork from beneath the water based on individual images. Bright edges are present at all scales, so gradient based methods will have trouble, and the average intensity of the fork is similar to the average intensity of turbulent water, so blurring the scene will significantly degrade the signal. A stronger signal of the bottom of the sink may be obtained by leaving the camera still and averaging incoming images. This may eliminate reflections and shadows, but it only provides one perspective of the fork, which is prone to noise due to fixed reflections; our quantitative results show that even a metal fork on a table is hard to localize from one perspective. By synthesizing images which are focused at the bottom of the sink, we get the benefits of averaging images from a still camera combined with the robustness of multiple perspectives. Neither water nor metal is particularly conducive to imaging with IR depth cameras, making it harder to solve this problem with such sensors.

E. Discussion

Overall our approach allows a systematic treatment of reflections and specular artifacts, enabling the robot to do accurate picking in challenging situations such as a fork in a sink with running water, as well as pick-and-place precisely enough to tighten a $0.25''$ nut on a bolt. These results perform at near the ability of the robot to localize itself. However limitations remain, due to the time it takes to collect data and the slow speed of the wrist camera. Additionally a very accurate calibration is necessary for this approach to produce reliable results; inaccurate calibration results in blurry images and lower localization accuracy. In principle, it should be possible to obtain super-resolution synthetic photographs, for example by filtering the camera pose and accounting for these inaccuracies. That said, our calibration system is accurate enough to be useful for many real-world applications. A second class of problems arises due to lighting and shadows. The arm's own shadow can be detected when creating models of objects and background, and leads to spurious detections if not enough data has been collected. Similarly when making a model of an object, if it is in an environment with strong shadows, the shadow itself will be part of the model, making it difficult to detect and localize the object.

V. Conclusion

In this paper we have contributed an approach to perception using light fields which can be implemented on a 7 DoF robotic arm with an eye in hand 2D RGB camera. We described the algorithms necessary to calibrate the camera and demonstrated the use of synthetic photography to extract 3D structure, remove specular highlights from images, and pick non-Lambertian objects. To our knowledge, this paper is the first to describe using Baxter (or any robotic arm) to collect light field data and render synthetic photographs. Furthermore, the light field capturing abilities of Baxter in this paradigm are unique in scale, flexibility, and precision when compared to other modalities of light field collection.

The detection and matching techniques in this paper rely on directly matching images, which is made possible by using synthetic photography to create a standard view such as an orthographic projection. However many techniques in computer vision, from SIFT to deep learning to color histograms, can be applied to the synthetic photograph as well as to individual images. In many cases we expect them to work better on the synthetic photograph because of noise reduction, the removal of specular artifacts, and the ability to orthographically project for a consistent point of view. Our approach is not a replacement for these techniques but rather one way to make them more powerful when they have access to a moving camera.

In the future we plan to explore methods to more intelligently collect data and minimize the time needed for accurate perception. Similarly, depending on the task required and accuracy needed, the robot can intelligently choose how much data to gather; for precise manipulation tasks it can aim for higher precision than for picking up an object, when it might only need a single frame. A camera array could allow collection of frames in parallel by allowing the robot to gather data from multiple viewpoints simultaneously.

Overall our approach represents one way to integrate information from across multiple camera, allowing the robot to configure its perceptual approach for the task at hand, moving low and slow for fine-grained manipulation tasks such as screwing the nut on the bolt, or taking fewer images from farther away when less precision is acquired. This approach represents steps toward a more general probabilistic framework for computer vision for robotics.

VI. Acknowledgments

This material is based upon work supported by the National Science Foundation under grant numbers IIS-1637614 and IIS-1426452, the National Aeronautics and Space Administration under grant number NNX16AR61G, DARPA under grant number D15AP00102, and General Dynamics Land Systems/US Army under grant numbers 40228388W911NF-10-2-0016. We also appreciate helpful comments from Lawson Wong and Liam Paull.

[4] https://youtu.be/ZHn2OQ3Yj7I

REFERENCES

[1] Joseph E Banta, LR Wong, Christophe Dumont, and Mongi A Abidi. A next-best-view system for autonomous 3-d object reconstruction. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 30 (5):589–598, 2000.

[2] Paul N Bennett. Assessing the calibration of naive bayes posterior estimates. Technical report, DTIC Document, 2000.

[3] Nikolaus Correll, Kostas E Bekris, Dmitry Berenson, Oliver Brock, Albert Causo, Kris Hauser, Kei Okada, Alberto Rodriguez, Joseph M Romano, and Peter R Wurman. Lessons from the amazon picking challenge. *arXiv preprint arXiv:1601.05484*, 2016.

[4] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *Computer Vision–ECCV 2014*, pages 834–849. Springer, 2014.

[5] Clemens Eppner, Sebastian Höfer, Rico Jonschkowski, Roberto Martín-Martín, Arne Sieverling, Vincent Wall, and Oliver Brock. Lessons from the amazon picking challenge: Four aspects of building robotic systems. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016. doi: 10.15607/RSS.2016.XII.036.

[6] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

[7] Todor Georgiev, Zhan Yu, Andrew Lumsdaine, and Sergio Goma. Lytro camera technology: theory, algorithms, performance analysis. In *IS&T/SPIE Electronic Imaging*, pages 86671J–86671J. International Society for Optics and Photonics, 2013.

[8] Evan Herbst, Peter Henry, Xiaofeng Ren, and Dieter Fox. Toward object discovery and modeling via 3-d scene comparison. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2623–2629. IEEE, 2011.

[9] Alexander Kasper, Zhixing Xue, and Rüdiger Dillmann. The kit object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, 31(8):927–934, 2012.

[10] Dov Katz and Oliver Brock. Manipulating articulated objects with interactive perception. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 272–277. IEEE, 2008.

[11] Dirk Kraft, Renaud Detry, Nicolas Pugeault, Emre Baseski, Frank Guerin, Justus H Piater, and Norbert Kruger. Development of object and grasping knowledge by robot exploration. *Autonomous Mental Development, IEEE Transactions on*, 2(4):368–383, 2010.

[12] Michael Krainin, Peter Henry, Xiaofeng Ren, and Dieter Fox. Manipulator and object tracking for in-hand 3d object modeling. *The International Journal of Robotics Research*, 30(11):1311–1327, 2011.

[13] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 307–314. IEEE, 1999.

[14] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

[15] Natalia Lyubova, David Filliat, and Serena Ivaldi. Improving object learning through manipulation and robot self-identification. In *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, pages 1365–1370. IEEE, 2013.

[16] Joseph Modayil and Benjamin Kuipers. The initial development of object knowledge by a learning robot. *Robotics and autonomous systems*, 56(11):879–890, 2008.

[17] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[18] Ren Ng. *Digital light field photography*. PhD thesis, stanford university, 2006.

[19] John Oberlin and Stefanie Tellex. Time-Lapse Light Field Photography With a 7 DoF Arm. In *RSS Workshop on Geometry and Beyond - Representations, Physics, and Scene Understanding for Robots*, 2016.

[20] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[21] Cody J Phillips, Matthieu Lecce, and Kostas Daniilidis. Seeing glassware: from edge detection to pose estimation and shape recovery. In *Robotics: Science and Systems*, 2016.

[22] Rethink Robotics. Baxter sdk version 1.2.0. http://sdk.rethinkrobotics.com, 2017.

[23] José Jeronimo Rodrigues, Jun-Sik Kim, Makoto Furukawa, Joao Xavier, Pedro Aguiar, and Takeo Kanade. 6d pose estimation of textureless shiny objects using random ferns for bin-picking. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3334–3341. IEEE, 2012.

[24] David Schiebener, Jun Morimoto, Tamim Asfour, and Aleš Ude. Integrating visual perception and manipulation for autonomous learning of object representations. *Adaptive Behavior*, 21(5):328–345, 2013.

[25] Brandon M Smith, Li Zhang, Hailin Jin, and Aseem Agarwala. Light field video stabilization. In *2009 IEEE 12th international conference on computer vision*, pages 341–348. IEEE, 2009.

[26] Richard Szeliski. *Computer vision: algorithms and applications*. Springer, 2010.

[27] Michael W Tao, Pratul P Srinivasan, Sunil Hadap, Szymon Rusinkiewicz, Jitendra Malik, and Ravi Ramamoor-

thi. Shape estimation from shading, defocus, and correspondence using light-field angular coherence. 2016.

[28] Ales Ude, David Schiebener, Norikazu Sugimoto, and Jun Morimoto. Integrating surface-based hypotheses and manipulation for autonomous segmentation and learning of object representations. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1709–1715. IEEE, 2012.

[29] Ting-Chun Wang, Jun-Yan Zhu, Ebi Hiroaki, Manmohan Chandraker, Alexei A Efros, and Ravi Ramamoorthi. A 4d light-field dataset and cnn architectures for material recognition. *arXiv preprint arXiv:1608.06985*, 2016.

[30] Sven Wanner, Stephan Meister, and Bastian Goldluecke. Datasets and benchmarks for densely sampled 4d light fields. In *VMV*, pages 225–226. Citeseer, 2013.

[31] Jason C Yang, Matthew Everett, Chris Buehler, and Leonard McMillan. A real-time distributed light field camera. *Rendering Techniques*, 2002:77–86, 2002.

[32] Matthias Zobel. Object tracking and pose estimation using light-field object models. 2002.