Exploiting & Refining Depth Distributions with Triangulation Light Curtains

Yaadhav Raaj Siddharth Ancha Robert Tamburo David Held Srinivasa G. Narasimhan
The Robotics Institute, Carnegie Mellon University
{ryaadhav, sancha, rtamburo, dheld, srinivas}@cs.cmu.edu

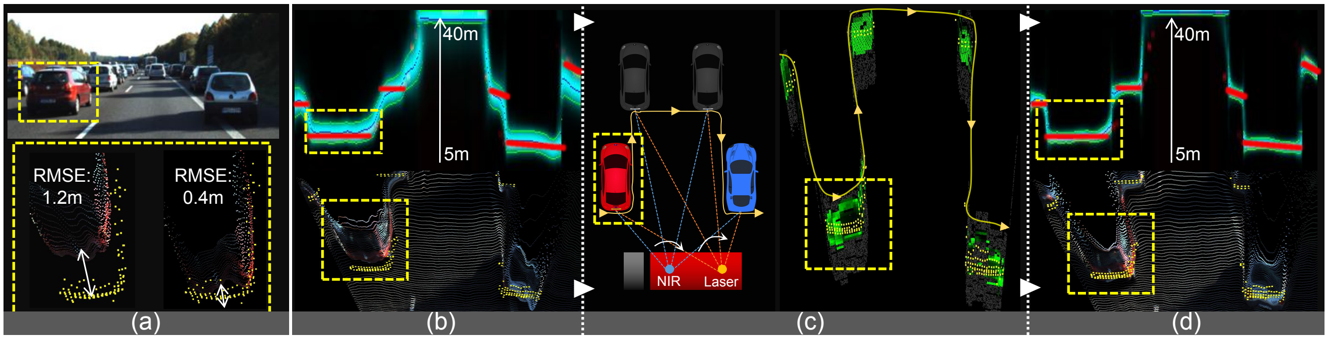

Figure 1: **(a)** We show how errors in Monocular Depth Estimation are corrected when used in tandem with an Adaptive Sensor such as a Triangulating Light Curtain (Yellow Points and Red lines are Ground Truth). **(b)** We predict a per-pixel Depth Probability Volume from Monocular RGB and we observe large per-pixel uncertainties ($\sigma = 3m$) as seen in the Bird's Eye View / Top-Down Uncertainty Field slice. **(c)** We actively drive the Light Curtain sensor's Laser to exploit and sense multiple regions along a curve that maximize information gained. **(d)** We feed these measurements back recursively to get a refined depth estimate, along with a reduction in uncertainty ($\sigma = 1m$).

Abstract

Active sensing through the use of Adaptive Depth Sensors is a nascent field, with potential in areas such as Advanced driver-assistance systems (ADAS). They do however require dynamically driving a laser / light-source to a specific location to capture information, with one such class of sensor being the Triangulation Light Curtains (LC). In this work, we introduce a novel approach that exploits prior depth distributions from RGB cameras to drive a Light Curtain's laser line to regions of uncertainty to get new measurements. These measurements are utilized such that depth uncertainty is reduced and errors get corrected recursively. We show real-world experiments that validate our approach in outdoor and driving settings, and demonstrate qualitative and quantitative improvements in depth RMSE when RGB cameras are used in tandem with a Light Curtain.

1. Introduction

Spinning fixed scan LIDARs have been the de-facto sensor of choice in safety critical systems such as Advanced driver-assistance systems (ADAS), due to their reliability in depth estimation. However, their reduced spatial-resolution, multi-path interference and their prohibitive cost has made en-masse adoption in personal vehicles hard. To counter these issues, depth estimation from RGB cameras has been heavily researched. However, issues such as oversaturation, feature correspondence errors and scale ambiguity has made relying on these sensors unsafe.

To capture the error and uncertainty in RGB-only depth estimation, previous work had formulated that task as a probabilistic regression problem, by predicting per-pixel depth distributions via a Depth Probability Volume (DPV) [16] [7] [28]. The DPV provides both a Maximum Likelihood-Estimate (MLE) of the depth map, as well as the corresponding per-pixel uncertainty measure. However, these works do not adaptively or physically correct for this uncertainty, instead relying purely on multi-view camera constraints for passive correction.

In this work, we have devised the first known framework that *adaptively* exploits the depth uncertainty in a per-pixel DPV from RGB images and refined it via an Adaptive Depth Sensor called a Triangulating Light Curtain [15]. It has a steerable Laser Line that can be driven by a Galvomirror in tandem with a Rolling Shutter camera to generate a 3D ruled surface to sample the world. We have chosen this sensor due to its low cost ($1k vs lidar ~$25k), high spatial angular resolution (0.02° vs lidar 0.4°), and high frame-rate (60fps vs lidar 20fps).

We begin by formulating an iterative Bayesian inference approach to adaptive depth sensing using only the Light Curtain (LC). This is done by building and adapting the 3D DPV representation as a collapsible 2D Uncertainty Field (UF), formulating a probabilistic depth representation of the sensor model and building planning and sensing policies within the sensor constraints. We then build a deep learning architecture that can generate a similar DPV from Monocular or Stereo RGB inputs, and use that as a prior for adaptive sensing. We then fuse the LC measurements back into our network to get a refined depth estimate (see Fig. 1).

We conducted experiments of adaptive depth sensing from the LC alone by starting with a Gaussian prior, and showed convergence to true depth with enough iterations. We then trained a network to predict depth distributions from RGB images, used that as a prior for sensing, and fed those new LC measurements back to the network. Through extensive experiments with a simulated LC (with KITTI dataset [10]) and sensors in the real-world, we show significant speedup in depth convergence and increased accuracy (see Fig. 1). As a result, our method has the potential of being a higher resolution, lower cost alternative to a LIDAR.

2. Prior Work

Depth from Active Sensors: Active sensors use a fixed scan light source / receiver to perceive depth. Long range outdoor depth from these such as commercially available Time-Of-Flight cameras [1] or LIDARs [3] [2] provide dense metric depth with confidence values with wide usage in research [10] [6] [8]. However, apart from low resolution, these sensors are difficult to procure and expensive, making everyday personal vehicle adoption challenging.

Depth from Adaptive Sensors: Adaptive sensors use a dynamically controllable light source / receiver instead. These have been making headway in the Long Range Outdoor space. Adaptive Sensing via focal length/baseline variation through the use of servos/motors [17] [9] [18] [21], directionally controlled Time-of-Flight Ranging using a MEMS mirror / laser [22] [23] [27], Gated Depth Imaging [24] [13] [12] and finally, sampling specific depth profiles using Triangulation Light Curtains [15] [25] [4] are just some examples. However, these methods do not seem to exploit or fuse data from RGB modalities yet. Various work by Bergman, Nishimura et. al. [5] [19] and Pittaluga et. al. [20] present sensors and algorithms for adaptive sensing via 2D angular sampling, providing precise depth at limited number of pixels. However, our light curtain approach does depth sampling via adaptive depth gating, giving useful information at every pixel at a higher resolution. Nishimura's sensor uses a SPAD where light is spread out over the entire FOV limiting it's range and operation outdoors due to ambient light. The light curtain however, maximizes the light energy on the region of interest via triangulation.

Depth from RGB: Depth from Monocular and Multi-Camera RGB has been extensively studied. We focus on a class of Probabilistic Depth estimation approaches that have reformulated the problem as a prediction of per-pixel depth distribution [16] [28] [7] [30] [14] [26]. Some of this work has actually passively exploited and refined [16] [26] the uncertainty in the depth values via Moving Cameras and Multi-View-Camera constraints, but have not used the capabilities of the slew of Adaptive Sensors available.

We hope to fill this gap by investigating if a Probabilistic Depth representation from RGB sensors can be exploited by an Adaptive Sensor such as a Light Curtain to potentially match the precision of LIDARs but in a low cost manner.

3. Sensor Setup

The Light Curtain device (Fig 2) consists of a rolling shutter Near-Infrared (NIR) camera rotated 90° (that images planes in the world per pixel column), a Line Laser module and a Galvomirror (that generates planes of light depending on the angle). The exact sensing location is obtained by intersecting (triangulating) the imaging and laser planes. Sweeping this laser line creates a 3D ruled surface called a curtain. We can place a curtain along any surface by controlling the galvo and rolling shutter speed subject to it's physical constraints, making the sensor adaptive in nature. Note that the image and laser planes have some divergence, so their intersection results in a volume in space (bounded by purple points in Fig. 2) with some *thickness*, where any objects that intersect it result in higher intensities in the NIR image. This means that as the sensing location approaches the true surface, pixel intensities on NIR image increases.

Real-world experiments are conducted using our array of sensors consisting of an RGB Stereo Camera Pair, the Light Curtain device, and a 128-beam Lidar for accuracy validation and RGB depth estimation network training. Simulated experiments are also conducted with KITTI dataset [10], through a Light Curtain Simulator that uses the ground truth depth map along with the ability to vary NIR instrinsics, laser extrinsincs, Galvomirror speed and laser divergence/thickness and angle.

4. Depth from Light Curtains only

Before considering RGB + Light Curtain fusion (sec. 6), we begin by focusing on the problem of adaptively discovering the depth of a scene using only the light curtain.

4.1. Representation

We wish to estimate the depth map $\mathbf{D} = \{d_{u,v}\}$ of the scene, which specifies the depth value $d_{u,v}$ for every camera pixel (u, v) at spatial resolution [H,W]. Since there is inherent uncertainty in the depth value at every pixel, we represent a *probability distribution* over depths for every pixel. Let us define $\mathbf{d}_{u,v}$ to be a *random variable* for depth predictions at the pixel (u, v). We quantize depth values into a set $\mathcal{D} = \{d_0, \dots, d_{N-1}\}$ of N discrete, uniformly spaced depth values lying in $(d_{\text{min}}, d_{\text{max}})$. All the predictions $\mathbf{d}_{u,v} \in \mathcal{D}$ belong to this set. The output of our depth

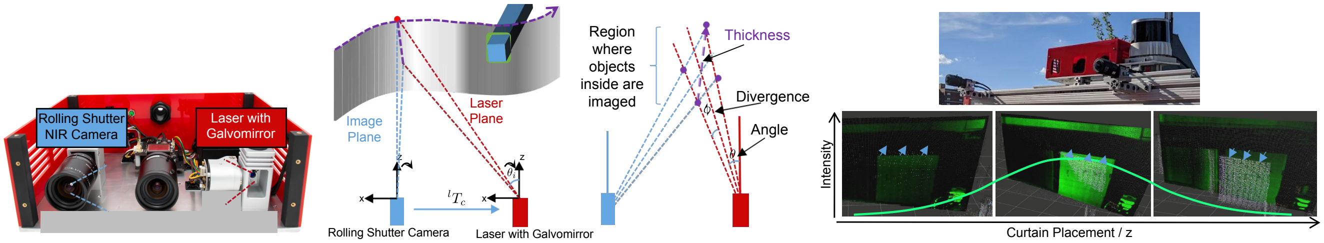

Figure 2: **Left:** Our adaptive sensor of choice, the Triangulation Light Curtain (LC) [15], consists of a laser line, Galvomirror and NIR camera. **Middle:** The light curtain senses a ruled 3D surface extruding from a given top-down 2D curve we call a curtain. Surfaces within the *thickness* of the curtain, result in higher intensity in the NIR image. **Right:** A planar curtain swept across various depths. As the curtain plane approaches the true surface, the measured intensity increases, due to the sensing location and curtain thickness. Above that we show our real-world sensor setup.

estimation method for each pixel is a probability distribution $P(\mathbf{d}_{u,v})$, modeled as a categorical distribution over $\mathcal{D}$. In this work, we use $N = 64$, resulting in a Depth Probability Volume (DPV) tensor of size [64, W, H]:

$$\mathcal{D} = \{d_0, \ldots, d_{N-1}\}; d_q = d_{\text{min}} + (d_{\text{max}} - d_{\text{min}}) \cdot q \quad (1)$$

$$\sum_{q=0}^{N-1} P(\mathbf{d}_{u,v} = d_q) = 1 \text{ (q is the quantization index)} \quad (2)$$

$$\text{Depth estimate } = \mathbb{E}[\mathbf{d}_{u,v}] = \sum_{q=0}^{N-1} P(\mathbf{d}_{u,v} = d_q) \cdot d_q \quad (3)$$

This DPV can be initialized using another sensor such as an RGB camera, or can be initialized with a Uniform or Gaussian distribution with a large σ for each pixel.

While an ideal sensor could choose to plan a path to sample the full 3D volume, our light curtain device only has control over a top-down 2D profile. Hence, we compress our DPV into a top-down an "Uncertainty Field" (UF) [28], by averaging the probabilities of the DPV across a subset of each column (Fig. 3). This subset considers those pixels (u, v) whose corresponding 3D heights $h(u, v)$ are between $(h_{\text{min}}, h_{\text{max}})$. The UF is defined for the camera column u and quantized depth location q as:

$$UF(u, q) = \frac{1}{|\mathcal{V}(u)|} \sum_{v \in \mathcal{V}(u)} P(\mathbf{d}_{u,v} = d_q)$$
$$\text{where } \mathcal{V}(u) = \{v \mid h_{\text{min}} \leq h(u, v) \leq h_{\text{max}}\} \quad (4)$$

We denote the categorical distribution of the uncertainty field on the u-th camera ray as:
$UF(u) = \text{Categorical}(d_q \in \mathcal{D} \mid P(d_q) = UF(u, q))$.

4.2. Curtain Planning

We can use the extracted Uncertainty Field (UF) to plan where to place light curtains. We adapt prior work solving light curtain placement as a constraint optimization / Dynamic Programming problem [4]. A single light curtain placement is defined by a set of control points $\{q(u)\}_{u=1}^{W}$, where u indexes columns of the camera image of width W, and $0 \leq q(u) \leq N - 1$. This denotes that the curtain intersects the camera rays of the u-th column at the discretized depth $d_{q(u)} \in \mathcal{D}$. We wish to maximize the objective $J(\{q(u)\}_{u=1}^{W}) = \sum_{u=1}^{W} UF(u, q(u))$. Let $\mathbf{X}_u$ be the 2D point in the top-down view that corresponds to the depth $q(u)$ on camera rays of column u. The control points $\{q(u)\}_{u=1}^{W}$ must be chosen to satisfy the physical constraints of the light curtain device: $|\theta(\mathbf{X}_{u+1}) - \theta(\mathbf{X}_u)| \leq \Delta\theta_{\text{max}}$ with θ_{max} being the max angular velocity of Galvo:

$$\arg \max_{\{q(u)\}_{u=1}^{W}} \sum_{u=1}^{W} UF(u, q(u))$$
$$\text{subj to } |\theta(\mathbf{X}_{u+1}) - \theta(\mathbf{X}_u)| \leq \Delta\theta_{\text{max}}, \; \forall 1 \leq u < W \quad (5)$$

4.3. Curtain Placement

The uncertainty field UF contains the current uncertainty about pixel-wise object depths $\mathbf{d}_{u,v}$ in the scene. Let us denote by $\pi(d^{c_k} \mid UF)$ the placement policy of the k-th light curtain, where $d^{c_k} = \{d^{c_k}_{u,v} \mid \forall u, v\}$. Our goal is to sample light curtain placements $d^{c_k} \sim \pi(d^{c_k} \mid UF)$ from this policy, and obtain intensities $i_{u,v}$ for every pixel.

To do this, we propose two policies: π_0 and π_1. In Fig. 4, we have placed a single curtain along the highest probability region per column of rays, but our goal is to maxi-

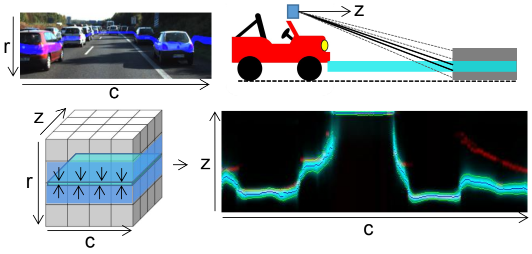

Figure 3: Our state space consists of a Depth Probability Volume (DPV) (left) storing per-pixel uncertainty distributions. It can be collapsed to a Bird's Eye Uncertainty Field (UF) (right) by averaging those rays in each row (blue pixels) of the DPV that correspond to a slice on the road parallel to the ground plane (right) (cyan pixels). Red pixels on UF represent the low resolution LIDAR ground truth.

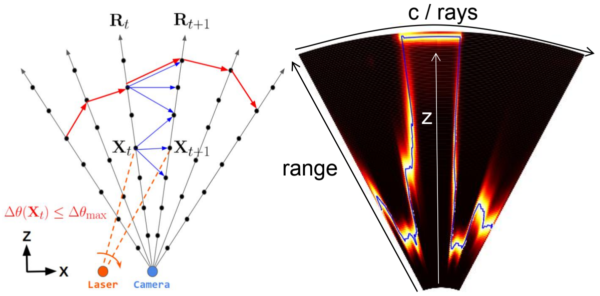

Figure 4: Given an Uncertainty Field (UF), our planner solves for an optimal galvomirror trajectory subject to it's constraints (eg. $\mathring{\theta}_{max}$). We show a 3D ruled surface / curtain placed on the highest probability region of UF.

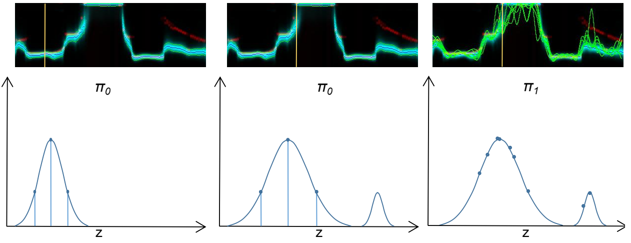

Figure 5: Sampling the world at the highest probability region is not enough. To converge to the true depth, we show policies that place additional curtains given UF. Let's look at a ray (in yellow) from the UF to see how each policy works. **Left:** π_0 given a unimodal gaussian with small σ. **Middle:** π_0 given a multimodal gaussian with larger σ. **Right:** π_1 given a multimodal gaussian with larger σ. Observe that π_1 results in curtains being placed on the second mode.

mize the information gained. For this, we generate corresponding entropy fields $H(u,q)_i$ to be input to the planner computed from $UF(u,q)$. We use two approaches to generate $H(u,q)$: π_0 finds the mean in each ray's distribution $UF(u)$ and selects a σ_{π_0} that determines the neighbouring span selected. π_1 samples a point on the ray given $UF(u)$.

As seen in Fig. 5, strategy π_0 is able to generate fields that adaptively place additional curtains around a consistent span around the mean with some σ_{π_0}, but is unable to do so in cases of multimodal distributions. π_1 on the other hand is able to place a curtain around the second modality, albeit with a lower probability. We will show the effects of both strategies in our experiments.

4.4. Observation Model

A curtain placement corresponds to specifying the depth for each camera ray indexed by u from the top-down view. After placing the light curtain, intensities $i_{u,v}$ are imaged by the light curtain's camera at every pixel (u,v). The measured intensity at each pixel is a function of the curtain placement depth $d^c_{u,v}$ on that camera ray, the unknown ground truth depth $\mathbf{d}_{u,v}$ of that pixel, the thickness of the light curtain $\sigma(u,v,d^c_{u,v})$ for a particular pixel and curtain placement, and the maximum intensity possible if a curtain is placed perfectly on the surface $p_{u,v}$ (varies from 0 to 1). From real world data Fig. 6, we find the intensity

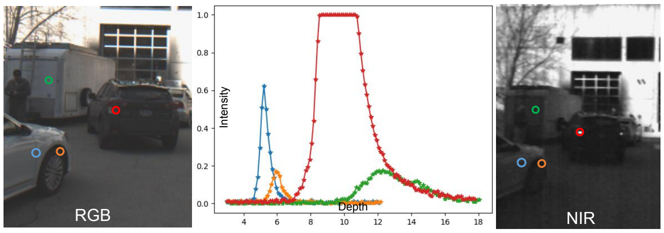

Figure 6: We sweep a planar light curtain across a scene at 0.1m intervals, and observe that the changes in intensity over various pixels follow an exponential falloff model. **Blue / Orange**: Car door has a higher response than the tire. **Green**: Object further away has a lower response with a larger sigma due to curtain thickness. **Red**: Retroreflective objects cause the signal to saturate

decays exponentially as the distance between the curtain placement $d^c_{u.v}$ and ground truth depth $\mathbf{d}_{u,v}$ increases, with the scaling factor $p_{u,v}$ parameterizing the surface properties. We also simulate sensor noise as a Gaussian distribution with standard deviation σ_{nse}. The overall sensor model $P(i_{u,v} \mid \mathbf{d}_{u,v}, d^c_{u,v})$ can be described as:

$$P(i_{u,v} \mid \mathbf{d}_{u,v}, d^c_{u,v}) \equiv$$

$$\mathcal{N}\left(i_{u,v} \mid \exp\left(-\left(\frac{d^c_{u,v} - \mathbf{d}_{u,v}}{\sigma(u,v,d^c_{u,v})}\right)^2\right) \cdot p_{u,v}, \sigma^2_{\text{nse}}\right) \quad (6)$$

Note that when $d^c_{u,v} = \mathbf{d}_{u,v}$ and $p_{u,v} = 1$, the mean intensity is 1 (the value), and it reduces exponentially as the light curtain is placed farther from the true surface. $p_{u,v}$ can be extracted from the ambient NIR image.

4.5. Recursive Bayesian Update

How do we incorporate the newly acquired information about the scene from the light curtain to update our current beliefs of object depths? Since we have a probabilistic sensor model, we use the Bayes' rule to infer the posterior distribution of the ground truth depths given the observations. Let $P_{\text{prev}}(u,v,q)$ denote the probability of the depth at pixel (u,v) being equal to d_q before sensing, and $P_{\text{next}}(u,v,q)$ the updated probability after sensing. Then by Bayes' rule:

$$\begin{aligned}
&P_{\text{next}}(u,v,q) \\
&= P(\mathbf{d}_{u,v} = d_q \mid i_{u,v}, d^{c_k}_{u,v}) \\
&= \frac{P(\mathbf{d}_{u,v} = d_q) \cdot P(i_{u,v} \mid \mathbf{d}_{u,v} = d_q, d^{c_k}_{u,v})}{P(i_{u,v} \mid d^{c_k}_{u,v})} \\
&= \frac{P(\mathbf{d}_{u,v} = d_q) \cdot P(i_{u,v} \mid \mathbf{d}_{u,v} = d_q, d^{c_k}_{u,v})}{\sum_{q'=0}^{N-1} P(\mathbf{d}_{u,v} = d_{q'}) \cdot P(i_{u,v} \mid \mathbf{d}_{u,v} = d_{q'}, d^{c_k}_{u,v})} \\
&= \frac{P_{\text{prev}}(u,v,q) \cdot P(i_{u,v} \mid \mathbf{d}_{u,v} = d_q, d^{c_k}_{u,v})}{\sum_{q'=0}^{N-1} P_{\text{prev}}(u,v,q') \cdot P(i_{u,v} \mid \mathbf{d}_{u,v} = d_{q'}, d^{c_k}_{u,v})} \quad (7)
\end{aligned}$$

Note that $P(i_{u,v} \mid \mathbf{d}_{u,v} = d_q, d^{c_k}_{u,v})$ is the sensor model whose form is given in Equation 6.

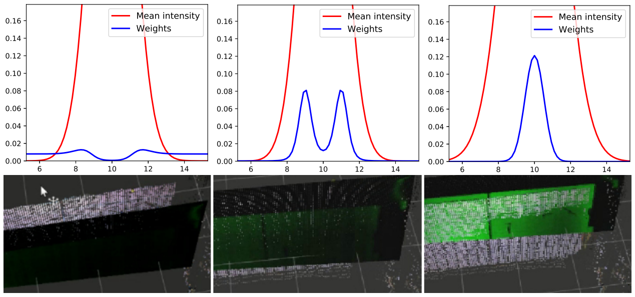

Figure 7: Visualization of the recursive Bayesian update method to refine depth probabilities after observing light curtain intensities. The curtain is placed at 10m. The red curves denote the expected intensity (Y-axis) as a function of ground truth depth (X-axis); this is the sensor model given in Eqn. 6. After an intensity is observed by the light curtain, we can update the probability distribution of what the ground truth depth might be using our sensor model and the Bayes' rule. The updated probability is shown by the blue curves, computed using the Bayesian update of Eqn. 7 (here, the prior distribution P_{prev} is assumed to be uniform, and $d^{c_k}_{u,v} = 10m$). **Left:** Low i return leads to an inverted Gaussian distribution at the light curtain's placement location, with other regions getting a uniform probability. **Middle:** Medium i means that the curtain isn't placed exactly on the object and the true depth could be on either side of the light curtain. **Right:** High i leads to an increased belief that the true depth is at 10m.

If we place K light curtains at a given time-step, we can incorporate the information received from all of them into our Bayesian update simultaneously. Since the sensor noise is independent of curtain placement, the likelihoods of the observed intensities can be multiplied across the curtains. Hence, the overall update becomes:

$$P_{\text{next}}(u, v, q)$$
$$= \frac{P_{\text{prev}}(u, v, q) \cdot \prod_{k=1}^{K} P(i_{u,v} \mid \mathbf{d}_{u,v} = d_q, d^{c_k}_{u,v})}{\sum_{q'=0}^{N-1} P_{\text{prev}}(u, v, q') \cdot \prod_{k=1}^{K} P(i_{u,v} \mid \mathbf{d}_{u,v} = d_{q'}, d^{c_k}_{u,v})}$$

The behavior of this model as the placement depth $d^c_{u,v}$, curtain thickness $\sigma(u, v, d^c_{u,v})$ and intensity i change is seen in Fig. 7. We observe that low intensities lead to an *inverting gaussian* like weight updates, with a low weight at the light curtain's placement location while other regions get uniform weights. This indicates that the method is certain that an object doesn't exist at the light curtain's location, but is uniformly uncertain about the other un-measured regions. A medium intensity leads due a bimodal gaussian, indicating that the curtain may not be placed exactly on the surface and could be on either side of the curtain. Finally, as the intensity rises, so does weight assigned to the light curtain's placement location.

5. Experiments with Light Curtain only

We first demonstrate depth estimation using just the Light Curtain as described in Sec. 4. In this initial baseline, we track the Uncertainty Field (UF) depth error by computing the RMSE error metric $\sqrt{\sum_{i=1}^{n} \frac{(\mathbb{E}(UF(u,q)) - \mathbf{d_{gt}}(u)_i)^2}{n}}$ against ground truth. We evaluate our method against several outdoor scenarios consisting of vehicles in a scene.

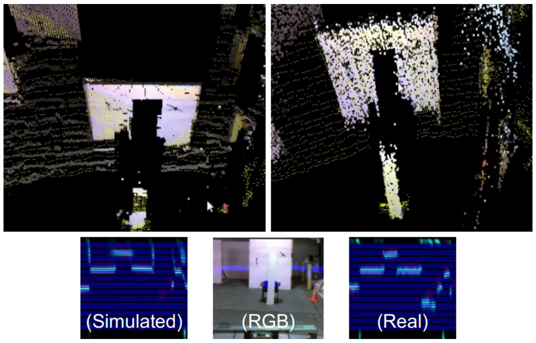

Figure 8: We demonstrate corroboration between simulated and real light curtain device by sweeping several planes across this scene. Colored point cloud is the estimated depth, and lidar ground truth in yellow. **Left:** LC simulated from the lidar depth. **Right:** Using the real device.

Policy	50LC @ 0.25m	25LC @ 0.5m	50LC @ 0.25m	25LC @ 0.5m	12LC @ 1.0m
RMS/m	1.156	1.374	1.284	1.574	1.927
Runtime /s	-	-	2	1	0.5

Table 1: Policy depicts different numbers of light curtains (LC) placed at regular intervals. The first two columns are simulations and the rest are real experiments. Sampling the scene by placing more curtains results in better depth accuracy (lower RMS) at the cost of higher runtime.

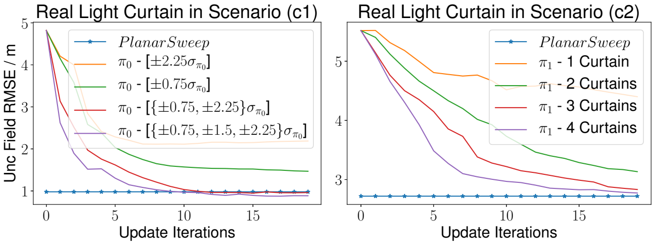

Figure 9: Curtain placement as a function of the Uncertainty Field (UF) converges within a lower number of iterations as opposed to a uniform planar sweep which took 25 iterations

Planar Sweep Curtain Placement: We are able to simulate the light curtain response using depth from LIDAR. A simple fixed policy not adapted to the UF helps validate our sensor model and provides corroboration between the simulated and real light curtains. We perform a uniform sweep across the scene above (at 0.25 to 1.0m intervals) (Fig. 8), incorporating intensity measurements at each pixel for each curtain using our process described earlier. Our simulated device is able to reasonably match the real device, and we also show how sweeping more curtains increases accuracy at the cost of increased runtime (Table. 1).

Policy based Curtain Placement: Sweeping a planar LC can be time consuming (25 iterations), so we want our curtains to be a function of our UF. We evaluated two different scenarios (c1, c2) for each placement policy (π_0, π_1), and we observed that planning and placing curtains as a function of UF results in much faster convergence (Fig. 9).

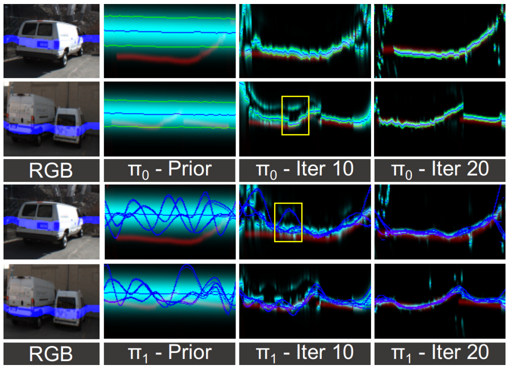

Figure 10: Looking at the top-down Uncertainty Field (UF), we see per pixel distributions in Cyan and the GT in Red. We start with a gaussian prior with a large σ, take measurements and apply the bayesian update, trying both policies π_0 and π_1. Note how measurements taken close to the true surface split into a bimodal distribution (Yellow Box)

6. Depth from Light Curtain + RGB Fusion

While starting from a uniform or Gaussian prior with a large uncertainty is a valid option, it is slow to converge. Furthermore, a light curtain's only means of depth estimation is extracted primarily along the ruled placement of the curtain, at least based on our above placement policies. We would ideally like to use information from a Monocular RGB camera or Stereo Pair to initialize our prior, with a similar DPV representation. For this, a Deep Learning based architecture is ideal, and we also reason that such an architecture could potentially learn to fuse/incorporate information from both modalities better.

6.1. Structure of Network

The first step is to build a network (Fig. 11) that can generate DPV's from RGB images. We extend the Neural-RGBD [16] architecture to incorporate light curtain measurements. Anywhere from 1 to N images, usually two (I_0, I_1), are fed into shared encoders, and the features are then warped into different fronto-parallel planes of the reference image I_0 using pre-computed camera extrinsics $R_{I_o}^{I_1}, t_{I_o}^{I_1}$. Further convolutions are run to generate a low resolution DPV $dpv_{\text{t}}^{\text{l0}}$ [H/4, W/4] where the log softmax operator is applied and regressed on. The transformation between the cameras acts as a constraint, forcing the feature maps to respect depth to channel correspondence. The add operator into a common feature map is similar to adding probabilities in log-space.

$dpv_{\text{t}}^{\text{l0}}$ is then fed into the DPV Fusion Network (a set of 3D Convolutions) that incorporate a downsampled version of $dpv_{\text{t}-1}^{\text{L}}$ along with the the light curtain DPV that we had applied recursive Bayesian updates on $dpv_{\text{t}-1}^{\text{lc}}$, and a residual is computed and added back to $dpv_{\text{t}}^{\text{l0}}$ to generate $dpv_{\text{t}}^{\text{l1}}$ to be regressed upon similarly. With a 30% probability, we train without $dpv_{\text{t}-1}^{\text{lc}}$ feedback by inputting a uniform distribution. Finally, $dpv_{\text{t}}^{\text{l1}}$ is then passed into a decoder with skip connections to generate a high resolution DPV $dpv_{\text{t}}^{\text{L}}$. This is then used to plan and place light curtains, from which we generate a new $dpv_{\text{t}}^{\text{lc}}$ to be fed into the next stage.

6.2. Loss Functions

Soft Cross Entropy Loss: We build upon the ideas in [29] and use a soft cross entropy loss function, with the ground truth LIDAR depthmap becoming a Gaussian DPV with σ_{gt} instead of a one hot vector. This way, when estimating $\mathbb{E}\left(dpv^{gt}\right)$ we get the exact depth value instead of an approximations limited by the depth quantization $\mathcal{D}$. We also make the quantization slightly non-linear to have more steps between objects that are closer to the camera:

$$l_{sce} = \frac{-\sum_i \sum_d \left(dpv^{\{\text{l0,l1,L}\}} * log\left(dpv^{gt}\right)\right)}{n} \tag{8}$$

$$\mathcal{D} = \{d_0, \dots, d_{N-1}\}; d_q = d_{\text{min}} + (d_{\text{max}} - d_{\text{min}}) \cdot q^{pow} \tag{9}$$

L/R Consistency Loss: We train on both the Left and Right Images of the stereo pair whose Projection matrices P_l, P_r are known [11]. We enforce predicted Depth and RGB consistency by warping the Left Depthmap into the Right Camera and vice-versa, and minimize the following metric:

$$D_l = \mathbb{E}\left(dpv_l^L\right) \qquad D_r = \mathbb{E}\left(dpv_r^L\right) \tag{10}$$

$$l_{dcl} = \frac{1}{n}\sum_i \left(\frac{\left|D_{\{l,r\}} - w\left(D_{\{r,l\}}, P_{\{l,r\}}\right)\right|}{D_{\{l,r\}} + w\left(D_{\{r,l\}}, P_{\{l,r\}}\right)}\right) \tag{11}$$

$$l_{rcl} = \frac{1}{n}\sum_i \left(||I_{\{l,r\}} - w\left(I_{\{r,l\}}, D_{\{l,r\}}, P_{\{l,r\}}\right)||_1\right) \tag{12}$$

Edge aware Smoothness Loss: We ensure that neighbouring pixels have consistent surface normals, except on the edges/boundaries of objects with the Sobel operator S_x, S_y via the term:

$$l_s = \frac{1}{n}\sum_i \left(\left|\frac{\partial I}{\partial x}\right| e^{-|S_x I|} + \left|\frac{\partial I}{\partial y}\right| e^{-|S_y I|}\right) \tag{13}$$

7. Light Curtain + RGB Fusion Experiments

We train and validate our algorithms on the KITTI dataset. We then trained the same network by initializing on those weights, but using our custom dataset to evaluate our algorithms with the real sensors on the Jeep.

For evaluation, we consider the RMSE metric against the entire depthmap as opposed to just the Uncertainty Field (UF) as $\sqrt{\sum_{i=1}^{n} \frac{(\mathbb{E}(\mathbf{d}_{u,v}) - \mathbf{d_{gt}}(u,v)_i)^2}{n}}$ against our ground truth.

DPV Prior from RGB: Our first goal is to ensure that our network is capable of generating a reasonable DPV with monocular RGB input, given the above loss functions. We do some simple experiments that explore these effects.

Table 2 shows successively improving performance as we increase σ_{gt}, with poorer performance when the depth

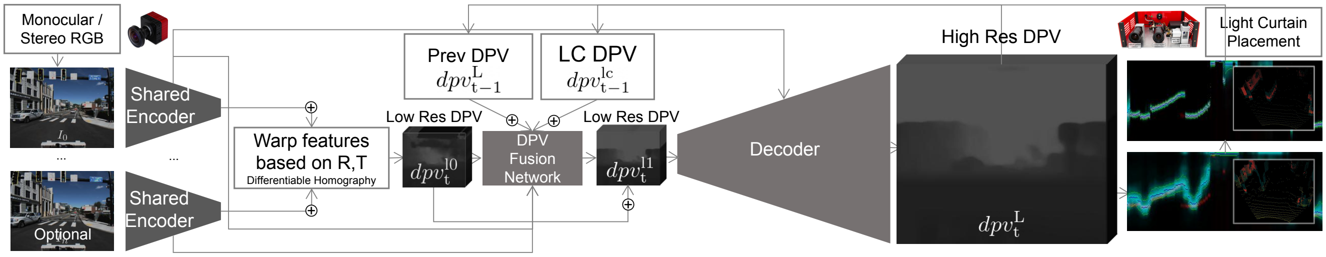

Figure 11: Our Light Curtain (LC) Fusion Network can take in RGB images from a single monocular image, multiple temporally consistent monocular images, or a stereo camera pair to generate a Depth Probability Volume (DPV) prior. We then recursively drive our Triangulation Light Curtain's laser line to plan and place curtains on regions that are uncertain and refine them. This is then fed back on the next timestep to get much more refined DPV estimate.

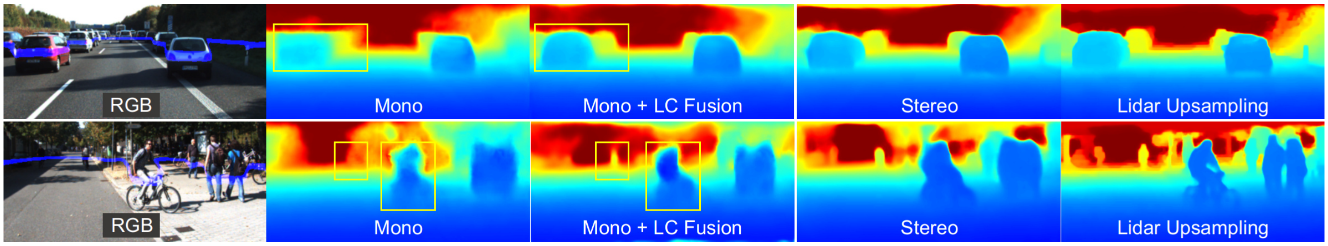

Figure 12: In KITTI + Simulated Light Curtain, we note improved depthmaps when Monocular inputs are fused with Light Curtain inputs. Note the improvements in regions bounded in the yellow box. Our network is also capable of ingesting Stereo inputs, and also solving the task of Lidar Upsampling

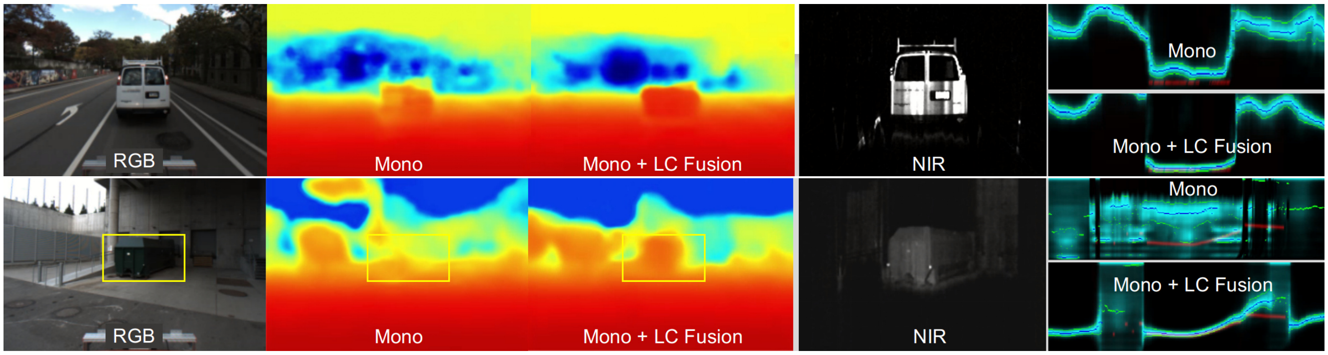

Figure 13: In real world Experiments, we are able to see the monocular scale ambiguity in domain specific scenarios (driving scenario with a van 8m away) get corrected by the Light Curtain, and we are able to see correction in an arbitrary scene (dumpster 15m away) provided to the system as well

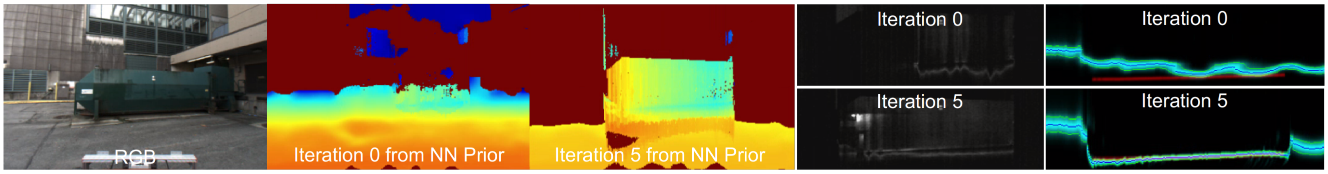

Figure 14: We show the internal state of the bayesian update at Iteration 0 and Iteration 5. Starting with a prior DPV from Monocular Depth estimation, we show the convergence of the sensor's laser and curtain profile on an object 10m away

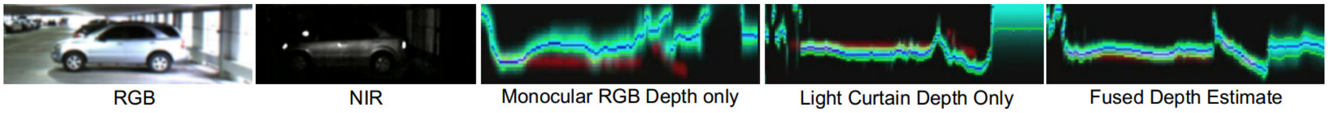

Figure 15: Monocular RGB alone suffers from scale ambiguity but does give an inital uncertain depth estimate on a car 15m away. Iterating on Light Curtain measurements from a mean-centered gaussian prior alone gives a more accurate depth but with a noisy profile, but starting with the RGB DPV results in a more accurate and smoother profile.

Parameters	$\sigma_{gt} = 0.05$	$\sigma_{gt} = 0.2$	$\sigma_{gt} = 0.3$	$\sigma_{gt} = 0.3$ with l_{dcl}, l_{rcl}	$\sigma_{gt} = 0.3$ with l_{dcl}, l_{rcl}, l_s
RMSE/m	3.24	3.16	3.06	2.93	2.90

Table 2: Effects of Soft Cross Entropy (σ_{gt}), Left/Right Consistency (l_{dcl}, l_{rcl}), Smoothness losses (l_s) on Monocular Depth Estimation.

Mono vs Stereo

Mono	2.904
Stereo	1.737

Lidar Upsample with DPV Fusion Network

Without DPV Fusion Network	1.118
With DPV Fusion Network	0.702

Table 3: **Left:** Stereo pair at t instead of Monocular pair at $t, t-1$ input to the network. **Right:** Fusing the GT LIDAR data with dpv_t^{l0} to generate dpv_t^{l1} and dpv_t^{L} with Bayesian inference vs DPV Fusion Network.

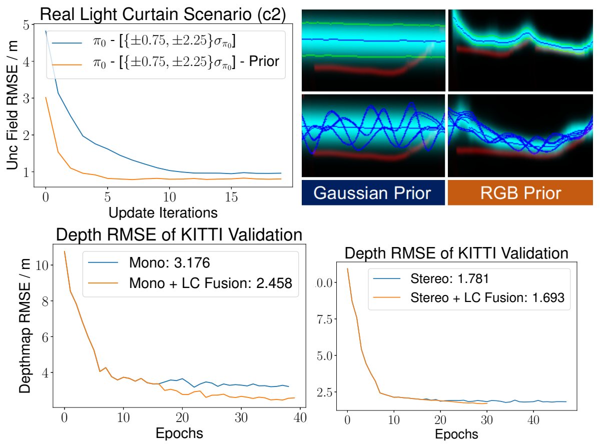

Figure 16: Top: Adaptive depth sensing with the Light Curtain: Starting from a Prior distribution from a Monocular Depth Network as opposed to a gaussian with a large σ leads to faster convergence towards the true depth. Bottom: Monocular (Left) and Stereo (Right) Depth Estimation show improvements when we enabled feedback of the sensed Light Curtain DPV at epoch 16 when training on KITTI dataset with light curtain simulator.

is effectively encoded as a one-hot vector (eg. $\sigma_{gt} = 0.05$), since the depth was more likely to be forced into one of the categories in $\mathcal{D}$. Adding in l_{dcl}, l_{rcl} and l_s improved performance further.

Stereo Inputs: Since our method can generalize to any N camera setup, we compare and contrasted monocular pair inputs at times $t, t-1$, against a stereo pair at time t as input (extrinsics known in both cases). As expected, we note significantly better performance with stereo input (Table 3).

Effect of a Stronger Prior: Previously, we had run our adaptive sensing algorithm from a gaussian prior with a large σ (Fig. 10). In various outdoor experiments, we show that a prior DPV from our network instead, yields higher accuracy and faster convergence towards the true depth (Fig. 15, Fig. 16)(a, b).

DPV Fusion Network: With this corrected DPV, we want to explore how to effectively handle erroneous measurements (due to low light curtain returns etc.), or fuse it other DPVs (from previous frame or from another sensor). With this in mind, we consider the sub task of LIDAR Upsampling. The Velodyne LIDAR in the KITTI dataset, can be converted into a low resolution depthmap, and consequently a low-res DPV we call dpv_t^{gt}. We could then fuse both dpv_t^{l0} and dpv_t^{gt} to generate dpv_t^{l1} using Bayesian inference. Alternatively, we could feed both of those inputs into our DPV Fusion Network, which relies on a series of 3D Convolutions. We note improved performance in this upsampling task using this approach as seen in Table 3.

Light Curtain Fusion Network: Finally, we combine all of these concepts into one. Here, we train our monocular and stereo depth estimation without light curtain feedback, and one where we enable dpv_t^{lc} to be planned and fed-back on the next stage via our DPV Fusion Network, as described in (Fig. 11). Training is done on the KITTI dataset with our light curtain simulator, with a maximum of 5 update iterations for performance and memory reasons. We observed qualitative (Fig. 13) and quantitative (Fig. 16(c, d)) performance improvement of depth with Monocular input, and marginal but visible improvement with Stereo. This is due to the wide baseline of 0.7m in the stereo pair, so we could see that smaller baseline pairs would benefit more with our light curtain measurements.

Performance: Our un-optimized implementation of each planning and curtain placement step takes 40ms. Depth convergence occurs in 5 iterations (5 fps) when starting from a monocular RGB prior and 10 iterations (2.5 fps) with a Gaussian prior. In temporally continuous operations, the prior from $t-1$ reduces convergence to 2 iterations (12.5 fps) depending on the camera motion. A well-engineered implementation could achieve 20-40 fps but much faster motion would require explicitly encoding 3D optical flow.

8. Future Work

We have demonstrated the first known work that has leveraged uncertainty in RGB-based depth estimation to drive an Adaptive Sensor such as a Light Curtain, in the context of ADAS (Fig. 1). Our approach *can generalize* to any sensor that uses the principle of driving a laser or light source to specific pixels that are uncertain, and can benefit from depth uncertainty information at a pixel. Normally non-incident and high reflectively surfaces with poor intensity returns are handled by the scaling factor term $p_{u,v}$ in our model, so we hope to build a better sensor model that utilizes albedo and normal information to predict this better. We could also model scene flow to handle temporally changing scenes (fast moving vehicle).

Project Page: The project page, datasets and code can be found at https://soulslicer.github.io/rgb-lc-fusion/

Acknowledgements: This paper was supported in parts by NSF grants IIS 1900821 and CPS 2038612. The authors thank the NAVLAB at the Robotics Institute for access to the Jeep used in our experiments.

References

[1] Luminar. https://www.luminartech.com/. 2

[2] Ouster. https://ouster.com/. 2

[3] Velodyne. https://velodynelidar.com/. 2

[4] Siddharth Ancha, Yaadhav Raaj, Peiyun Hu, Srinivasa G. Narasimhan, and David Held. Active perception using light curtains for autonomous driving. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 751–766, Cham, 2020. Springer International Publishing. 2, 3

[5] A. W. Bergman, D. B. Lindell, and G. Wetzstein. Deep adaptive lidar: End-to-end optimization of sampling and depth completion at low sampling rates. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11, 2020. 2

[6] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving, 2020. 2

[7] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 1, 2

[8] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps, 2019. 2

[9] D. Gallup, J. Frahm, P. Mordohai, and M. Pollefeys. Variable baseline/resolution stereo. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 2

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency, 2017. 6

[12] Yoav Grauer and Ezri Sonn. Active gated imaging for automotive safety applications. In Robert P. Loce and Eli Saber, editors, *Video Surveillance and Transportation Imaging Applications 2015*, volume 9407, pages 112 – 129. International Society for Optics and Photonics, SPIE, 2015. 2

[13] Tobias Gruber, Frank Julca-Aguilar, Mario Bijelic, Werner Ritter, Klaus Dietmayer, and Felix Heide. Gated2depth: Real-time dense lidar from gated images, 2019. 2

[14] Eddy Ilg, Özgün Çiçek, Silvio Galesso, Aaron Klein, Osama Makansi, Frank Hutter, and Thomas Brox. Uncertainty estimates and multi-hypotheses networks for optical flow, 2018. 2

[15] Whittaker Joe Bartels, Jian Wang and Srinivasa G. Agile depth sensing using triangulating light curtains. In *2019 IEEE International Conference on Computer Vision (ICCV)*, 2019. 1, 2, 3

[16] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa Narasimhan, and Jan Kautz. Neural rgb-¿d sensing: Depth and uncertainty from a video camera, 2019. 1, 2, 6

[17] Abdulla Mohamed, P. Culverhouse, A. Cangelosi, and C. Yang. Active stereo platform: online epipolar geometry update. *EURASIP Journal on Image and Video Processing*, 2018:1–16, 2018. 2

[18] Y. Nakabo, Toshiharu Mukai, Yusuke Hattori, Y. Takeuchi, and N. Ohnishi. Variable baseline stereo tracking vision system using high-speed linear slider. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1567–1572, 2005. 2

[19] M. Nishimura, David B. Lindell, Christopher A. Metzler, and G. Wetzstein. Disambiguating monocular depth estimation with a single transient. In *ECCV*, 2020. 2

[20] Francesco Pittaluga, Zaid Tasneem, Justin Folden, Brevin Tilmon, Ayan Chakrabarti, and Sanjeev Koppal. Towards a mems-based adaptive lidar, 10 2020. 2

[21] A. Schneider, N. Sharma, and Bryan Tripp. Visually guided vergence in a new stereo camera system. 2018. 2

[22] Zaid Tasneem, Dingkang Wang, Huikai Xie, and Koppal Sanjeev. Directionally controlled time-of-flight ranging for mobile sensing platforms. 06 2018. 2

[23] B. Tilmon, E. Jain, S. Ferrari, and S. Koppal. Foveacam: A mems mirror-enabled foveating camera. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11, 2020. 2

[24] Stefanie Walz, Tobias Gruber, Werner Ritter, and Klaus Dietmayer. Uncertainty depth estimation with gated images for 3d reconstruction, 2020. 2

[25] Jian Wang, Joseph Bartels, William Whittaker, Aswin C Sankaranarayanan, and Srinivasa G Narasimhan. Programmable triangulation light curtains. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. 2

[26] Zhihao Xia, Patrick Sullivan, and Ayan Chakrabarti. Generating and exploiting probabilistic monocular depth estimates, 2019. 2

[27] T. Yamamoto, Y. Kawanishi, I. Ide, H. Murase, F. Shinmura, and D. Deguchi. Efficient pedestrian scanning by active scan lidar. In *2018 International Workshop on Advanced Image Technology (IWAIT)*, pages 1–4, 2018. 2

[28] Gengshan Yang, Peiyun Hu, and Deva Ramanan. Inferring distributions over depth from a single image, 2019. 1, 2, 3

[29] Gengshan Yang, Peiyun Hu, and Deva Ramanan. Inferring distributions over depth from a single image. In *Proceedings of (IROS) IEEE/RSJ International Conference on Intelligent Robots and Systems*, November 2019. 6

[30] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo, 2018. 2