Exploiting and Refining Probabilistic Depth Estimates with RGB and Triangulating Light Curtain Fusion

Anonymous CVPR 2021 submission

Paper ID ****

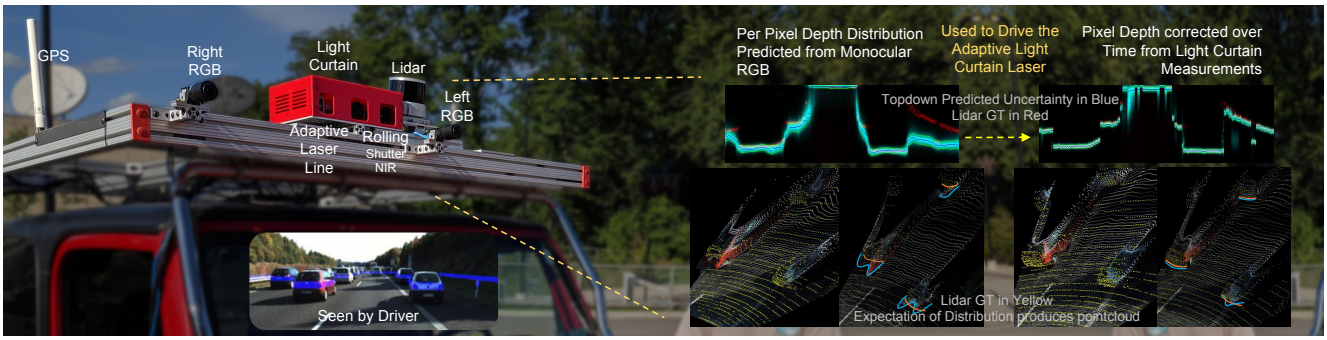

Figure 1: We present a novel method that predicts per pixel depth uncertainity from Monocular/Stereo RGB cameras, uses that information to adaptively place light curtains at regions of low uncertainty, and refines depth estimates over time

Abstract

Programmable Triangulating Light Curtains are a low-cost, high speed and high resolution approach to depth sensing, but require the user to specify a 2.5D ruled surface over where sensing should occur. Objects that intersect this surface show up with a high intensity, while those that don't do not. We have devised a novel algorithm, that exploits prior depth distributions from either Monocular or Stereo RGB cameras to sample depth in regions of the world most uncertain using the Light Curtain. We then incorporate this data to get a refined depth estimate over time. We evaluate this setup with real world real-time experiments such as in the context of Advanced driver-assistance systems (ADAS)

1. Introduction

Programmable Triangulating Light Curtains were introduced last year, and have demonstrated viability in being a low cost (1k$ vs lidar's 25k$), high spatial angular resolution (0.02° vs lidar's 0.4°), and high frame-rate (60fps vs lidar's 20fps) sensor. They have demonstrated having strong returns in the presence of artifacts such as smoke, and also in seeing small targets such as pedestrians further away. They do however, require a user to specify at what depth per pixel one want's sensed, through providing a physically possible 2.5D ruled surface that intersects the world to provide returns.

Conversely, depth estimation through RGB cameras have been a heavily researched area, but do not provide the realiability in estimates as compared to a Lidar, which can be cost prohibitive. This has made both RGB and Lidar's en-masse adoption in safety critical systems such as Advanced driver-assistance systems (ADAS) rare.

Our work, looks into marrying both modalities to counter each sensor's limitations, through the use of Geometric, Probabilistic and Physical constraints. We begin by formulating an iterative Bayesian inference approach to depth sensing using the Light Curtain alone, leveraging on Depth Probability Volume (DPV) and corresponding 2D Uncertainty Field (UF) representations. We then build upon existing planning framework based on Dynamic Programming using the Light Curtain's Galvanometer velocity and acceleration constraints. Lastly, we expand upon Generating and Exploiting Probabilistic Depth Estimates using Deep Learning based approaches, and use that as a prior to drive Light Curtain placement for further refinement.

2. Prior Work

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation

ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. [3] [2]

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. [1]

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

3. Sensor Setup

3.1. Light Curtain Basics

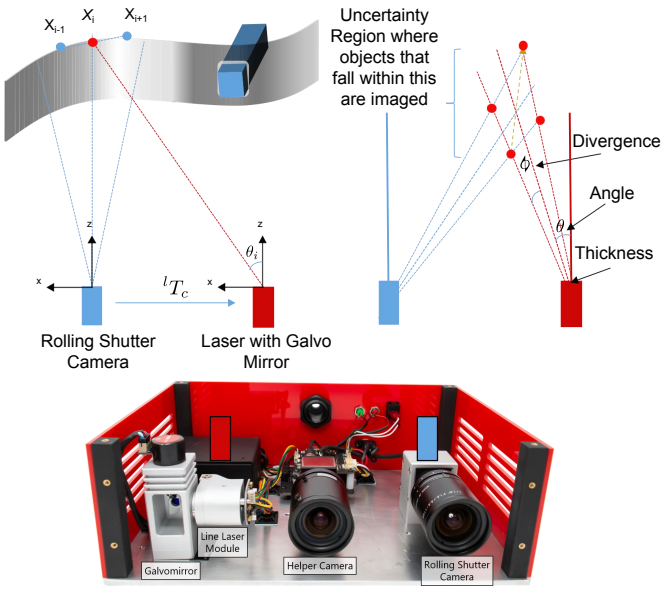

Figure 2: The Programmable Light Curtain device, consisting of a steerable laser, an IR camera and a microcontroller, capable of generating a slice in space to image in 3D

The Light Curtain consists of a rolling shutter NIR camera flipped vertically, a Line Laser module and a Galvomirror. The rolling shutter NIR camera runs in sync with the laser, where each row of the camer can be thought of as a plane with some divergence going out vertically into space. The laser/projector fires a similar vertical sheet of light as a plane into space, which intersects with the camera's vertical plane to produce an intersecting volume. We call this the *thickness* of the curtain. Any objects that lie within this space will then be imaged by the camera. We we will know the 3D position of said objects. Hence, a top-down XZ profile can be provided in the NIR camera's frame, to generate a ruled 3D curtain surface at 50fps

3.2. Simulator

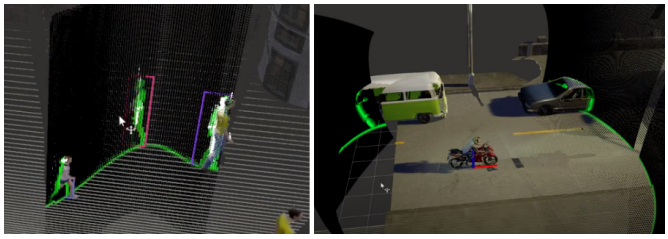

Figure 3: Light Curtain Simulator in CARLA environment

We also designed and open sourced a light curtain simulator to test and evaluate our algorithms, allowing for the Light Curtain paramters such as NIR instrinsics, Laser/NIR extrinsincs, Galvomirror speed etc. to be controlled.

3.3. Sensor Array

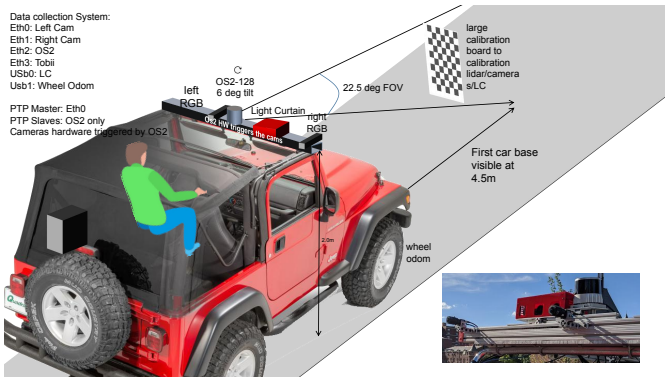

Figure 4: Sensor Array consisting of a Stereo camera pair, the Light Curtain device and a 128 Beam Lidar

Our sensor array consists of a Stereo Camera Pair with a baseline of 0.7m, the Light Curtain device located close to the main/left camera to minimize depth/volume transformation parallax artifacts, and an Ouster OS2-128 Lidar for algorithm accuracy validation and to assist in training depth estimation networks. All sensors are calibrated to the common reference frame of the left RGB camera.

4. Formulation

We begin by looking at a Light Curtain only problem of adaptively discovering the depth of a scene. We wouldn't know the path along the rays/pixels on which an object may lie/intersect, hence we choose to go with a sampling based approach. We treat each ray as initially having either a uniform distribution, or a gaussian with a large variance in the center, and we attempt to formulate our problem with a Recursive Bayesian update approach.

4.1. Representation

Our state space is represented as a tensor of some fixed resolution [320x240] with each pixel in image I encoding the depth value $\mathbf{d}(u,v)$ as a bernoulli distribution $P(\mathbf{d}(u,v))$. D_c represents the list of depths quantizing the space of each pixel defined within (d_{min}, d_{max}) of some fixed size N [64], resulting in a Depth Probability Volume (DPV) tensor of size [64, 240, 320]

$$D_c = \{d_0 ... d_n\} \quad d_i = d_{min} + (d_{max} - d_{min}) * t \quad (1)$$
$$P(\mathbf{d}(u,v)) = I(u,v)$$
$$\sum_d (P(\mathbf{d}(u,v))) = 1 \qquad \mathbb{E}[P(\mathbf{d}(u,v))] = \mathbf{d}(u,v) \quad (2)$$

While an ideal sensor could choose plan a path to sample in the full 3D volume, our Light Curtain device only has control over a collapsed XZ space. Hence, we select a subset of rays that correspond to a plane that we wish to sample, and generate an Uncertainty Field (UF) as follows:

$$P(\mathbf{d}(u)) = \frac{\sum_{u,v} P(\mathbf{d}(u,v)).\mathbf{1}}{\sum_{u,v} \mathbf{1}}$$
$$where\ h_{min} > \mathbb{E}[P(\mathbf{d}(u,v))] > h_{max} \quad (3)$$

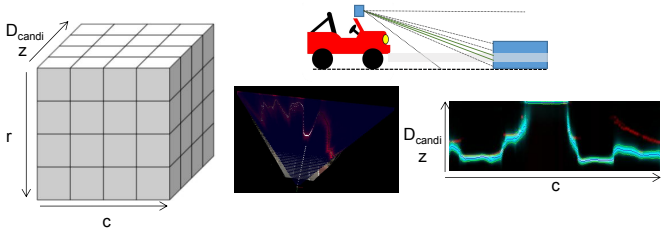

Figure 5: Our state space consisting of a Depth Probability Volume (DPV) its corrsp Bird's Eye Uncertainty Field (UF)

4.2. Curtain Planning

With the Uncertainty Field (UF) extracted from the state space, we can use this to figure out where to place light curtains. We build upon prior work solving Light Curtain placement as a Constraint Optimization and Dynamic Programming problem. A single light curtain placement is defined by a set of T control points $\{\mathbf{X}_t\}_{t=1}^T$. We wish to maximize the objective $J(\mathbf{X}_1, \ldots, \mathbf{X}_T) = \sum_{t=1}^T UF(\mathbf{X}_t)$ where $UF(\mathbf{X})$ is the uncertaintiy field probabilies at the anchor location of $\mathbf{X}$.

The control points $\{\mathbf{X}_t\}_{t=1}^T$, where each $\mathbf{X}_t$ lies on the the camera ray $\mathbf{R}_t$, must be chosen to satisfy the physical constraints of the light curtain device: $|\theta(\mathbf{X}_{t+1}) - \theta(\mathbf{X}_t)| \leq \Delta\theta_{\text{max}}$ with θ_{max} being the maximum angular velocity of the Galvomirror. The problem is also discretized such that $\mathbf{X}_t \in D_c$ and also lies along $\mathbf{R}_t$

$$\arg \max_{\{\mathbf{X}_t\}_{t=1}^T} \sum_{t=1}^T UF(\mathbf{X}_t) \qquad \text{where } \mathbf{X}_t \in D_c$$
$$\text{subject to } |\theta(\mathbf{X}_{t+1}) - \theta(\mathbf{X}_t)| \leq \Delta\theta_{\text{max}},\ \forall 1 \leq t < T \quad (4)$$

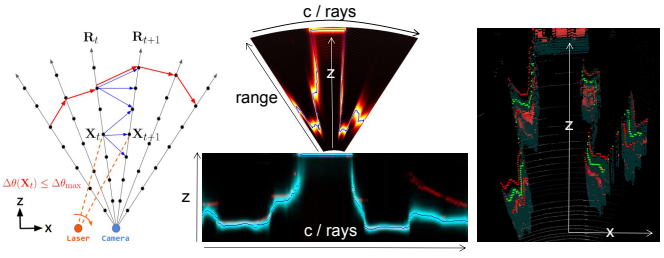

Figure 6: Left: Light Curtain constraint graph subject to max angular velocity of Galvomirror. Right: Placing an optimal curtain along the highest probability region per column of rays

In the figure above, we have placed a single curtain along the highest probability region per column of rays, but ideally, we should place more curtains spanning the uncertaintiy of those distributions. To do this, we generate corresponding entropy fields $H(\mathbf{X})_i$ to be fed to the planner from $UF(\mathbf{X})$ based on two approaches: $m0$ attempts to normalize each ray's distribution $P(\mathbf{d}(u))$ and warp the distribution such that a selected span from the mean is maximized. $m1$ attempts to sample a point on the ray given $P(\mathbf{d}(u))$.

As seen in Fig. 7, strategy $m0$ is able to generate fields that adaptively place additional curtains around a consistent span around the mean, but is unable to do so in cases of multimodal distributions. $m1$ on the other hand is able to place a curtain around the 2nd modality, albeit with a lower probability. The inconsistency from ray to ray in $m1$ however, may be impossible to image due to it exceeding the acceleration bounds, hence a spline fit is used with control points every 5 to 10 rays, resulting in an imagable but non-flat curtain placement. We see the effects of both in later experiments.

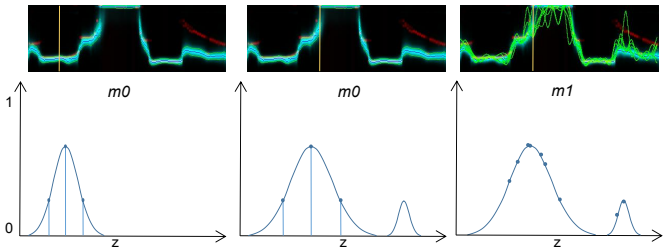

Figure 7: We look at a depth distribution of one of the rays in UF (yellow line), and figure out where additional curtains (blue points) can be placed such as to maximize information gained. Observe that $m1$ is able to handle multimodal distributions

4.3. Observation Model

We now have returns from curtains $\mathbf{C}$ with each $\mathbf{C}_i$ containing $[x, y, z, i]$, the 3D position and intensity value in the same spatial resolution as I, planned based on a particular policy, conditioned on our prior distribution $P(\mathbf{d}_t)$ at time t. We need to convert $\mathbf{C}_i$ into a likelihood distribution $P(\mathbf{c}_t|\mathbf{d}_t)$, such that this Hidden Markov Model representation and sum of log likelihoods holds true:

$$P\left(\mathbf{d}_0, ..., \mathbf{d}_T, \mathbf{c}_1, ..., \mathbf{c}_T\right) = P\left(\mathbf{d}_0\right) \prod_{\mathrm{t=1}}^{\mathrm{T}} P\left(\mathbf{c}_t|\mathbf{d}_t\right) P\left(\mathbf{d}_t|\mathbf{d}_{t-1}\right)$$
$$\log\left(P\left(\mathbf{c}_t|\mathbf{d}_t\right)\right) = \sum_{i=0}^{n} \left(\log\left(P\left(\mathbf{c}_{it}|\mathbf{d}_t\right)\right)\right) \tag{5}$$

We represent $P\left(\mathbf{c}_{it}|\mathbf{d}_t\right)$ as a linear combination of a gaussian and a uniform distribution, where σ is a function of the thickness of the light curtain as described earlier where $t \in [-1..1]$. t is a function of the intensity value i, based on two possible profiles where z either takes a value of 0.5 or 1.0, and m is some control factor we tune:

$$P\left(\mathbf{c}_{it}|\mathbf{d}_t\right) = \frac{\mathcal{N}\left(D_c, \mu_c, \sigma_c\right)(t) + U\left(D_c\right)(1-t)}{\sum\left(\mathcal{N}\left(D_c, \mu_c, \sigma_c\right)(t) + U\left(D_c\right)(1-t)\right)} \tag{6}$$

$$t = \left(\frac{-1}{(z) + (mi)}\right) + 1 \tag{7}$$

As seen in Fig. 8, as the intensity of the light curtain varies from 0 to 1 on the x axis, we vary the value of t on the y axis. When z is 0.5, low intensity returns (eg. $<$ 0.1) results in a negative t value. This means that when the intensity is high, a higher return results in a larger peak probability for both cases when z is 1.0 or 0.5. But when the intensity is low or close to 0, $z = 1.0$ causes $P\left(\mathbf{c}_t|\mathbf{d}_t\right)$ to tend to a uniform distribution, but $z = 0.5$ to tend to an inverted gaussian. With this case, the rationale is that having no return at a location doesn't mean that we have derived no information at all, but rather, we know that this particular location is less likely to have an object and the rest of the the

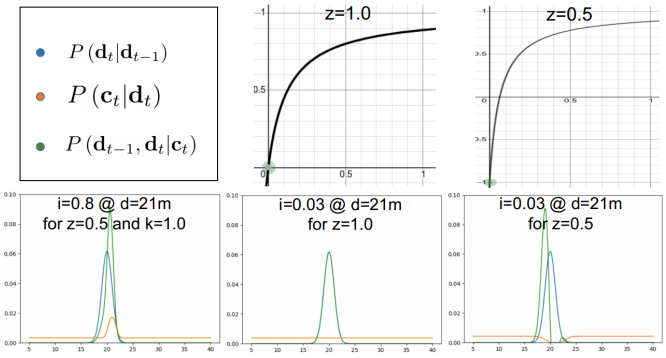

Figure 8: The effect on the posterior when getting a close to 0 intensity return from the Light Curtain when z is either 1.0 or 0.5

locations along the ray are equally uncertain/uniform. We show in experiments that this results in significantly faster convergence, and we call this the *Inverting Gaussian model*

5. Light Curtain only Experiments

In this initial baseline, we attempt to track the Uncertainty Field (UF) depth error by computing the RMSE error metric $\sqrt{\sum_{i=1}^{n} \frac{(\mathbb{E}(P(\mathbf{d}(u)_i)) - \mathbf{d_{gt}}(u)_i)^2}{n}}$ against the ground truth. We use 3 scenarios: One in a KITTI driving scene using the LC simulator (a), one indoors in the basement using the both a simulated and real Light Curtain (b), and lastly in various outdoor driving scenes with the real device (c0/c1/c2).

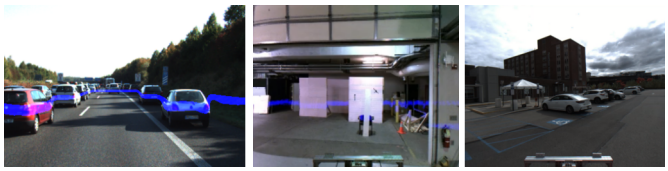

Figure 9: Scenarios (a), (b), (c) from left to right

Planar Sweep: A simple sanity check between simulation and the real sensor involves involves performing a uniform sweep across the scene in (b).

As seen in Fig. 10, our simulated Light Curtain (LC) is able to reasonably match the real device. We also demonstrate how decreasing the steps the curtain takes reduces runtime but increases RMSE. Results seen in 1

Effect of Dynamic Sigma: Earlier, we had noted how σ_c defined for each C_i measurement in $P\left(\mathbf{c}_t|\mathbf{d}_t\right)$ is a function of the thickness of the curtain. We also experiment by making σ_c fixed. We observe that σ_c being a function of the curtain thickness is critical to better performance over

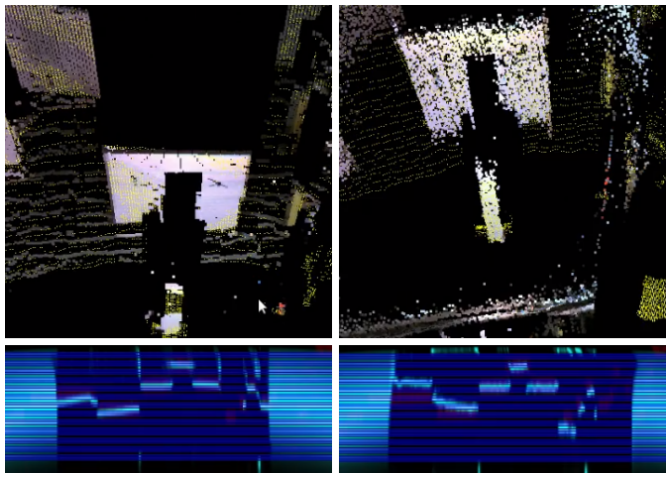

Figure 10: Doing a simple planar sweep across the scene. Colored pointcloud is the estimated depth, and Lidar ground truth in yellow. Left: LC simulated from the Lidar Depth. Right: Using the real LC

Policy	Runtime/s	RMSE/m
Sweep 50 **C** Step 0.25m (Sim)	-	1.156
Sweep 25 **C** Step 0.5m (Sim)	-	1.374
Sweep 50 **C** Step 0.25m (Real)	2	1.284
Sweep 25 **C** Step 0.5m (Real)	1	1.574
Sweep 12 **C** Step 1.0m (Real)	0.5	1.927

Table 1: Error wrt to time and number of Sweep steps

larger steps/placements. Results seen in 2

Policy	Runtime/s	RMSE/m
Sweep **C** Step 0.25m (Dyn)	2	1.276
Sweep **C** Step 0.5m (Dyn)	1	1.532
Sweep **C** Step 1.0m (Dyn)	0.5	2.013
Sweep **C** Step 0.25m (Fixed)	2	1.218
Sweep **C** Step 0.5m (Fixed)	1	1.658
Sweep **C** Step 1.0m (Fixed)	0.5	2.290

Table 2: σ_c in generated $P(\mathbf{c}_t|\mathbf{d}_t)$ being fixed vs being dynamic as a function of curtain thickness with real LC

Effect of Inverting Gaussian Model: Our Observation Model ensures that the sensor distribution tends to an inverted gaussian when intensities are low. We test the effect on (a) and (b) in the above case, and in cases where low intensities tend to a uniform distribution. We see significantly improved performance when low intensities tend to an Inverted Gaussian rather than a uniform distribution. Results seen in 11 12

Effect of Placing more Curtains: In both policies, placing more light curtains results in much faster convergence, at the cost of increased runtime. Results seen in 13

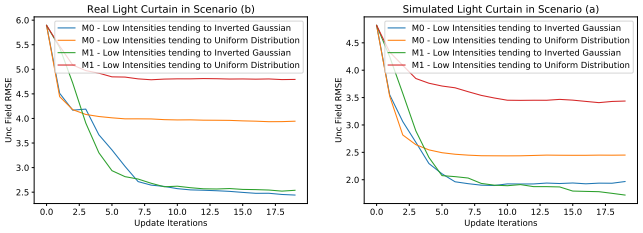

Figure 11: RMSE of Depth in UF over every iteration in scenarios left/(a) right/(b). Note the largely improved performance when low intensities tend to an Inverted Gaussian

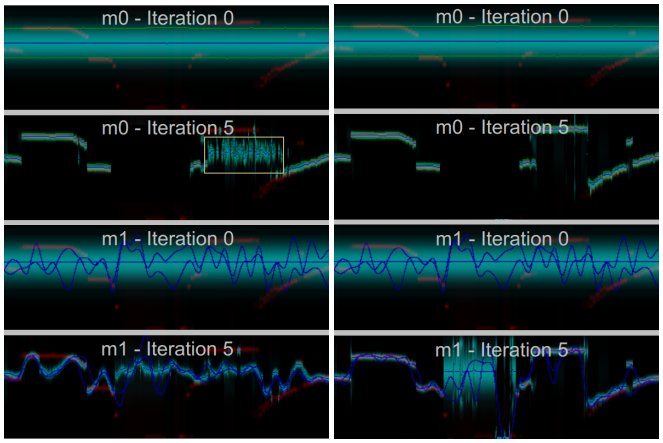

Figure 12: **Left:** Policies $m0$ and $m1$ where low intensities result in no information (Uniform Distribution). **Right:** Where low intensities result in an Inverted Gaussian based on our Sensor Model

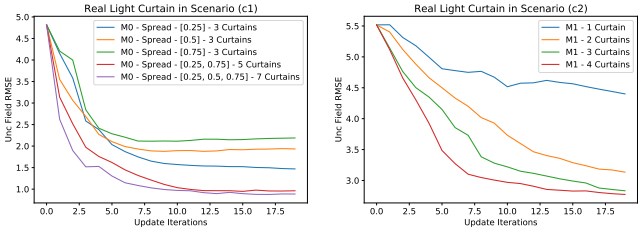

Figure 13: Placing more curtains results in faster convergence to the ground truth

6. Network Architecture

6.1. Motivation

Effect of a Stronger Prior: While starting from a uniform or gaussian prior with a large uncertainity is a valid option, it is slow. As a quick primer, we show that a prior generated from a monocular RGB camera in a domain specific scenario yields better results. Results seen in 15

Hence, we need to work towards building a Neural Net-

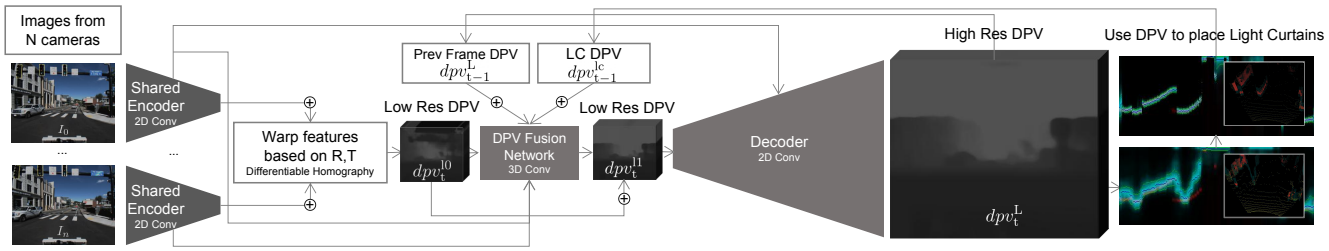

Figure 14: Our network takes in RGB images to generate a Depth Probability Volume, plans and places Light Curtains based on this DPV, and recursively refines it. This is then fed back on the next timestep to get much more refined DPV estimate.

work that is capable of ingesting either Monocular/Stereo images, and generating a DPV to act as our prior. Additionally, a network that ingests a DPV from Light Curtain measurements is also ideal.

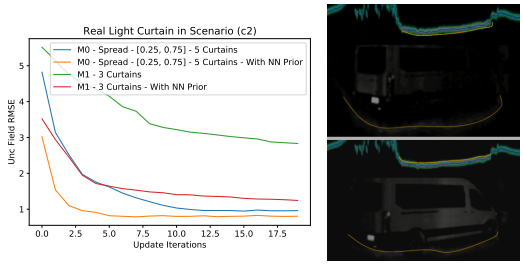

Figure 15: **Right:** Starting from a Prior distribution from a Monocular Depth Network leads to faster convergence. **Left:** Policy $m0$ placing curtains along 3 Locations of a prior distribution from a Monocular Depth Network as seen in the Light Curtain's NIR image.

6.2. Structure of Network

The first step is to build a network that can generate DPV's similar to our light curtain only estimation strategy from RGB images. To this end, we build upon the MVS-Net/PSMNet architecture. N images, usually 2 I_0, I_1 are fed into encoders that share weights, and the features are then warped into different fronto-parallel planes of the reference image I_0 using pre-computed R_i^0, t_i^0. Further convolutions are run to generate a low resolution DPV dpv_t^{l0} [80, 96] where the log softmax operator is applied and regressed on. The transform between the camera's act as a constraint, forcing the feature maps learnt to respect depth to channel correspondence. The add operator into a common feature map is analagous to adding probabilities in log-space.

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

6.3. Loss Functions

Soft Cross Entropy Loss: We build upon the ideas found in [4] 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 taking $\mathbb{E}\left(dpv^{gt}\right)$ we get the exact depth value instead of an approxmiation limited by the depth quantization D_c. We also make the quantization slightly non-linear to have more steps between objects that are closer to the camera.

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

$$D_c = \{d_0, ...d_n\} \quad d_i = d_{min} + (d_{max} - d_{min}) * t^{pow} \quad (9)$$

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

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

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

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

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:

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

6.4. Datasets

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 ILIM dataset to evaluate our algorithms on our sensor platform / jeep.

7. Experiments

Effects of our loss function: We do some simple experiments, training the task of monocular depth estimation, and we explore the effects of enabling/disabling the various loss functions:

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

Table 3: Effects of various loss functions for the baseline of Monocular Depth Estimation only

We note succesively improving performance as we increase σ_{gt}, with poorer performance when the depth 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 D_c. Adding in l_{dcl}, l_{rcl} and l_s improved performance further, but we did not see any significant performance improvement in varying t^{pow} especially once σ_{gt} was increased.

References

[1] Ancha, Raaj, Hu, Narasimhan, and Held. Active perception using light curtains for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2020. 2

[2] 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. 2

[3] 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

[4] 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