2017 IEEE Conference on Computer Vision and Pattern Recognition

Dynamic Time-of-Flight

Michael Schober
Max Planck Institute for
Intelligent Systems

Amit Adam[*] Omer Yair[*] Shai Mazor[*]

Sebastian Nowozin
Microsoft Research

Abstract

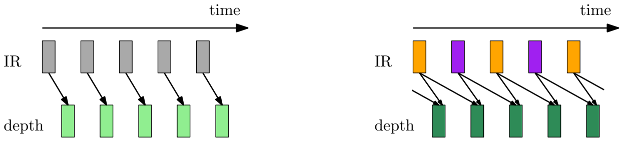
Time-of-flight (TOF) depth cameras provide robust depth inference at low power requirements in a wide variety of consumer and industrial applications. These cameras reconstruct a single depth frame from a given set of infrared (IR) frames captured over a very short exposure period. Operating in this mode the camera essentially forgets all information previously captured - and performs depth inference from scratch for every frame. We challenge this practice and propose using previously captured information when inferring depth. An inherent problem we have to address is camera motion over this longer period of collecting observations. We derive a probabilistic framework combining a simple but robust model of camera and object motion, together with an observation model. This combination allows us to integrate information over multiple frames while remaining robust to rapid changes. Operating the camera in this manner has implications in terms of both computational efficiency and how information should be captured. We address these two issues and demonstrate a realtime TOF system with robust temporal integration that improves depth accuracy over strong baseline methods including adaptive spatio-temporal filters.

1. Introduction

Current time-of-flight depth cameras operate by capturing a set of RAW intensity frames under active illumination in the infrared band. A depth algorithm then combines multiple captured RAW frames to produce a single depth image [17]. Since both the camera and objects in the scene may move, the cameras are inherently designed to collect the set of IR frames over a very short exposure period. Likewise, the standard algorithms do not use previously captured frames because compensating for motion is not straightforward.

If the scene and camera were static we could obtain more accurate estimates by using a larger number of IR frames captured over longer periods of time since then our signal will be stronger and we integrate more light. In addition, if our hardware allows this, we will also be able to use different active measurement patterns at different frames. This can enable better depth disambiguation.

In this paper we propose a realtime solution for integrating time-of-flight observations over large time periods in the presence of object and camera motion. Figure 1 illustrates our approach. The standard "static" approach is illustrated in Figure 1a, where depth is computed from the measured RAW intensities through simple phase-space methods [17, 16] for modulated TOF, or via a simple generative model [1] for pulsed TOF. All current depth cameras are *forgetful* because they discard previously captured information and use only the most recent RAW frames, as shown in Fig. 1a.



(a) Static TOF camera

(b) Dynamic TOF camera

Figure 1: Basic idea of the proposed method: (a) an ordinary (static) time-of-flight camera captures a set of infrared (IR) frames using the same active measurement pattern (grey) in each time step and uses the captured information only once, in the current time step, to reconstruct depth; (b) the proposed *dynamic* time-of-flight camera uses different measurement patterns in each time step (in this case two, shown in orange and purple) and integrates multiple measurements in time (additional arrows) to improve the depth accuracy.

But RAW frames that were captured in the near past *do* contain information about the scene depth, even when objects or the camera are moving. We propose a model to use this information across frames to improve depth accuracy.

On the right, Figure 1b illustrates the novel *sensing* and *inference* framework we describe in this paper. *Inference* is done using both current frame's IR images, and the previous frame's IRs, as shown by the additional arrows. *Sensing* is more flexible and may be done using different active measurement patterns, as shown by the orange and purple boxes. We obtain the added flexibility in sensing and the ability to use previous observations by a probabilistic generative model combining *motion* and *observation*.

We remark that the (hardware) ability to change active measurement patterns is not theoretical. It is common to distinguish between *modulated* TOF (e.g. [3, 38]) and *pulsed* TOF technologies (e.g. [40, 1]). In the former both the illumination and the integration profiles are sinusoidal, and changing the measurement pattern is not straightforward. In pulsed TOF, the integration profile is more flexible and may indeed be changed between different frames. In this work we use a pulsed TOF camera, similar to the one used by [1].[1] This device enables us to toggle between two different active

[*] A. Adam, O. Yair, and S. Mazor were with Microsoft AIT, Haifa, Israel. {email.amitadam, omeryair, smazor.shai}@gmail.com.

[1] The hardware details of this camera are described in [40, 12, 42].

1063-6919/17 $31.00 © 2017 IEEE
DOI 10.1109/CVPR.2017.26

measurement patterns at 30Hz.

Computationally there is no difference between modulated and pulsed time of flight. Following [1], the so-called response curve (see Section 2) may either be sine-like (modulated TOF) or more general (pulsed TOF). Our approach is general and handles both technologies seamlessly.

Contributions. In this paper we make the following contributions over the state-of-the-art in TOF imaging:

- We formulate a generative model for TOF observations with camera/object motion;
- We perform principled Bayesian inference on this model to obtain improved depth estimates using IR images captured over several frames;
- We design the active measurement patterns such that we collect complementary information over time;
- Using a regression approach we demonstrate the first realtime temporal integration of TOF sensor data, at low compute and memory budget.

2. Bayesian Time-of-Flight

In this section we give a brief summary of the Bayesian time-of-flight model proposed in [1]. This model is a probabilistic generative model $P(\vec{R}, \vec{\theta})$ that for each pixel relates unknown imaging conditions $\vec{\theta}$ to an observed response vector $\vec{R}$. In the basic version of the model proposed in [1] the imaging conditions correspond to the depth t, the effective reflectivity (albedo) ρ, and an ambient light component λ which illuminates the imaged surface. We write $\vec{\theta} = (t, \rho, \lambda)$ for all the unknown imaging conditions. We specify the generative model as

$$\begin{aligned} \vec{\theta} &\sim P(\vec{\theta}), && \text{(prior)} && (1)\\ \vec{\mu} \mid \vec{\theta} &= \rho \vec{C}(t) + \rho \lambda \vec{A}, && \text{(SP mean response)} && (2)\\ \Sigma \mid \vec{\mu} &= \text{diag}(\alpha \vec{\mu} + K), && \text{(noise)} && (3)\\ \vec{R} \mid \vec{\mu} &\sim \mathcal{N}(\vec{\mu}, \Sigma). && \text{(observed response)} && (4) \end{aligned}$$

In (3) of the above model α and K correspond to sensor-specific constants describing *shot noise* and *read noise*, respectively [13]. In (2) we use $\vec{C}$ and $\vec{A}$ to describe the noise-free camera response as follows. For a given pixel, the function $\vec{C} : [t_{\text{min}}, t_{\text{max}}] \to \mathbb{R}^n$ maps a depth t to an ideal responses vector that would be observed on a surface with 100 percent reflectivity and no ambient light. Likewise the vector $\vec{A} \in \mathbb{R}^n$ corresponds to the response caused by one unit of ambient light on such ideal surface.

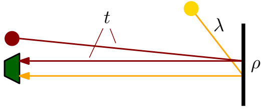


Figure 2: Model (2).

Because light is additive, we can see in (2) that the active illumination response $\vec{C}(t)$ and ambient light response $\lambda\vec{A}$ are summed and scaled by the surface reflectivity ρ, as shown in Figure 2. A more detailed derivation of the model (1)–(4) is available in [1]. The model is named "single-path (SP) model" because it describes the direct single light path response at a surface.

Inference in the above model corresponds to estimating the posterior distribution, using Bayes rule [4],

$$P(\vec{\theta}|\vec{R}) \propto P(\vec{R}|\vec{\theta})\, P(\vec{\theta}). \qquad (5)$$

The difficulty in this inference problem is the nonlinear function $\vec{C}(t)$ and in [1] the authors proposed a solution based on importance sampling, but in general other approximate inference methods could be used. We now describe a variation of the above model.

2.1. Modeling Multipath

In (2) we model the response due to the direct reflection of the emitted light. In real scenes *multipath* effects due to indirect light corrupts the observation [17].

Based on [14] the authors of [1] consider a simple "two-path" (TP) model for multipath effects. The only change in model (1)–(4) is to substitute (2) with

$$\vec{\mu} \mid \vec{\theta} = \rho \left(\vec{C}(t) + \rho_2\, \vec{C}(t_2) + \lambda\, \vec{A} \right), \qquad \text{(TP mean)} \qquad (6)$$

where the imaging conditions $\vec{\theta} = (t, \rho, \lambda, t_2, \rho_2)$ now contain the parameters of a second surface at depth t_2 with reflectivity ρ_2, as shown in Figure 3. In [1] this model improved depth accuracy in realistic scenes. This simple multipath model is readily supported by our proposed dynamic time-of-flight model.

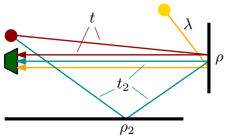


Figure 3: Model (6).

3. Dynamic Time-of-Flight

We now extend the Bayesian time-of-flight model to include temporal dynamics. To introduce temporal dependencies, we first introduce a time index s to the observation vector $\vec{R}^{(s)}$, and also assume the unknown imaging conditions to be time-dependent as $\vec{\theta}^{(s)}$. For the temporal dynamics we use the formalism of *state space models* (SSM) [11], also known as general hidden Markov models (HMM). We assume Markovian dynamics on the sequence of imaging conditions for each spatial location $\vec{\theta}^{(s)}$. For one pixel this gives a joint distribution over S frames as

$$\begin{aligned} \vec{\theta}^{(s)} | \vec{\theta}^{(s-1)} &\sim P(\vec{\theta}^{(s)} | \vec{\theta}^{(s-1)}), && \text{(temporal dynamics)} && (7)\\ \vec{R}^{(s)} \mid \vec{\theta}^{(s)} &\sim P(\vec{R}^{(s)} \mid \vec{\theta}^{(s)}). && \text{(observation model)} && (8) \end{aligned}$$

Equivalently, we can write (7)–(8) in HMM form,

$$P(\vec{R}^{(1:S)}, \vec{\theta}^{(1:S)}) = \underbrace{\prod_{s=1}^{S} P(\vec{R}^{(s)} | \vec{\theta}^{(s)})}_{\text{(A)}} \underbrace{\prod_{s=1}^{S-1} P(\vec{\theta}^{(s+1)} | \vec{\theta}^{(s)})}_{\text{(B)}} P(\theta^{(1)}). \quad (9)$$

In the above equation we have used the notation $1:S$ to describe the set of integers between one and S such that $\vec{\theta}^{(1:S)}$ is the tuple $(\vec{\theta}^{(1)}, \dots, \vec{\theta}^{(S)})$; this notation is commonly used in the literature on SSMs. (A) in (9) corresponds to the *observation model* which is identical to the per-frame Bayesian model (1)–(4). The part (B) is the model of the temporal dynamics which couples multiple frames together. We will describe this dynamics model in more detail below. For the choice $S = 1$ we recover the original static model in which only the most recent observation together with the prior is used for inferring $\vec{\theta}$. We now describe a model of temporal change for the evolving imaging conditions.

The temporal model $P(\vec{\theta}^{(s+1)}|\vec{\theta}^{(s)})$ is a critical component in our approach; if we manage to accurately describe the evolution over time of imaging conditions at each pixel then the temporal model leads to statistically more efficient use of the available observations and consequently to improved depth estimates. However, there is a risk in making temporal assumptions well known in the visual tracking literature [43]: if the temporal assumptions are too strong they may override evidence present in the observation likelihood leading to systematic bias, drift, or—in the case of tracking—being "stuck in the background". To prevent this we propose a general way to *robustify* the temporal dynamics.

3.1. Robust Temporal Dynamics

To robustify a given temporal model Q, we propose a temporal model of the following mixture form.

$$P(\vec{\theta}^{(s+1)} \mid \vec{\theta}^{(s)}) = \omega\, P(\vec{\theta}^{(s+1)}) + (1-\omega)\, Q(\vec{\theta}^{(s+1)} \mid \vec{\theta}^{(s)}). \quad (10)$$

Here $P(\vec{\theta}^{(s+1)})$ is the independent prior as in (1), and Q is a given, generally simpler, temporal model. The *mixture weight* $\omega \in [0, 1]$ blends between an independent model ($\omega = 1$) and the dynamics described by Q ($\omega = 0$).

Intermediate values of ω lead to a robust temporal model in the sense that if two observations $R^{(s)}$ and $R^{(s+1)}$ differ sufficiently strongly, then the model can fall back on the independent model, explaining each observation separately. The robustness can also be seen from the observation that the mixture model (10) will usually have heavier tails in $\vec{\theta}^{(s+1)}$ compared to the simpler model Q because the prior is defined over the full domain. We will verify the claimed robustness experimentally in Section 6.4.

To complete the description of our motion model (10) we now describe the derivation of Q that we use.

3.2. Specification and Empirical Prior

The change in imaging conditions depends on two factors, the *camera trajectory* and the *scene*. To find suitable priors, we will use empirical data.

To understand typical camera trajectories we use eleven handheld camera trajectories from the SLAM benchmark [39]. We resample the trajectories from their original 100Hz to 30Hz, our target frame rate, yielding 32k six-dimensional camera motion vectors. We then approximately fit Normal distributions to the change in camera translation and rotation, which all have a mean change of zero. For frame-to-frame translation we obtain standard deviations $\sigma_{tx} = \sigma_{tz} = 0.004$m for the horizontal and forward-backward motion and $\sigma_{ty} = 0.001$m for the vertical motion. For rotation we obtain standard deviations $\sigma_{rp} = \sigma_{ry} = 0.0075$rad for pitch and yaw, and $\sigma_{rr} = 0.003$rad for roll.

To understand scene geometry we leverage the rendering simulation approach proposed in [1]. Specifically we use the same five scenes used in [1] and the camera motion model just discussed to randomly sample pairs of frames with simulated camera motion. This gives us a pair of ground truth depth frames, together with ground truth reflectivity and ambient components. For each of the five scenes we randomly sample ten pairs, yielding a total of 50 pairs of simulated ground truth frames and a total of 10.6M sampled pairs of imaging conditions $(\theta_i^{(s+1)}, \theta_i^{(s)})$ to estimate the prior $Q(\vec{\theta}^{(s+1)} \mid \vec{\theta}^{(s)})$ from.

By inspecting the empirical histograms of the change in imaging conditions, shown in Figure 4, for a 30Hz handheld camera we propose the following temporal model for the imaging conditions at each sensor pixel.

$$t^{(s+1)} \mid t^{(s)} \sim \text{Laplace}(\mu = t^{(s)},\, b = 0.75), \quad (11)$$

$$f_\rho \sim \text{Laplace}(\mu = 1, b = 0.15), \quad (12)$$

$$\rho^{(s+1)} \mid \rho^{(s)} = f_\rho\, \rho^{(s)}, \quad (13)$$

$$f_\lambda \sim \text{Laplace}(\mu = 1, b = 0.25), \quad (14)$$

$$\lambda^{(s+1)} \mid \lambda^{(s)} = f_\lambda\, \lambda^{(s)}. \quad (15)$$

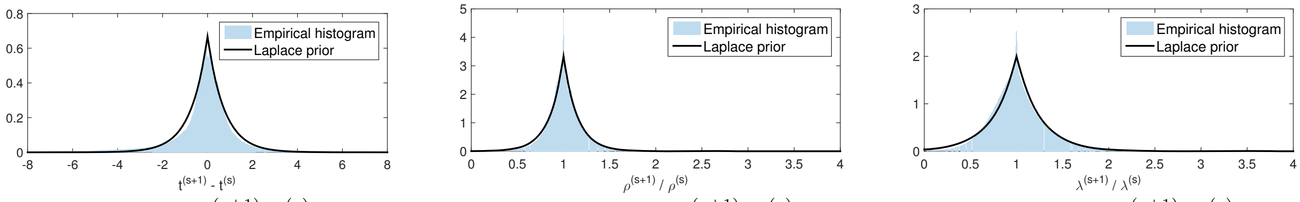
Here the *Laplace distribution* [22, 23] has the probability density function $\text{Laplace}(x; \mu, b)$ which is equal to $\exp(-|x-\mu|/b)/(2b)$. Whereas the magnitude of the depth change does not depend on the current depth, the change in the reflectivity and ambient light are best modelled via a multiplicative change, as we illustrate in Fig. 4.

The above is the prior for the single-path model where $\vec{\theta} = (t, \rho, \lambda)$. For the two-path model we believe t_2 and ρ_2 to qualitatively behave as t and ρ and therefore select identical priors, $t_2^{(s+1)} \mid t_2^{(s)} \sim \text{Laplace}(\mu = t_2^{(s)},\, b = 0.75)$, and $f_{\rho_2} \sim \text{Laplace}(\mu = 1, b = 0.15)$, with $\rho_2^{(s+1)} \mid \rho_2^{(s)} = f_{\rho_2}\, \rho_2^{(s)}$. We now describe how the model (11)–(15) is used for joint temporal inference.

3.3. Dynamic Depth Inference

In general, inference in non-linear state space models such as (9) is difficult and computationally demanding, typically requiring sequential Monte Carlo approximations [8, 10] or Markov chain Monte Carlo (MCMC) approximations [28].

For this reason, we approximate (9) as follows. We fix S to a small number and only consider the limited past as



(a) Depth $P(t^{(s+1)} \mid t^{(s)})$ (b) Reflectivity $P(\rho^{(s+1)} \mid \rho^{(s)})$ (c) Ambient $P(\lambda^{(s+1)} \mid \lambda^{(s)})$

Figure 4: Prior distributions estimated from empirical data: (a) the change of depth prior is an offset to the previous $t^{(s)}$ with the magnitude of change being independent of the current depth; (b) the change in reflectivity is multiplicative; (c) like the reflectivity the change in ambient light is multiplicative.

described by the most recent S observations $\vec{R}^{(1:S)}$. This is an approximation because it assumes that the influence of past observations decays quickly enough with time such that after S frames we can ignore these old measurements [28].

In this approximate setting we are given, for each pixel, a fixed length sequence of measurement vectors $\vec{R}^{(1:S)} = (\vec{R}^{(1)}, \dots, \vec{R}^{(S)})$, and we would like to infer the posterior distribution over imaging conditions, given by Bayes rule as

$$P(\vec{\theta}^{(1:S)} \mid \vec{R}^{(1:S)}) \propto P(\vec{\theta}^{(1:S)}, \vec{R}^{(1:S)}). \tag{16}$$

For the static single frame case ($S = 1$) the authors of [1] proposed a solution based on importance sampling [26, 33]. We found this solution difficult to extend to the dynamic case (16) due to the higher dimensionality of our problem [27]. Instead we propose to perform posterior inference using Markov chain Monte Carlo (MCMC) [6].

3.4. Markov Chain Monte Carlo Approximation

Intuitively MCMC works as follows: we start with an initial state $\vec{\theta}^{(1:S)}$, perhaps sampled from our prior. We then iteratively and randomly perturb this state via a Markov chain in such a manner that it eventually will be distributed according to (16). By sampling from the posterior in such an iterative fashion we can generate a correlated sequence of posterior samples which we then use to summarize the posterior, for example, to make a point prediction for depth. We use the *Metropolis-Hastings* (MH) chain construction [18] and found this method simple to implement and reliable. We provide additional validation in the supplementary materials.

To apply MCMC sampling we need to specify the MH transition kernel that we use. We use a mixture kernel, picking a random $s \in \{1, \dots, S\}$, then performing at random one of the following proposal perturbations.

$$t'^{(s)} \sim \mathcal{N}(\mu = t^{(s)}, \sigma = 10\text{cm}), \tag{17}$$

$$\rho'^{(s)} \sim \mathcal{N}(\mu = \rho^{(s)}, \sigma = 0.1), \tag{18}$$

$$\lambda'^{(s)} \sim \mathcal{N}(\mu = \lambda^{(s)}, \sigma = 1), \tag{19}$$

$$\vec{\theta}'^{(1:S)} \sim P(\vec{\theta}^{(1:S)}). \tag{20}$$

The first three kernels (17)–(19) make a small modification to a single imaging condition in the s'th frame. The last kernel (20) is an independent prior kernel, which is used to escape local maxima of the likelihood surface [25].

Each of the above proposals is of the general form $\vec{\theta}'^{(1:S)} \sim W(\vec{\theta}'^{(1:S)} \mid \vec{\theta}^{(1:S)})$. We accept or reject each proposed perturbation $\vec{\theta}'^{(1:S)}$ with probability $a(\vec{\theta}^{(1:S)} \to \vec{\theta}'^{(1:S)}) = \min\{1, \bar{a}\}$ according to the MH rule [18], where

$$\bar{a} = \frac{W(\vec{\theta}^{(1:S)} \mid \vec{\theta}'^{(1:S)})\, P(\vec{\theta}'^{(1:S)} \mid \vec{R}^{(1:S)})}{W(\vec{\theta}'^{(1:S)} \mid \vec{\theta}^{(1:S)})\, P(\vec{\theta}^{(1:S)} \mid \vec{R}^{(1:S)})}. \tag{21}$$

At all times the imaging conditions $\vec{\theta}^{(1:S)}$ are constrained to a box region specified by $t^{(s)} \in [80\text{cm}, 550\text{cm}]$, $\rho^{(s)} \in [0, 1]$, $\lambda^{(s)} \in [0, 10]$ for the SP model and additionally $t_2^{(s)} \in [t, t + 150\text{cm}]$ and $\rho_2^{(s)} \in [0, 2]$ for the TP model.

While MCMC is fast enough to perform offline experiments it is not suitable for realtime operation. We now describe how we achieve realtime performance.

4. Realtime Inference

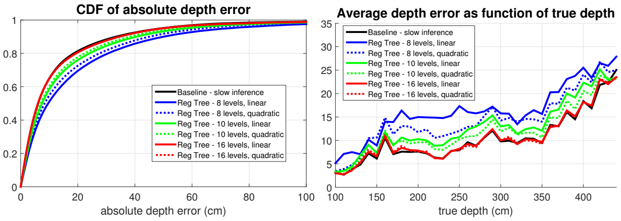
For realtime inference we use an expensive offline preprocessing step and a cheap and efficient model at runtime. This was first proposed for TOF imaging in [1] for the static case. However, it is not clear that this approach extends to the higher-dimensional problem of predicting depth from multiple frames. In the offline step we repeatedly perform the following:

1. We sample $\vec{\theta}^{(1:S)}$ and $\vec{R}^{(1:S)}$ from the prior.
2. We perform MCMC to obtain the posterior mean $\hat{\theta}^{(1:S)}(\vec{R}^{(1:S)})$.
3. We store the pair $(\vec{R}^{(1:S)}, t^{(S)})$, where $t^{(S)}$ is the most recent depth estimate.

By repeating this procedure we collect a large number (several millions) of inference inputs and outputs. We then train a least squares regression tree [5] model f on a quadratic feature expansion of $\vec{R}^{(1:S)}$ to predict the scalar $t^{(S)}$.

At runtime we observe a response $\vec{R}^{(1:S)}$ for each pixel and evaluate the tree $f(\vec{R}^{(1:S)})$ to estimate depth.

As noted in [1], regression trees are a good choice for a regression mechanism for several reasons. First, they scale well with the number of features. The dimensionality of our feature vector is linear in the number of frames S we use and hence this scaling is important. In addition, there exist power-efficient hardware implementations of regression trees suitable for mobile devices [36].



(a) Cumulative error distribution (b) Errors wrt the true depth

Figure 5: Validation of the regression tree approximation: a single tree of sufficient depth is expressive enough to represent the slow but accurate depth inference function. (a) the cumulative distribution function of absolute errors; (b) the same data broken down as a function of the true depth.

4.1. Evaluation Results

Because the tree model f is trained on the output of the MCMC inference it can at most match its depth accuracy. We now empirically study the approximation quality.

To this end we generate test data using the same offline procedure used during training of the trees. Using the test data we then compare the MCMC inferred depth against the fast regression trees. From the results in Figure 5 we can see that trees of depth 16 essentially match the predictive performance of the MCMC depth.

The efficiency of our regression trees is very high ($8 \cdot 10^6$ pixels/s). With this efficient runtime, we now consider designing a good time-of-flight measurement sequence.

5. Dynamic Sensing Design

The per-pixel response $\vec{R}^{(s)}$ measured in the s'th frame depends on two inputs. *First*, beyond our control, it depends on the imaging conditions at the corresponding surface patch, modelled by $\vec{\theta}^{(s)}$. *Second*, within our control, it depends on a *measurement design*, that determines the idealized response curve $\vec{C}$ and the ambient vector $\vec{A}$.

The measurement design of a single frame is described through a parametrization $Z \in \mathcal{Z}$. We provide further details on the parametrization below. To show in notation that each frame can have a different design we write $Z^{(s)}$ for the design of the s'th frame. We write the response curve as $\vec{C}^{(s)}$ and the ambient vector as $\vec{A}^{(s)}$ were the superscript denotes the implicit dependence on $Z^{(s)}$.

Because we control this design, we are free to actively select a different design $Z^{(s)}$ for each frame. We are going to argue that for a new frame we should select a design that is different to the design used in earlier frames. Instead of directly requiring designs to differ, we start from the principle that a measurement sequence should reveal *on average* the largest amount of information about a scene. To this end, we extend the *Bayesian experimental design* procedure originally proposed in [1] from the single frame case to the multi-frame setting.

In order to discuss the design problem we first describe the set of possible measurement sequences.

5.1. Design Space

We adopt the same the design space used in [1] and now give a short summary. In each capture period a fixed number L of laser pulses are emitted, typically several thousand. For each pulse we can choose how to integrate the reflected signal by selecting a specific time delay and a specific exposure time, both on the order of nanoseconds. Both the set of feasible time delays and feasible exposure times are finite sets and the product set determines all B possible ways a single emitted pulse can be measured. For each pulse the reflected light is accumulated into one of the (four, in our camera) coordinates of the response vector $\vec{R}$.

Therefore, a *design* for a single frame can be represented as an integer-valued matrix $Z \in \mathbb{N}^{4 \times B}$ with $\sum_j \sum_i Z_{ij} = L$. In addition there are hardware related constraints regarding the number of different delays and exposure times that can be used of the form $\sum_j 1_{Z_{ij}>0} \leq w$, for all i. Altogether we summarize these constraints as $Z \in \mathcal{Z}$.

The dynamic model uses the same design space applied to each frame, that is, we simply have individually $Z^{(s)} \in \mathcal{Z}$.

5.2. Dynamic Design Problem

Like the work of [1] we base the design objective on the principle of Bayesian experimental design and decision theory [9, 4]: a design is good if it leads to good expected depth accuracy. Formally we assume a loss function $\ell(\vec{\theta}^{(1:S)}, \hat{\theta}(\vec{R}^{(1:S)}))$ which quantifies prediction error given the ground truth $\vec{\theta}^{(1:S)}$ and a point estimate $\hat{\theta}(\vec{R}^{(1:S)})$. We use the sum of absolute depth errors of the last frame $\ell = |t^{(S)} - \hat{t}^{(S)}|$, ignoring estimates in the other imaging conditions. The design $Z^{(1)}, \dots, Z^{(S)}$ will be operated cyclically, as shown in Fig. 1b. Therefore, to measure performance we take the average across all cyclical rotations $J \in \mathcal{J}$ of which there are S. For example, with $S = 2$ this would be $(Z^{(1)}, Z^{(2)})$ and $(Z^{(2)}, Z^{(1)})$. We solve

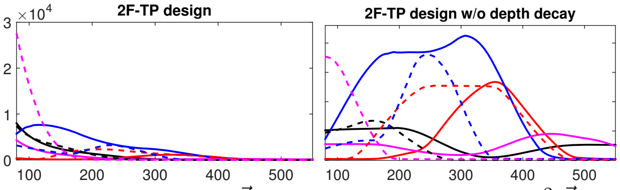
$$\min_{Z \in \mathcal{Z}} \frac{1}{S} \sum_{J \in \mathcal{J}} \mathbb{E}_{\vec{\theta}^{J(1:S)}} \mathbb{E}_{\vec{R}^{J(1:S)}} \left[\ell(\vec{\theta}^{J(1:S)}, \hat{\theta}(\vec{R}^{J(1:S)})) \right], \quad (22)$$

where the first expectation is over the prior distribution (Section 3.1), and the second expectation is over the forward model $P(\vec{R}^{J(1:S)} \mid \vec{\theta}^{J(1:S)})$ (Section 3).

Optimization of (22) is challenging but fortunately we found that the simulated annealing approach proposed in [1] for the static model extends readily to the dynamic case and we can find good local optima within several hours.

5.3. Optimization Results

We design a measurement sequence for two frames ($S = 2$) by minimizing (22) using 20,000 simulated annealing



(a) Response design $\vec{C}(t)$ (b) Response design $t^2\,\vec{C}(t)$

Figure 6: Designed response curve for the 2F-TP model. The solid lines correspond to the first four responses captured in the first frame; the dashed lines correspond to the second four responses captured in the most recent frame. (a) eight response curves with the $1/t^2$ depth decay, as the sensor will observe; (b) the same curves but multiplied with t^2 for visualizing the structure over the full depth range.

iterations on a basis set of $B = 3193$ delay/exposure pairs (103 possible delays, 31 exposures).

We show the resulting design in Figure 6. From Figure 6b it is clear that the response curves corresponding to the first frame (solid lines) are different to the curves of the second frame (dashed lines). We study the effect of this difference further in Section 6.2.

6. Experiments and Results

In the experiments we will validate the key contribution of using information from multiple time steps by showing improved depth accuracy. We demonstrate our proposed method for the case $S = 2$ because this is the simplest case where we use measurements from multiple time steps.

In all experiments we name the methods as 1F for the case $S = 1$ and 2F for the case $S = 2$. The SP model (2) is used in the 1F-SP and 2F-SP models, and the TP model (6) is used in the 1F-TP and 2F-TP models.

All known TOF depth methods use only IR frames from a single time step. To provide a fairer comparison we propose a simple baseline method that also makes use of previously captured frames. This baseline method works as follows: we perform depth inference using the 1F-SP or 1F-TP models for each frame individually, but then average the depth output. This is likely to work well if the camera is static or moves very slowly. We call this resulting method SP-avg or TP-avg, depending on whether we use the 1F-SP or 1F-TP method to do per-frame depth inference.

6.1. In-Model Validation

We validate the model as follows: we sample observations from the 2F-TP model, then use all possible four models and two baseline methods to infer depth. By construction the model assumptions are satisfied for the 2F-TP model and it should outperform the other methods. We also include the 1F-SP and 2F-SP models for comparison.

We show results in Table 1 and there are two observations to make: first, the TP model outperforms the SP model in all settings. Second, the dynamic 2F-TP model performs best for the medium and large error pixels: the static 1F-TP model has a 50 percent larger error for the 75% quantile and a 10 percent larger error for the median error. This demonstrates synthetically that the dynamic models (2F-SP and 2F-TP) can improve depth accuracy.

	Absolute error quantile (cm)		
Model	25%	50%	75%
Static 1F-SP	3.57	9.20	23.29
Baseline SP-avg	4.21	9.90	21.50
Dynamic 2F-SP (**ours**)	**3.28**	**7.78**	**16.82**
Static 1F-TP	**2.55**	6.79	21.70
Baseline TP-avg	2.87	7.26	18.69
Dynamic 2F-TP (**ours**)	2.56	**6.16**	**14.48**

Table 1: In-model validation: we sample observations from the 2F-TP prior and use four different models to explain the observations. Consistently, the use of observations from two frames improves the performance, in particular the worst quarter of errors (the 75% quantile) are reduced.

6.2. Dynamic Measurement Design Validation

In Section 5.2 we described a design procedure taking into account temporal dependencies. We now verify experimentally that by using a complementary design over time we obtain improved depth accuracies.

For this experiment, we take the 2F-TP design Z shown in Figure 6. We take the four response curves from the second frame and duplicate these to obtain an additional design Z' for the 2F-TP model that uses identical response curves for the first and second frames.

We simulate $\vec{\theta}_i^{(1:S)} \sim P(\vec{\theta}^{(1:S)})$, $i = 1, \ldots, 2048$ from the 2F-TP prior, then create two sets of response vectors by simulating the forward model, once for the design Z and once for Z'. We then perform posterior inference using the 2F-SP and 2F-TP models and compare depth accuracy against the known ground truth.

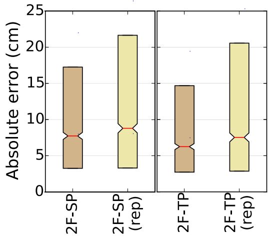


Figure 7: Dynamic design versus static design.

The results are shown as box plots in Figure 7. The 2F-SP and 2F-TP boxes use the complementary design Z, whereas the 2F-SP-rep and 2F-TP-rep boxes use the replicated design Z'. The complementary design ("2F-SP" and "2F-TP") has a better depth accuracy in terms of the median absolute depth error and also significantly better 75% quantile error compared to 2F-SP-rep and 2F-TP-rep, where we use the two frame model on the static design.

This confirms that the design objective selects a measurement design in such a way as to beneficially use and integrate information from both the current and past frames.

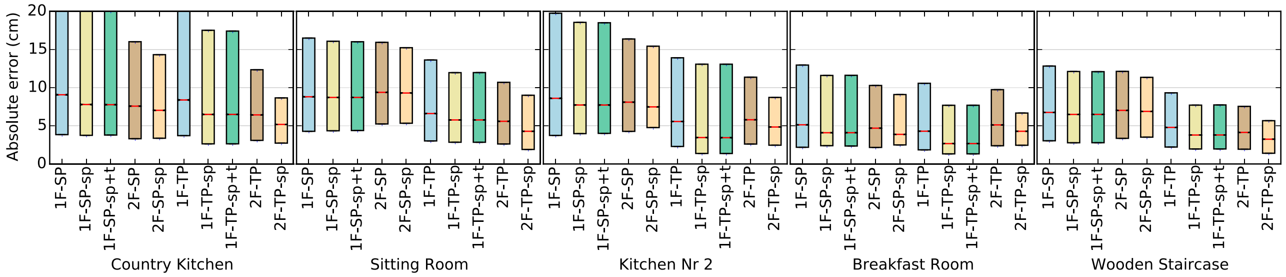


Figure 8: Box plots showing the absolute depth errors on five simulated scenes. Our main findings: 1. Our proposed dynamic models (2F-*) improve over the static models (1F-*) at the 25/50/75 quantiles. 2. We also improve over the baseline two frame methods (*-avg) and the spatio-temporal averaging (*-sp), (*-sp+t).

6.3. Rendering Simulations

We now study the performance of the proposed method using the light transport simulation approach of [1]. We use the modified physically-based renderer *Mitsuba* [20] from [1] to simulate realistic responses containing both direct and multipath components, while maintaining the ability to have ground truth depth. We use five scenes, in which we use approximate handheld camera motion trajectories. We then perform depth inference using our different models and compare the inferred depth against the ground truth depth using the median absolute error.

We compare against three baseline methods based on spatio-temporal filtering. In contrast to RGB images it is not possible to temporally filter RAW images because the measurement patterns are different over time. However, it is possible to filter depth output and we use the *guided filter* [19] using the depth image as guide image. We created three baselines: 1F-TP-sp, a spatial-filtered depth map from the 1F-TP method; 1F-TP-sp+t, a spatio-temporal-filtering of the two most recent depth outputs, using the guided filter on a $(\text{width}, \text{height}, 2)$ depth map tensor as in video denoising; 2F-TP-sp, a spatial-filtered depth map from the 2F-TP depth output. We optimize the guided filter parameters (radius and regularizer) to minimize the median abs error against the ground truth for each method.

Fig. 8 shows the results in the form of box plots (25/50/75 percentiles) for all five scenes and all methods. These results demonstrate the robustness of the 2F-TP model compared to all other methods: the highest errors (75 percentile) are consistently reduced across scenes, and in four out of five scenes the median error is the lowest. With respect to the single-frame methods we get a reduction of error because we use more data. While a simple baseline temporal averaging may sometimes work, we see that the behaviour of the these methods (SP-avg and TP-avg) is variable across scenes depending on the amount of motion/depth-discontinuities present. Our method is robust and performs well across all scenes. In addition we note that the two-path models (*-TP) fare better than the single-path (*-SP) models.

Regarding the spatio-temporal baselines, the results show wheras spatial filtering improves every method, temporal filtering on the depth is largely ineffective. Our method of temporal inference on the RAW frames combined with spatial filtering (2F-TP-sp) is the most effective.

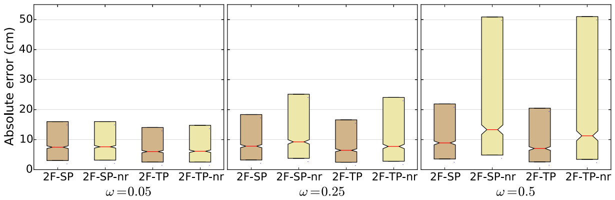


Figure 10: Box plots comparing depth accuracy of our robust motion model (2F-SP, 2F-TP) against a standard non-robust motion model (2F-SP-nr, 2F-TP-nr, corresponding to $\omega = 0$). For large motion ($\omega = 0.25$, $\omega = 0.5$) the robust models still work, whereas the non-robust standard motion model fails.

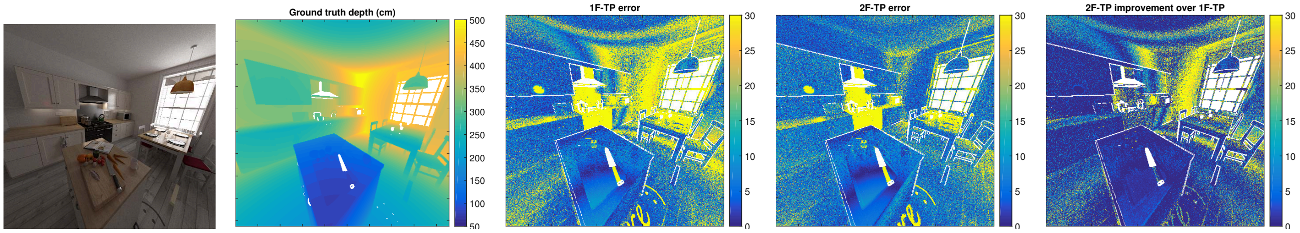
For visual inspection of the effects of the motion model we show a representative improvement of the 2F-TP model over the 1F-TP model in Figure 9.

6.4. Robustness of the Motion Model

Our motion model (10) is robust because it has the fallback option of reverting to an independent prior, ignoring past information. Whether this construction is successful depends on how robustly this switching is performed.

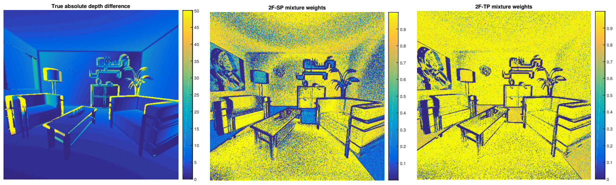
In a first experiment we show that a non-robust analogue of our model, corresponding to the choice $\omega = 0$ in (10) fails in the presence of depth discontinuities. To this end we sample from a 2F-TP prior with various choices of $\omega \in \{0.05, 0.25, 0.5\}$ corresponding roughly to small, medium, and strong motion. We then perform depth inference on the sampled responses using either our robust models, 2F-SP and 2F-TP, or using the non-robust models 2F-SP-nr and 2F-TP-nr where we set $\omega = 0$ so that only $Q(\vec{\theta}^{(s+1)}|\vec{\theta}^{(s)})$ is used. The results in Figure 10 confirm the usefulness of the mixture construction (10).

To further study the robustness of the motion model we perform the following experiment. Like in Section 6.3 we simulate a pair of frames and perform inference using either the 2F-SP or the 2F-TP model, using 16384 MCMC iterations for burn-in and inference, each. We visualize the belief of the model that a pixel can be explained using the



(a) Scene (b) True depth (c) 1F-TP Bayes error (d) 2F-TP Bayes error (e) Error reduction

Figure 9: Visual comparison of Bayes 1F-TP and Bayes 2F-TP errors: we can observe a significant error reduction in areas with strong multipath (corner, floor) throughout the whole scene. (Scene "Country Kitchen" by Jay-Artist, CC-BY)



(a) Change in depth (b) 2F-SP model M (c) 2F-TP model M

Figure 11: Robust change detection by the dynamic model.

local motion model Q in equation (10). To this end we compute a *motion map* statistic M for each pixel, where $0 \leq M \leq 1$, with a value close to one if the local motion model Q explains the observed response (more details in the supplementary). We expect most of the image to be explainable this way, with exceptions along depth edges.

We visualize the results for one scene in Figure 11. In Figure 11a we show the absolute ground truth difference in depth for two adjacent frames. In Figure 11b and 11c we show the inferred motion maps for the 2F-SP and the 2F-TP models. Indeed, the 2F-SP model and even more so the 2F-TP model are able to robustly separate the image into two classes: pixels where a slowly changing depth is observed (in yellow), and pixels where there is an abrupt change in depth (in blue). For pixels undergoing abrupt depth changes the model reverts to explaining the response through the independent prior beliefs. Hence (10) is a robust prior.

6.5. Live Demonstration

We run our 2F regression trees on sequences captured by a pulsed TOF camera similar to the one in [1], using designed measurement patterns. We show live results in a supplementary video and document.

7. Related Work

Motion compensation methods for phase-based time-of-flight [15] either require properties specific to phase-based TOF or optical flow. More generally, temporal denoising of depth images has been considered in [21, 29, 7, 24]. In these works explicit or implicit motion compensation is performed, using additional input from an accompanying RGB camera. In particular, structured light RGB-D data has been enhanced temporally as well as spatially in [21] using a joint bilateral filter together with a motion-compensated temporal filter, requiring 2 seconds per frame. Likewise the heuristic proposed in [29] enhanced RGB-D data spatio-temporally, achieving 1.4 frames per second and similarly the heuristic spatio-temporal filter in [24] takes more than ten seconds per frame. Our spatio-temporal filtering baseline is representative of these approaches. [37] proposes a Markov random field for spatio-temporal denoising. This model assumes a static camera and requires one seconds per frame. In contrast, our work does not use an additional RGB input and we do not assume a static camera but achieve realtime performance.

Recent work [32, 31, 30] performs 3D scene reconstruction from depth data over long temporal periods. These schemes are based on aligning the 3D point clouds of each depth frame, and integrating these in a common 3D scene representation. A key limitation of these methods is the reliance on depth as input: for low-reflectivity surfaces or in when strong ambient light is present the depth reconstruction may fail and these methods cannot be applied. Our approach is complementary because we integrate RAW IR responses temporally to improve depth output, which can serve as input to geometric scene reconstruction methods.

Our design of active sensing sequences resembles the goal of classic *active vision* systems [2] to sense optimal information relative to information already collected. Structured light systems have used acquisition patterns designed to minimize ambiguity and to maximize spatial resolution [35]. Furthermore, 3D scanning based on phase-shift based active illumination has leveraged models of motion to compensate for motion artifacts [41]. Closest to our approach is the recent *information gain* based adaptive structured light approach of [34]. Like in our approach a coherent probabilistic model is combined with decision theory in order to acquire information that maximizes future expected utility.

8. Conclusion

We proposed a procedure for temporal sensing and integration of time-of-flight RAW observations. We achieve realtime performance and robust improved depth accuracy. We believe that we made a first step to leverage the internal operation of time-of-flight cameras to more intelligently capture and integrate environment information.

References

[1] A. Adam, C. Dann, O. Yair, S. Mazor, and S. Nowozin. Bayesian time-of-flight for realtime shape, illumination and albedo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[2] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International Journal of Computer Vision*, 1(4):333–356, 1988.

[3] C. S. Bamji, P. O'Connor, T. A. Elkhatib, S. Mehta, B. Thompson, L. A. Prather, D. Snow, O. C. Akkaya, A. Daniel, A. D. Payne, T. Perry, M. Fenton, and V.-H. Chan. A 0.13 μm CMOS system-on-chip for a 512 $\times$ 424 time-of-flight image sensor with multi-frequency photo-demodulation up to 130 MHz and 2 GS/s ADC. *J. Solid-State Circuits*, 50(1):303–319, 2015.

[4] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, 1985.

[5] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.

[6] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

[7] M. Camplani and L. Salgado. Efficient spatio-temporal hole filling strategy for kinect depth maps. In *IS&T/SPIE Electronic Imaging*, pages 82900E–82900E. International Society for Optics and Photonics, 2012.

[8] O. Cappé, S. J. Godsill, and E. Moulines. An overview of existing methods and recent advances in sequential Monte Carlo. *Proceedings of the IEEE*, 95(5):899–924, 2007.

[9] M. H. DeGroot. *Optimal Statistical Decisions*. McGraw-Hill, New York, 1970.

[10] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009.

[11] J. Durbin and S. Koopman. *Time Series Analysis by State Space Methods: Second Edition*. Oxford Statistical Science Series. OUP Oxford, 2012.

[12] S. Felzenshtein, G. Yahav, and E. Larry. Fast gating photosurface. US Patent 8717469, 2014.

[13] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.

[14] D. Freedman, Y. Smolin, E. Krupka, I. Leichter, and M. Schmidt. SRA: fast removal of general multipath for ToF sensors. In D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, pages 234–249. Springer, 2014.

[15] J.-M. Gottfried, R. Nair, S. Meister, C. S. Garbe, and D. Kondermann. Time of flight motion compensation revisited. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 5861–5865. IEEE, 2014.

[16] M. Gupta, S. K. Nayar, M. B. Hullin, and J. Martin. Phasor imaging: A generalization of correlation-based time-of-flight imaging. Technical report, 2014.

[17] M. E. Hansard, S. Lee, O. Choi, and R. Horaud. *Time-of-Flight Cameras - Principles, Methods and Applications*. Springer Briefs in Computer Science. Springer, 2013.

[18] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[19] K. He, J. Sun, and X. Tang. Guided image filtering. In *European conference on computer vision*, pages 1–14. Springer, 2010.

[20] W. Jakob. Mitsuba renderer, 2010. `http://www.mitsuba-renderer.org`.

[21] S.-Y. Kim, J.-H. Cho, A. Koschan, and M. Abidi. Spatial and temporal enhancement of depth images captured by a time-of-flight depth sensor. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 2358–2361. IEEE, 2010.

[22] P. S. Laplace. Mémoir sur la probabilité des causes par les évènements. *Memoires de l'Academie des Sciences de Paris*, 6, 1774.

[23] P. S. Laplace. Memoir on the probability of the causes of events. *Statistical Science*, 1(3):pp. 364–378, 1986. English translation by S.M. Stigler.

[24] B.-S. Lin, M.-J. Su, P.-H. Cheng, P.-J. Tseng, and S.-J. Chen. Temporal and spatial denoising of depth maps. *Sensors*, 15(8):18506–18525, 2015.

[25] J. S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6(2):113–119, 1996.

[26] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[27] D. J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

[28] B. Marthi, H. Pasula, S. J. Russell, and Y. Peres. Decayed MCMC filtering. In *UAI*, 2002.

[29] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov. Temporal filtering for depth maps generated by kinect depth camera. In *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pages 1–4. IEEE, 2011.

[30] R. A. Newcombe, D. Fox, and S. M. Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–352, 2015.

[31] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011, Basel, Switzerland, October 26-29, 2011*, pages 127–136. IEEE, 2011.

[32] R. A. Newcombe, S. Lovegrove, and A. J. Davison. DTAM: dense tracking and mapping in real-time. In D. N. Metaxas, L. Quan, A. Sanfeliu, and L. J. V. Gool, editors, *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 2320–2327. IEEE, 2011.

[33] A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[34] G. Rosman, D. Rus, and J. W. Fisher. Information-driven adaptive structured-light scanners. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 874–883, 2016.

[35] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. *ACM Transactions on Graphics (TOG)*, 21(3):438–446, 2002.

[36] T. Sharp. Implementing decision trees and forests on a GPU. In *Computer Vision–ECCV 2008*, pages 595–608. Springer, 2008.

[37] J. Shen and S.-C. S. Cheung. Layer depth denoising and completion for structured-light RGB-D cameras. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1187–1194. IEEE, 2013.

[38] J. Stühmer, S. Nowozin, A. W. Fitzgibbon, R. Szeliski, T. Perry, S. Acharya, D. Cremers, and J. Shotton. Model-based tracking at 300Hz using raw time-of-flight observations. In *International Conference on Computer Vision (ICCV)*, 2015.

[39] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012.

[40] E. Tadmor, I. Bakish, S. Felzenshtein, E. Larry, G. Yahav, and D. Cohen. A fast global shutter image sensor based on the VOD mechanism. In *2014 IEEE Sensors*. IEEE, 2014.

[41] T. Weise, B. Leibe, and L. Van Gool. Fast 3D scanning with automatic motion compensation. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007.

[42] G. Yahav, S. Felzenshtein, and E. Larry. Capturing gated and ungated light in the same frame on the same photosurface. US Patent Application 20120154535, 2010.

[43] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011.