Tackling 3D ToF Artifacts Through Learning and the FLAT Dataset - Supplementary

Qi Guo[a,b], Iuri Frosio[a], Orazio Gallo[a], Todd Zickler[b], Jan Kautz[a]

[a]NVIDIA, [b]Harvard SEAS,

1 Working Principles of ToF Cameras

An ideal ToF camera illuminates the scene with an amplitude-modulated, sinusoidal signal $g(t) = g_1 \cos(\omega t) + g_0$ [1]. If the camera and light-source are co-located, the scene static, and the light is reflected only once, the signal reaching a camera pixel is:

$$s\left(t\right) = \frac{r}{c^2 \tau_0^2}\left[g_1 \cos(\omega t - 2\omega\tau_0) + g_0\right] + e_0, \tag{1}$$

where r is the reflectivity of the surface (including color and angle of view), τ_0 is time lapse of light traveling back to the pixel with one reflection at speed c, and e_0 is the environmental light. Coherently with [1], we can rewrite Equation 1 as:

$$\begin{array}{c} a = \frac{rg_1}{c^2\tau_0^2},\ a_0 = \frac{rg_0}{c^2\tau_0^2} + e_0 \\ s\left(t\right) = a\cos(\omega t - 2\omega\tau_0) + a_0 \end{array}. \tag{2}$$

The received signal $s(t)$ is still sinusoidal, but the phase of the received signal is changed, which allows estimating d from the $2\omega\tau_0$. Since ω is generally in the order of MHz and the sampling rate of a traditional sensor is much lower, *homodyne* ToF cameras modulate the incident signal with a high frequency, periodic function $b\cos\left(\omega t - \psi\right)$[1], whose phase ψ is programmable. After some trigonometry simplifications, the measured, modulated signal is:

$$\begin{aligned} \tilde{i}_{\psi,\omega}(t) &= s(t) b\cos(\omega t - \psi) \\ &= \frac{ab\cos(\psi - 2\omega\tau_0)}{2} + \frac{ab\cos(2\omega t + \psi + 2\omega\tau_0)}{2} + a_0 b\cos(\omega t - \psi) \end{aligned} \tag{3}$$

To collect a sufficiently large number of photons, the camera exposure time T is generally much larger than $\pi c/\omega$. Measuring $\tilde{i}_{\psi}(t)$ for an exposure time T is equivalent to integrating over a temporal range T; consequently, all the periodic terms in Equation 3 vanish and we have (apart from a scale factor):

$$i_{\psi,\omega} = \int_{-T/2}^{T/2} \tilde{i}_{\psi,\omega}(t)dt \approx ab\cos(\psi - 2\omega\tau_0) = af_{\psi,\omega}, \tag{4}$$

[1] In each pixel, electrons generated by incident photons are redirected into one of two buckets by a rapidly changing electric field.

where a is the *scene response*, encoding the irradiance reaching the pixel at time τ_0, after an impulse of light is sent out from the source, and we denote $f_{\psi,\omega} = b\cos(\psi - 2\omega\tau_0)$ as the *camera function*. We call $i_{\psi,\omega}$ the *raw correlation measurement* at phase ψ and frequency ω. Notice that the environmental light term is cancelled in the raw correlation measurement.

Using multiple raw measurements captured at a single frequency ω and $K \geq 2$ phases $\boldsymbol{\psi} = (\psi_1, \dots, \psi_K)$, $\boldsymbol{i}_{\boldsymbol{\psi},\omega} = (i_{\psi_1,\omega}, \dots, i_{\psi_K,\omega})$, we can compute the object distance d as:

$$d = \frac{c}{2\omega} \arctan\left(\frac{\sin\boldsymbol{\psi}\cdot\boldsymbol{i}_{\boldsymbol{\psi},\omega}}{\cos\boldsymbol{\psi}\cdot\boldsymbol{i}_{\boldsymbol{\psi},\omega}}\right). \tag{5}$$

However, if $d > \pi c/\omega$, Equation 5 wraps. Therefore, additional measurements at $L \geq 2$ frequencies $\boldsymbol{\omega} = (\omega_1, \dots, \omega_L)$ are needed. For each frequency, we take K raw correlation measurements at $\boldsymbol{\psi} = (\psi_1, \dots, \psi_K)$. For each frequency ω_l in $\boldsymbol{\omega}$, the object distance d and raw correlation measurements $i_{\boldsymbol{\psi},\omega_l}$ are related by:

$$d = \frac{c}{2\omega_l} \arctan\left(\frac{\sin\boldsymbol{\psi}\cdot\boldsymbol{i}_{\boldsymbol{\psi},\omega_l}}{\cos\boldsymbol{\psi}\cdot\boldsymbol{i}_{\boldsymbol{\psi},\omega_l}}\right) + \frac{\pi c k_l}{\omega_l}, k_l \in \mathbb{N}. \tag{6}$$

Combining Equation 6 for all frequencies, gives a linear mod system, that could be solved in closed form, using Goshov and Solodkin's algorithm [2]:

$$d = \sum_l A_l(\boldsymbol{\omega}) \arctan\left(\frac{\sin\boldsymbol{\psi}\cdot\boldsymbol{i}_{\boldsymbol{\psi},\omega_l}}{\cos\boldsymbol{\psi}\cdot\boldsymbol{i}_{\boldsymbol{\psi},\omega_l}}\right) + B(\boldsymbol{\omega}). \tag{7}$$

Notice $\{A_l(\boldsymbol{\omega})\}_{l=1,\dots,L}$ and $B(\boldsymbol{\omega})$ are constants once $\boldsymbol{\omega}$ is fixed. From Equation 7 one can easily obtain $\partial d/\partial \boldsymbol{i}_{\boldsymbol{\psi},\omega_l}$. Thus we directly adopt the differentiable Equation 7 in the pipeline for depth reconstruction, which enables our network to work only on the domain of raw correlation measurements. Derivatives with respect to d can be propagated through Equation 7 to train the network.

When there is multi-path interference (MPI), for the same illumination signal $g(t) = g_1\cos(\omega t) + g_0$, the signal reaching a camera pixel becomes:

$$s(t) = \int_{-\infty}^{t} \left(a(\tau)\cos(\omega t - 2\omega\tau) + a_0(\tau)\right) d\tau, \tag{8}$$

where the scene response $a(\tau)$ is now a function of time travel τ, and the raw correlation measurement is:

$$\begin{aligned} i_{\psi,\omega} &= \int_{-T/2}^{T/2} \left(\int_{-\infty}^{t} \left(a(\tau) b\cos(\omega t - 2\omega\tau) + a_0(\tau)\right) d\tau\right) \cos(\omega t - \psi)\, dt \\ &\approx \int_{-\infty}^{t} a(\tau) b\cos(\psi - 2\omega\tau) d\tau \\ &= \int_{-\infty}^{t} a(\tau) f_{\psi,\omega}(\tau) d\tau, \end{aligned} \tag{9}$$

where $a(\tau)$ is the *scene response function*, and $f_{\psi,\omega}(\tau)$ is the *camera function*. We denote $i_{\psi,\omega}(\tau) = a(\tau) f_{\psi,\omega}(\tau) d\tau$, notice $i_{\psi,\omega}(\tau_0)$ is the single reflection component in each raw measurement. If multiple measurements with different phases and frequencies are captured, $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}(\tau) = \{i_{\psi_1,\omega_1}(\tau), \dots, i_{\psi_K,\omega_L}(\tau)\}$ forms a manifold by varying τ and function a. We call this manifold, the "single-bounce-measurement-manifold".

If the raw measurements $\boldsymbol{i}_{\boldsymbol{\psi},\omega}$ are captured at a single frequency ω, and $K \geq 2$ phases $\boldsymbol{\psi} = (\psi_1, \dots, \psi_K)$, we have:

$$\begin{aligned}
\boldsymbol{i}_{\boldsymbol{\psi},\omega}(\tau) &= a(\tau) \begin{pmatrix} f_{\psi_1,\omega}(\tau) \\ \vdots \\ f_{\psi_K,\omega}(\tau) \end{pmatrix} d\tau = a(\tau) \begin{pmatrix} b\cos(\psi_1 - 2\omega\tau) \\ \vdots \\ b\cos(\psi_K - 2\omega\tau) \end{pmatrix} d\tau \\
&= a(\tau) b \cos(2\omega\tau) d\tau \begin{pmatrix} \cos(\psi_1) \\ \vdots \\ \cos(\psi_K) \end{pmatrix} + a(\tau) b \sin(2\omega\tau) d\tau \begin{pmatrix} \sin(\psi_1) \\ \vdots \\ \sin(\psi_K) \end{pmatrix} \\
&= \alpha_1 \begin{pmatrix} \cos(\psi_1) \\ \vdots \\ \cos(\psi_K) \end{pmatrix} + \alpha_2 \begin{pmatrix} \sin(\psi_1) \\ \vdots \\ \sin(\psi_K) \end{pmatrix},
\end{aligned} \tag{10}$$

where $\alpha_1 = a(\tau) b \cos(2\omega\tau) d\tau$ and $\alpha_2 = a(\tau) b \sin(2\omega\tau) d\tau$ are variables. This indicates the "single-bounce-measurement-manifold" is an hyper-plane spanned by vectors $\begin{pmatrix} \cos(\psi_1) \dots \cos(\psi_K) \end{pmatrix}^T$ and $\begin{pmatrix} \sin(\psi_1) \dots \sin(\psi_K) \end{pmatrix}^T$ in $\mathbb{R}^K$. Therefore, the raw correlation measurements with MPI $\boldsymbol{i}_{\boldsymbol{\psi},\omega}$, which is the sum of $\boldsymbol{i}_{\boldsymbol{\psi},\omega}(\tau)$, always lies on the hyper-plane, and one cannot discriminate a raw measurement with MPI from a raw measurements with only one single bounce.

However, if the camera uses $L \geq 2$ frequencies, and for each frequency $K \geq 2$ raw measurements are captured, the "single-bounce-measurement-manifold" becomes:

$$\begin{aligned}
\boldsymbol{i}_{\boldsymbol{\psi},\omega}(\tau) = a(\tau) \begin{pmatrix} f_{\psi_1,\omega}(\tau) \\ \vdots \\ f_{\psi_K,\omega}(\tau) \end{pmatrix} d\tau = a(\tau) \begin{pmatrix} b\cos(\psi_1 - 2\omega_1\tau) \\ \vdots \\ b\cos(\psi_K - 2\omega_L\tau) \end{pmatrix} d\tau \\
= a(\tau) b \ d\tau \begin{pmatrix} \cos(2\omega_1\tau)\cos(\psi_1) \\ \vdots \\ \cos(2\omega_L\tau)\cos(\psi_K) \end{pmatrix} + a(\tau) b \ d\tau \begin{pmatrix} \sin(2\omega_1\tau)\sin(\psi_1) \\ \vdots \\ \sin(2\omega_L\tau)\sin(\psi_K) \end{pmatrix}, \forall a, \tau
\end{aligned} \tag{11}$$

which is a non-linear subspace in $\mathbb{R}^{K\times L}$. The raw measurements with MPI, $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}} = \sum_{-\infty}^{t} \boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}(\tau)$ will not lie on the manifold anymore. Multiple-bounce artifacts can then be eliminated by learning a projection from the raw measurements with MPI to the single bounce manifold.

2 Characterization of the Kinect 2 Device

We describe here our characterization procedure of the Kinect 2 device. Our characterization takes into account several aspects:

- *Camera Response Function*: we verify the linearity of the Kinect 2 sensor.
- *Camera Functions*: The Kinect 2 emits signals with three frequencies; for each frequency it captures raw measurements of three phases, which produces nine raw correlation measurements. For convenience, we denote the nine measurements as $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}} = (i_{\psi_1,\omega_1}, \ldots, i_{\psi_9,\omega_9})$. We describe here the procedure to estimate each of the nine camera functions.
- *Per-pixel Delay*: Because of the hardware setup as well as camera / emitter alignment, each pixel of a ToF camera measures the phase of the returning signal with a different delay [3]. We show here how to estimate this delay.
- *Vignetting*: We show how to estimate the vignetting effect in the Kinect 2.
- *Noise Distribution*: We demonstrate our procedure to estimate the distribution of the noise .

2.1 Camera Response Function

We first verify whether the Kinect 2 can be treated as a linear camera. By placing different neutral density filters in front of the camera observing a fixed scene, we are able to draw the measured intensity of each raw measurement i_{ψ_j,ω_j} with respect to the transmittance of the filters. A typical camera response function is shown in Figure 1, which validates the linear assumption of Kinect 2.

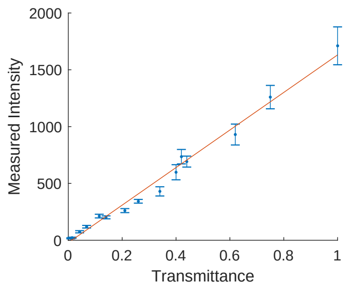

Fig. 1: A typical camera response function of Kinect 2. Error bars indicate the standard deviation of the noise.

2.2 Camera Function

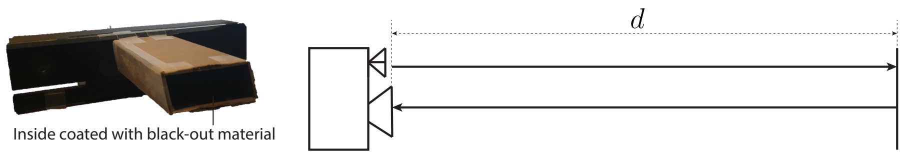

Fig. 2: Experimental setup of camera function calibration.

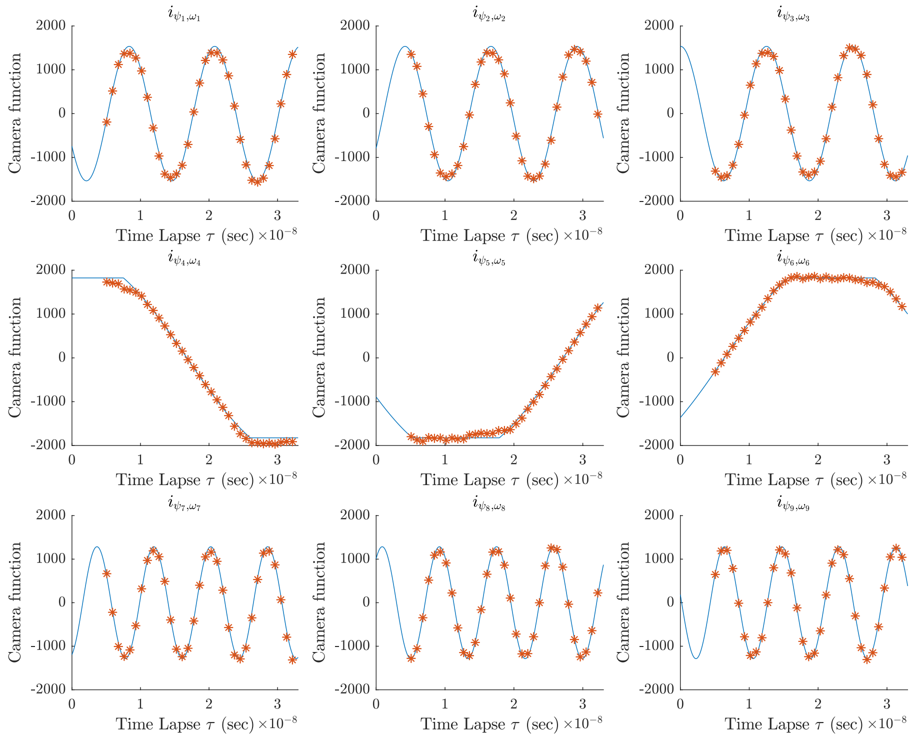

Fig. 3: Camera functions of Kinect 2. The red stars indicate experimental measurements, whereas the blue lines indicate the fitted camera functions.

We assume the camera function to be identical in each pixel. Figure 2 shows the setup used for camera function calibration. We place a light absorption tube in front of the camera light source, to reduce the emission angle of Kinect 2, and limit MPI, so that we can obtain an almost single bounce reflection from the calibration board. By placing the calibration board to a set of known depths $\{d_n\}_{n=1,\dots,N}$, we could obtain a set of measurements $\{\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}(d_n)\}_{n=1,\dots,N}$, in which $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}(d_n) = (d_n)^{-2}\boldsymbol{f}_{\boldsymbol{\psi},\boldsymbol{\omega}}(2d_n/c)$. By compensating for the inverse-square term, we obtain a series of observations $\{\boldsymbol{f}_{\boldsymbol{\psi},\boldsymbol{\omega}}(2d_n/c)\}_{n=1,\dots,N}$, representing the camera functions. Figure 3 shows the observed points and the fitting of nine camera functions of Kinect 2. Interestingly, camera functions in the second row with the lowest frequency is not sinusoidal, whereas the first and third are clearly sinusoidal. We fit the first and third frequency using $b\cos(\psi - 2\omega\tau)$, and the second using $\max(\min(b_1\cos(\psi - 2\omega\tau), b_2), -b_2)$.

2.3 Per-pixel Delay

We found that there is a different time delay $\nabla\tau(x, y)$ in the camera function for each pixel (x, y), which means a more accurate representation of the camera

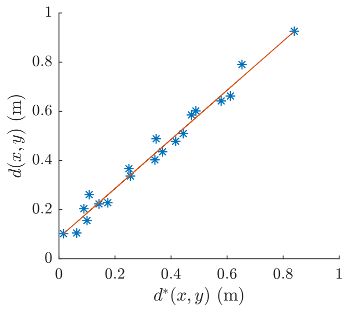

Fig. 4: Relationship between the measured distance $d(x,y)$, which is affected by a time delay $\nabla\tau(x,y)$, and the true distance $d^*(x,y)$.

function is $\boldsymbol{f}_{\boldsymbol{\psi},\boldsymbol{\omega}}(\tau+\nabla\tau(x,y))$. To estimate this delay, we acquired raw data with the Kinect 2 looking at a large, flat lambertian plane (a wall), so to have a planar ground truth shape and raw correlation measurements containing only a single reflection.

For each measurement, we put the Kinect 2 in a fixed location looking at the wall. We first place checkerboards on the wall, and use the Camera Calibration Toolbox[2] to estimate the extrinsic parameters and locate the plane in the camera reference system. Then, we took off the checkerboards and take raw measurements of the plane. We average 100 shots of the raw measurements to reduce the shot noise. Since the camera functions of the first and the third frequency are sinusoidal, we use both of them to estimate distance using Equation 7. Theoretically, the time delay $\nabla\tau(x,y)$, the true distance $d^*(x,y)$ and the measured distance $d(x,y)$ satisfy the following relationship:

$$\nabla\tau(x,y) = 2\left(d(x,y) - d^*(x,y)\right)/c. \tag{12}$$

Figure 4 shows a typical relationship between $d^*(x,y)$ and $d(x,y)$. By fitting the linear model $d(x,y) = d^*(x,y) + c\nabla\tau(x,y)/2$, we could obtain the time delay for any pixel (x,y).

2.4 Vignetting

After the complete calibration of the camera function $\boldsymbol{f}_{\boldsymbol{\psi},\boldsymbol{\omega}}(\tau+\nabla\tau(x,y))$, we estimate the vignetting effect of the Kinect 2. We assume the vignetting effect is the same across raw correlation measurements of different ψ,ω. Similar to the calibration of per-pixel delay, we take raw correlation measurements (averaging 100 shots to reduce noise) of a large, lambertian, plane $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}(x,y)$ to limit the MPI. The ground truth value of $\boldsymbol{i}_{\boldsymbol{\psi},\boldsymbol{\omega}}(x,y)$, indicated as $\boldsymbol{i}^*_{\psi,\omega}(x,y)$, can then be computed as in Equation 4, which takes into account the sinusoidal nature of the emitted signal as well as the varying depth for pixels that are far from the

[2] `http://www.vision.caltech.edu/bouguetj/calib_doc/`

camera optical axis. Once $\boldsymbol{i}^*_{\boldsymbol{\psi},\boldsymbol{\omega}}(x,y)$ is computed, we estimate the vignetting ratio $r(x,y)$ as:

$$r(x,y) = \frac{\left(\sum_j (i_{\psi_j,\omega_j}(x,y))^2\right)^{1/2}}{\left(\sum_j (i^*_{\psi_j,\omega_j}(x,y))^2\right)^{1/2}}. \tag{13}$$

To reduce the effects of noise, we repeat the process on 4 different scenes, taking 100 measurements per scene and averaging $r(x,y)$ over all the scenes and measurements.

2.5 Noise Distribution

We use an non-parametric model to record the noise distribution. We assume each pixel to be independent from the others and having an identical noise distribution. For a pixel (x,y), we take 100 raw measurements of a fixed scene $\{\boldsymbol{i}^j_{\boldsymbol{\psi},\boldsymbol{\omega}} = (i^j_{\psi_1,\omega_1},\ldots,i^j_{\psi_9,\omega_9})\}_{j=1,\ldots,100}$, and computed the average for each channel $i^*_{\psi_k,\omega_k} = \sum_j i^j_{\psi_k,\omega_k}$. We estimate then the empirical distribution, indicated by $p(i_{\psi_k,\omega_k}|i^*_{\psi_k,\omega_k})$, by accumulating samples from different pixels with the same discretized average value. We repeat the process for 15 different scenes to make sure that the samples are sufficient for each discretized average value within the perception range of the camera. We record a look-up table of $p(i_{\psi_k,\omega_k}|i^*_{\psi_k,\omega_k})$ for every $i^*_{\psi_k,\omega_k}$. Experimental data acquired in this way show that the noise distribution for different ψ_k and ω_k are indeed identical, as initially assumed. Noise can therefore be represented by a single look-up table for any phase and frequency. A visualization of a part of the look-up table is in Figure 5.

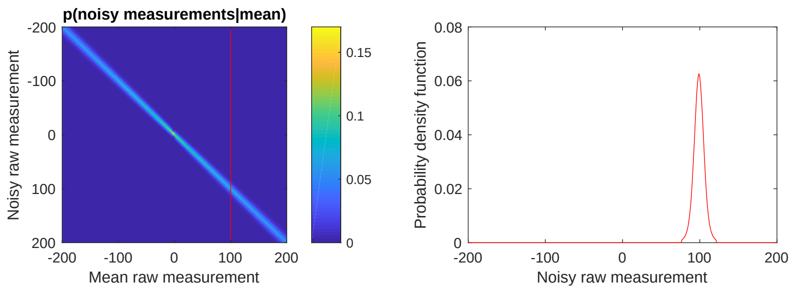

Fig. 5: Left: Distribution of noisy measurements as a function of the sample average. This distribution has been generated from different raw correlation measurements of 15 scenes; the sample average has been obtained from 100 measurements of the same scene. Right: a slice from the noise distribution (red line in the left panel).

3 The FLAT dataset

Figure 6 shows several sample 3D models used to generate the FLAT dataset, and Figure 7 demonstrates nine raw correlation measurements of a sample scene. Table 1 provides the detailed statistics of training and testing conducted in the main paper using FLAT dataset. The FLAT dataset contains 2087 scenes in total; 1141 scenes are used for training the networks in the main paper, whereas 117 are dedicated to testing. MPI is accurately simulated in each of these scenes thanks to the use of transient rendering simulation [4]. Motion in the training and testing scene can be generated with the approximated motion model provided within the FLAT dataset; in this case, MPI is approximated too. 26 of the testing scenes contains full motion data, i.e. for these scenes we simulate and render the moving objects in different positions and orientations, thus MPI and motion are correctly simulated for this subset of testing data.

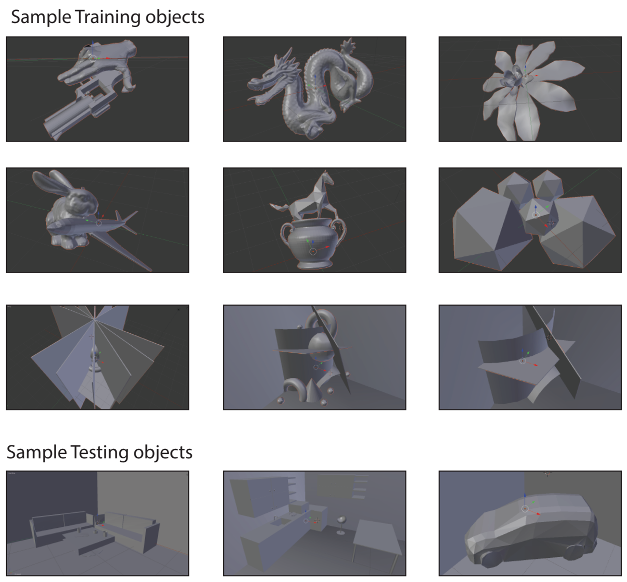

Fig. 6: Sample 3D objects used in FLAT datset for training and testing

	Training	Testing Static, MPI	Testing Motion, MPI
Number of scenes	1141	91	26
Motion Simulation	Approximated	Approximated	Correct
MPI Simulation	Correct in static	Correct in static	Correct

Table 1: Details of training and testing sets in the FLAT dataset.

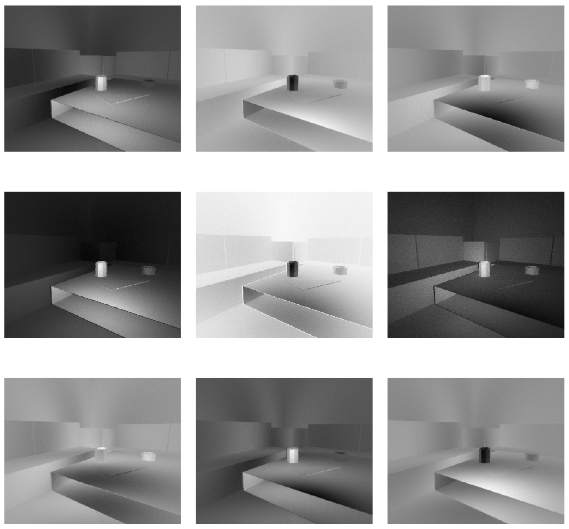

Fig. 7: Sample raw measurements of a scene in the FLAT dataset.

4 More Analysis

4.1 Shot noise versus Monte Carlo rendering noise

We compared the calibrated readout/Poisson noise of the Kinect 2 camera with the noise from the Monte Carlo (MC) transient rendering. To evaluate the MC noise, we render the transient propagation of a scene setting multiple times, use each to create the raw correlation measurements, and compute the variance of each pixel in the raw measurements. We assume the MC rendering is un-biased, so the MC noise is reflected by the variance of pixel values. In this test, the average variance of each pixel caused by the Monte Carlo rendering is 0.7LSB, while the calibrated shot noise of the Kinect 2 is greater than 20LSB. Thus the rendering noise of the FLAT dataset is negligible.

4.2 Reduction of shot noise by our method

As for the capability of MRM to reduce the effect of readout/Poisson noise, we simulated raw measurements of a scene with a cube (2x2x2m) only. The convex shape of the cube guarantees that there is no MPI. We render raw measurements of the cube with different albedos (and thus different signal-to-noise ratios), thus pixel values of the raw measurements cover almost the entire dynamic range of a real Kinect 2. The mean depth error on this dataset is 3.8cm if we use MRM-LF2*, and 4.4cm for the LF2* module only (no shot noise reduction algorithm).

References

1. Heide, F., Heidrich, W., Hullin, M., Wetzstein, G.: Doppler time-of-flight imaging. In: ACM Transactions on Graphics (SIGGRAPH). (2015)
2. Gushov, V., Solodkin, Y.N.: Automatic processing of fringe patterns in integer interferometers. Optics and Lasers in Engineering (1991)
3. Xiang, L., Echtler, F., Kerl, C., Wiedemeyer, T., Lars, hanyazou, Gordon, R., Facioni, F., laborer2008, Wareham, R., Goldhoorn, M., alberth, gaborpapp, Fuchs, S., jmtatsch, Blake, J., Federico, Jungkurth, H., Mingze, Y., vinouz, Coleman, D., Burns, B., Rawat, R., Mokhov, S., Reynolds, P., Viau, P., Fraissinet-Tachet, M., Ludique, Billingham, J., Alistair: libfreenect2: Release 0.2 (April 2016)
4. Jarabo, A., Marco, J., Muñoz, A., Buisan, R., Jarosz, W., Gutierrez, D.: A framework for transient rendering. In: ACM Transactions on Graphics (SIGGRAPH ASIA). (2014)