CS838 Project Proposal: Optimal Coding Functions for Continuous-wave Time of Flight Imaging

Zhicheng Gu, Nikhil Nakhate, Zhenyu Zhang, Felipe Gutierrez Barragan

March 27, 2017

Background: Time of flight (ToF) refers to the time that a light pulse takes to travel from a source to a target scene and back to a detector. This technique is often used to recover scene depths and shape. Continuous-wave ToF (CW-ToF) systems are one type of ToF cameras that, due to their low-cost and compact size, have the potential to impact a wide set of applications. In these systems the light source and sensor exposure are temporally modulated. The radiant intensity (brightness of light source) emitted by the light source is determined by a periodic modulation function $M(t)$. The sensor exposure is modulated by a periodic demodulation function, $D(t)$. The light reflected from the scene and incident on the sensor will be a scaled and phase shifted version of the modulation function denoted as the incident radiance, $L(t)$. The brightness, $B(t)$, measured by the sensor is the temporal correlation between $L(t)$ and $D(t)$. Note that the scene depth, Γ, is encoded in the temporal shift (phase shift, $\phi = \frac{2\Gamma}{c}$) of the received radiance and consequently only a fixed range of depths can be recovered from a given modulation frequency (ω) due to the periodicity of the function. This process is illustrated by Figure 1.

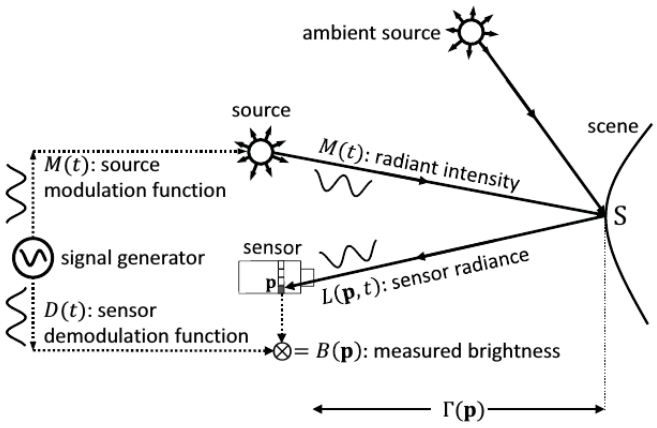

Figure 1: Illustration of a CW-ToF setup.

The brightness measured by the camera sensor can be expressed as shown in equation 1, where β is the scene albedo (light absorbed by scene), $f(\Gamma)$ is the correlation function (equation 2), and L_a is the ambient illumination (assumed to be constant). Therefore to determine the unknowns (β, Γ, L_a) we need at least 3 brightness measurements. γ_i is the result of integrating $D(t)$ over the integration period τ.

$$B_i(\beta, \Gamma, L_a) = \beta F_i(\Gamma) + \gamma_i L_a, \quad 1 \leq i \leq K, \quad K \geq 3 \tag{1}$$

$$F_i(\Gamma) = \int_0^\tau D_i(t) M_i(t - \frac{2\Gamma}{c}) dt \tag{2}$$

Problem Statement: Current CW-ToF systems use sinusoidal or square waves as the coding functions (i.e modulation/demodulation functions), which are highly sub-optimal. Recent work on the signal processing theory of ToF has shown that the choice of coding function has a significant impact on measurement errors. This is due to the fact that certain functions are less robust to noise sources (photon noise, shot noise, etc) than others. *The goal of this project is to explore neural network designs from which we could extract an optimal coding function or generate it.*

Data: We have developed a physics-based simulator for CW-ToF. We will use this to generate a synthetic data set for training, testing, and evaluating the generated coding functions. The simulator takes as input: a modulation function, a demodulation function, scene properties (true depth, ambient illumination, albedo).Given these inputs, the simulator simulates various steps ofthe C-ToF imaging process, including light transport (light emission, propagation, reflection and shading) and sensor physics (de-modulation, gain, saturation, ADC noise, quantization). It uses a physically accurate affine noise model, including both photon noise and sensor read noise. A forward pass through the simulator would follow the steps outlined in figure 2.

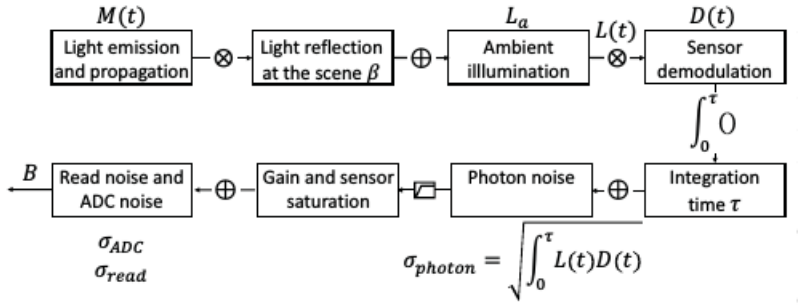

Figure 2: Chain of steps for simulation of CW-ToF imaging.

Preliminaries: We present the following preliminaries that will be used when outlining each approach.

1. *The correlation function:* In CW-ToF cameras the only two components that we have control over are $M(t)$ and $D(t)$. These two functions are summarized by the $F(\Gamma)$. Therefore, an optimal $F(\Gamma)$ implies an optimal $M(t)$ and $D(t)$.

2. *Delta δ coding function:* Note that if we let $M(t)$ be a delta function $\delta(t)$, then the correlation function, $F(\Gamma)$, will be equal to $D(t-2\Gamma/c)$. Therefore we could simply fix $M(t)$ to a delta function and just try to find an optimal $D(t)$.

3. *Coding function discretization:* $M(t)$ and $D(t)$ can be discretized as a set of N linearly interpolated points. The discrete counterparts of these functions would be the $N \times 1$ vectors $\mathbf{m}$ and $\mathbf{d}$. Let $\mathbf{C}_m$ be the $N \times N$ circulant matrix constructed from $\mathbf{m}$ (See end of document for the description of a circulant matrix). Then the discrete version of equation 2 would be:

$$\mathbf{f}_i = \mathbf{C}_{\mathbf{m}_i}\mathbf{d}_i\Delta t, \quad 1 \leq i \leq K, \quad K \geq 3 \tag{3}$$

Regression Approach: In this approach, the goal is to have the artificial neural network learn the depth recovery process. Given K input radiance vectors ($\mathbf{l}_i$) the network should learn how to output the correct depth. To this end we will employ the following network design:

1. *First layer (Demodulation weights):* We want to model the vector matrix product of equation 3 with the k input radiance vectors $\mathbf{l}_i$ and the weights. The learned weights on this layer would correspond to the demodulation function.

2. *First Activation (Noise modeling):* The correlation performed by the first layer does not take into account the noise sources a typical ToF is subject to (see figure 2. The noise sources are proportional to the magnitude of the correlation measurements so we can model them in this layer once we have a correlation measurement (i.e the resulting matrix vector product).

3. *Secondary layers (Regressing depth):* The following layers in the network will learn how to use the K brightness measurements to give an estimated depth measurement.

4. *Output layer (depth):* This layer should output a real positive number: the depth estimate Γ_{est}.

5. *Loss function:* The incoming radiance vectors l_i will have a true depth, Γ_{true} associated to them. We can use a square difference loss between Γ_{true} and Γ_{est} to propagate through the network.

If successful, we should be able to extract an optimal demodulation function from the first layer weights, and implement the secondary layers on the hardware to compute depth.

Generative Approach: This approach will involve an initial literature review on Generative Neural Networks. The literature review combined with our problem specific knowledge should give us more insight on the network design. The idea is to train a network that output a optimal coding functions given scene parameters (ambient illumination, scene albedo, depth ranges).

Evaluation: We will develop a depth error metric for a full scene (a depth map for a scene such as a wall, an object, etc) and compare each resulting coding function from each approach to commonly used coding function such as sinusoidal and square.

- For the regression approach, if the network is able to learn to recover accurate depth estimates we will be able to compare the network depth estimates vs. the depth estimates of a sinusoidal coding function setup running on the ToF simulator.

- For the generative approach, we might end up with ready to use coding functions. So we can input those coding functions to the ToF simulator and see how they perform.

Software Tools: Since the simulation code is in Python we will use python-based tools/frameworks. We will use Keras for prototyping of the neural network designs. If we find that Keras lacks certain functionality/flexibility for our specific neural network design, we will use Tensorflow which removes a layer of abstraction (e.g Keras) when constructing the neural network and hence should give us enough flexibility to implement any custom design. For the cloud platform, any could with linux system should fit our requirement.

Hardware: If needed, we have access to multiple GPUs.

Appendix

Circulant matrix: Let $\mathbf{m}$ be the following $1 \times N$ vector

$$\mathbf{m} = \begin{bmatrix} m_1 & m_2 & m_3 & \dots & m_N \end{bmatrix} \tag{4}$$

Then the circulant matrix, $\mathbf{C_m}$, generated by that vector is defined as the following $N \times N$ matrix

$$\mathbf{C_m} = \begin{bmatrix} m_1 & m_2 & m_3 & \dots & m_N \\ m_N & m_1 & m_2 & \dots & m_{N-1} \\ m_{N-1} & m_N & m_1 & \dots & m_{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_2 & m_3 & m_4 & \dots & m_1 \end{bmatrix} \tag{5}$$

References