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

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

March 29, 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 CW-ToF the light source and sensor exposure are temporally modulated. The radiant intensity (i.e intensity of light) emitted by the 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 a sensor pixel (p) will be a scaled, phase shifted, and vertically shifted version of the $M(t)$ denoted as the incident radiance, $L(\mathbf{p}, t)$ (see equation 1. The brightness, $B(\mathbf{p})$, measured at a sensor pixel is the temporal correlation between $L(\mathbf{p}, t)$ and $D(t)$. Note that the scene depth, Γ, is encoded in the temporal shift (phase shift, $\phi = \frac{2\Gamma}{c}$) of in $L(\mathbf{p}, t)$. This process is illustrated by Figure 1.

$$L(\mathbf{p}, t) = \beta(\mathbf{p}) M(t - \frac{2\Gamma}{c}) + L_a \tag{1}$$

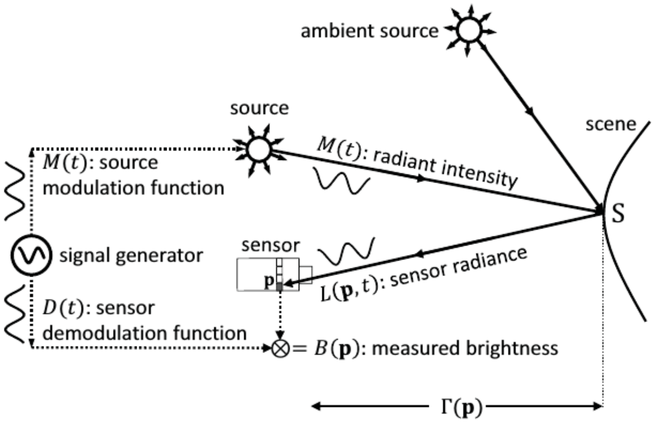
ambient source
source
scene
$M(t)$: source modulation function
$M(t)$: radiant intensity
S
signal generator
sensor
p
$L(\mathbf{p}, t)$: sensor radiance
$D(t)$: sensor demodulation function
$= B(\mathbf{p})$: measured brightness
$\Gamma(\mathbf{p})$

Figure 1: Illustration of a CW-ToF setup.

The brightness measured by the sensor can be expressed as shown in equation 2, where $\beta(\mathbf{p})$ is the scene albedo (light absorbed), L_a is the ambient illumination (assumed constant), and $\gamma_i = \int_0^\tau D(t)dt$. To determine the unknowns (β, Γ, L_a) we need at least 3 brightness measurements.

$$B_i(\mathbf{p}, \beta, \Gamma, L_a) = \int_0^\tau D_i(t) L_i(\mathbf{p}, t) dt = \beta(\mathbf{p}) \int_0^\tau D_i(t)(M_i(t - \frac{2\Gamma}{c}))dt + L_a \gamma_i \tag{2}$$

$$1 \leq i \leq K, \quad K \geq 3 \tag{3}$$

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 of the CW-ToF imaging process, including light transport (light emission, propagation, reflection and shading) and sensor physics (demodulation, gain, saturation, ADC noise, quantization). It uses a physically accurate noise model, including both photon noise and sensor read noise. A forward pass through the simulator would follow the steps outlined in figure 2.

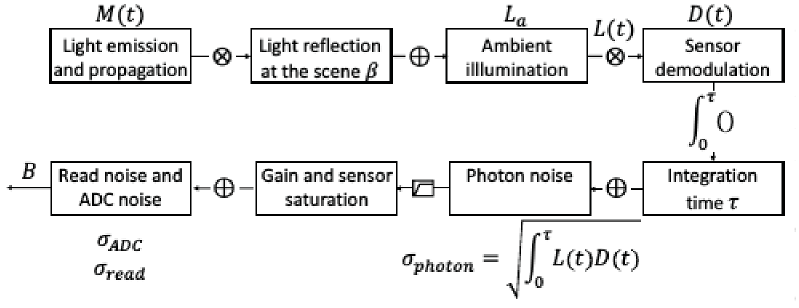

Figure 2: Chain of steps for simulation of CW-ToF imaging.

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

1. *Delta δ coding function:* Note that if we let $M(t)$ be a delta function, $\delta(t)$, then the integral in equation 2 will be equal to $D(-\frac{2\Gamma}{c})$.

2. *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 $N \times 1$ vectors ($\mathbf{m}$ and $\mathbf{d}$). Then the discrete version of equation 4 would be:

$$b_i = (\beta \Delta t)\mathbf{d}_i^t \mathbf{m}_i + L_a \sum_{j=1}^{N} d_{i,j} \tag{4}$$

Regression Approach (Single Pixel): 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-vector product of equation 4 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 brightness measurements (b_i) so we can model the noise in this layer once we compute the brightness measurement (multiplication of input vectors and first layer weights).

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.

Regression Approach (Full Depth Image): Note that in the previously described approach we are recovering the depth for a single point/pixel of the scene. In this approach we will scale the network described in the first approach such that we will have input measurements for all points in a scene and we will predict a depth map of a full image.

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 functions such as sinusoidal and square.

- For the regression approaches, 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.