PHYSICS AWARE VIDEO DENOISING

by

Trevor Seets

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Electrical and Computer Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2024

Date of final oral examination: 5/30/2024

The dissertation is approved by the following members of the Final Oral Committee:
Andreas Velten, Associate Professor, Biostatistics and Medical Informatics
Mohit Gupta, Associate Professor, Computer Science
William A. Sethares, Professor, Electrical and Computer Engineering
Christie Lin, Assistant Adjunct Professor, Medical Physics

*"I want to know why. Why **everything**. I don't know the answers, but a few days ago I didn't know there were questions."*

-Sir Terry Pratchett

ACKNOWLEDGMENTS

This dissertation would not have been possible without the support of my advisor, mentors, collaborators, friends and family. First, I would like to thank my advisor, Prof. Andreas Velten, for his guidance and support, without which this dissertation would not be possible. I always looked forward to our interesting conversations, and the excitement from Andreas when exploring a new idea. I appreciate the freedom and support Andreas provides, and I could not have asked for a better advisor.

I would also like to thank my lab-mates at the Computational Optics Group: Prof. Atul Ingle, Talha Sultan, Alex Bocchieri, Wei Lin, Dr. Felipe Gutierrez-Barragan, Yizhou Lu, Dr. Khadijeh Masumnia-Bisheh, Ehsan Ahmadi, Dr. Ji Hyun Nam, Dr. Xiaochun Liu, and Dr. Sebastian Bauer. I am grateful for the interesting conversation, and friendship from all of you. Prof. Atul Ingle was my mentor for my first year in the group; Atul helped me quickly learn the ropes of research and I am very grateful for his mentorship and kindness. Thanks to Talha for our "two person jobs" in the lab, and Alex for his "middle management" of the same jobs; those were some of the more fun days in the lab.

Next, I would like to thank Dr. Christie Lin, and Dr. Adam Uselmann who were my mentors at OnLume during my year long internship. I learned a lot about how research can be translated to industry from both of you. I thank Christie for her kindness, support, and her service on my committee, as well as many enlightening conversations; I learned a lot about leadership and innovation from Christie.

Finally, I would like to thank my family for their unending support and encouragement. I would like to thank my parents Chris and Michelle, and my sister, Skyler, for always encouraging me. I would like to thank my partner, Fiona, for keeping me sane with the time we spend together. This dissertation is dedicated to my family.

The work in my dissertation is based upon work supported by the Department of Energy [*] / National Nuclear Security Administration under Award Number

[*] **U.S. Department Funding Disclaimer:**This report was prepared as an account of work spon-

DE-NA0003921, the National Science Foundation (GRFP DGE-1747503, CAREER 1846884), DARPA (REVEAL HR0011-16-C-0025), the Wisconsin Alumni Research Foundation, NIH/NCATS Grant No. 1UL1TR002373 to UW ICTR, and McPherson Eye Research Institute "Expanding our Vision" grant.

sored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In this dissertation, we develop video denoising algorithms in a number of application spaces. We begin by considering video denoising for single photon cameras where we impose a strong temporal prior that lends itself to an event representation based on statistical changepoint detection along time. This representation consists of changes in brightness and the brightness between changes, we use this method to produce high speed video. The bulk of this dissertation then considers the application space of fluorescence guided surgery (FGS) where the goal is to denoise noisy fluorescence video to assist in real time surgical guidance. We consider two noise models in FGS, one based on ideal hardware where the difficulty lies in detection of very low signals requiring large temporal integration times. We develop an algorithm, OFDVDnet, based on motion compensation to deal with this low signal regime that compresses the data temporally before using a neural network for denoising. OFDVDnet exploits the presence of a second clean RGB reference camera in FGS to improve denoising performance. In the second noise model, we develop a realistic model for noise seen on a commercial FGS system. FGS contains a difficult spatially varying bias term, coming from imperfect hardware, that breaks most zero-mean assumptions used in video denoising methods. We call this bias term laser leakage light(LLL), and it is one of the core difficulties in improving FGS system sensitivity and is normally considered a hardware problem. We treat LLL as a noise term that can be removed and develop a new set of baseline denoising models to deal with this difficult bias term. We then use our baseline models to impose a cost function of FGS hardware design which before was ill-posed in a complex trade-off space. Importantly, throughout this dissertation we base our problems based on physical understanding of the measurement process and use realistic noise models based on this understanding.

1 INTRODUCTION

Any physical measurement will inevitably be corrupted with *noise* which is normally considered as deviations from some true state of the universe due to imperfections in the measurement device. This definition of noise gives the impression that with enough engineering we can create devices that make a perfect noiseless measurement. In imaging, our cameras are rapidly improving and approaching, in some sense, the perfect device for measuring light intensity with sensitivity at the single photon level. However, noise will still be present in images taken by a perfect camera; the very act of measuring light induces noise due to the quantum nature of light. Imagine a perfect camera measuring every photon and a light source that emits one photon every two seconds, if our camera only measures for a second, sometimes it will see a photon, sometimes it won't and other times it may even see two photons. But on average this perfect camera will see half a photon in a second which is never a possible measurement, but this average is the most useful definition of the system; the average is the signal we are trying to measure. So noise should not be understood as deviations from a true state of the universe, because often there exists an abstraction of the system that is more useful than knowing the entirety. Noise should rather be considered as a natural unavoidable aspect of the measurement that gives imperfect information about a signal. This definition leaves it up to the task for what we consider signal and what we consider noise. Most of this dissertation explores removing noise from video by using physical models to understand signal and noise.

The simplest definition of signal for a scene point in an image is the mean photon rate or photon flux coming from that scene point. In video, this gets more difficult due to motion, because an object may move to different points in the scene throughout time, leading to a definition of signal that is time dependent. In video, we can no longer just make a long exposure measurement to find a good estimate of the mean photon flux, instead we need to rely on physical models of both the scene and the measurement process to reconstruct a video. The first 3 chapters of this dissertation explore different assumptions for both these aspects, starting from simple models and priors before moving toward a data driven approach for more complex video.

The first useful prior is that for zero-mean noise, and in a video without motion, a temporal average will converge to the signal with enough time. If we can deal with motion in the scene, a simple average will be the optimal temporal denoising strategy. The problem then becomes dealing with motion, where the strategy will be dependent on the sort of motion you expect in the scene. We begin by looking at simple rigid body motion cases in Chapter 2, but use a single photon camera which produces high frame rate video with a lot of noise. This represents the limit high speed video where each frame consists of at most one photon which leads to incredibly noisy individual frames. This makes spatial denoising of this data incredibly hard; however, by using a strong temporal prior we are able to successfully reconstruct high speed video from this single photon data. The core idea is based on an event representation where we statistically detect motion in the scene to define bounds of a temporal average, to produce a number of events that define a piece-wise constant temporal representation of the video. This reduces noise and greatly compresses the data while also preserving the high frame rate of the data.

We then move onto a specific domain problem in fluorescence guided surgery (FGS) for the next three chapters. FGS is a surgical technique that is used to help surgeons visualize and delineate tissues of interest using a fluorescent camera. We choose to pursue FGS because the fluorescence process is inherently photon inefficient and produces very dim signals if measured at video frame rates. Luckily FGS systems also provide a clean reference video that is a standard RGB video of the scene which can be integrated into denoising methods to improve results. With the additional constraint that FGS is most useful as a real-time device, FGS becomes a incredibly interesting use case for video denoising where we tackle how to use the reference video in the denoising process. Chapter 3 focuses on how to use a motion corrected temporal average as a temporal sufficient statistic to allow for memory efficient use of denoising neural networks.

Chapter 4, continues to study FGS video denoising but focuses on creating a realistic noise model and introduces a causal constraint. Interestingly, FGS also contains a spatially varying background term that can obscure the fluorescent signal, and current FGS systems have read noise that is both temporally and spatially correlated. This full noise model breaks the zero-mean noise assumption used in earlier chapters and much of the video denoising literature, leading to many state of the art methods failing. We study what is required in a deep learning method to succeed in denoising this problem.

Chapter 5 seeks to close the loop between computation and hardware by using the denoising models developed in Chapter 4 as a hardware design cost function. The hardware design of FGS systems exists in an ill-posed trade-off space. By using computational methods to create a cost landscape over this trade-off space, a cost function can be found to optimize hardware choices to give optimal performance for a system equipped with denoising algorithms.

Chapter 6 is one of my other projects away from video denoising. This chapter is based on retro-reflections where collimated light is reflected back at the light source from a focused imaging system. The resulting retro-reflection can be imaged and used to determine things about a distant imaging system, such as rotation, focusing distance, or classification of the imaging system.

1.1 Intro and Motivation for Fluorescence Guided Surgery Denoising

Fluorescence Guided Surgery

The current standard of care (SOC) in many surgeries rely heavily on non-quantitative measures such as surgeon perception of tissue under normal lighting or tactile tissue cues. Fluorescence guided surgery (FGS) is a promising technique to improve the SOC by giving surgeons a quantifiable fluorescence video feed that helps identify the state of different tissues leading to an improvement in surgical decision making and an improvement in patient outcomes (Sutton et al. (2023)). FGS relies on a fluorescent contrast agent, either a drug or a naturally occurring fluorophore, that when imaged by an FGS imaging system helps delineate or classify tissues of interest. Fig 1.1(a) shows an overview of the working of an FGS system. First, a contrast agent is injected into a patient which binds to a target tissue. Next, excitation light illuminates the patient, the contrast agent will absorb this excitation light and emit light of a higher wavelength. This emitted light is fluorescence and is what we wish to capture, this light first passes through an emission filter designed to remove the excitation light before being imaged by the fluorescence camera. Additionally, a reference camera captures a co-focused RGB video of the scene that can help the surgeon orient the fluorescence video.

Fig 1.1(b) shows an overview of contrast agents that are currently used in clinical and research environments, in terms of their signal level and measurement complexity. Currently in the clinic, non-targeted drugs are commonly used; the most popular is indocynanine green (ICG) which is most used in tissue perfusion (Komorowska-Timek and Gurtner (2010); Yoshimatsu et al. (2021)) and lymph node (Kitai et al. (2005)) targeting applications but has limited utility in other surgical operations such as tumor targeting due to low specificity (Zhang et al. (2017)). Expanding the use of FGS to other surgeries require creating new contrast agents; however, this is an expensive process not helped by the fact that acceptable agents have tight constraints (Zhang et al. (2017)) on signal levels, the stokes shift

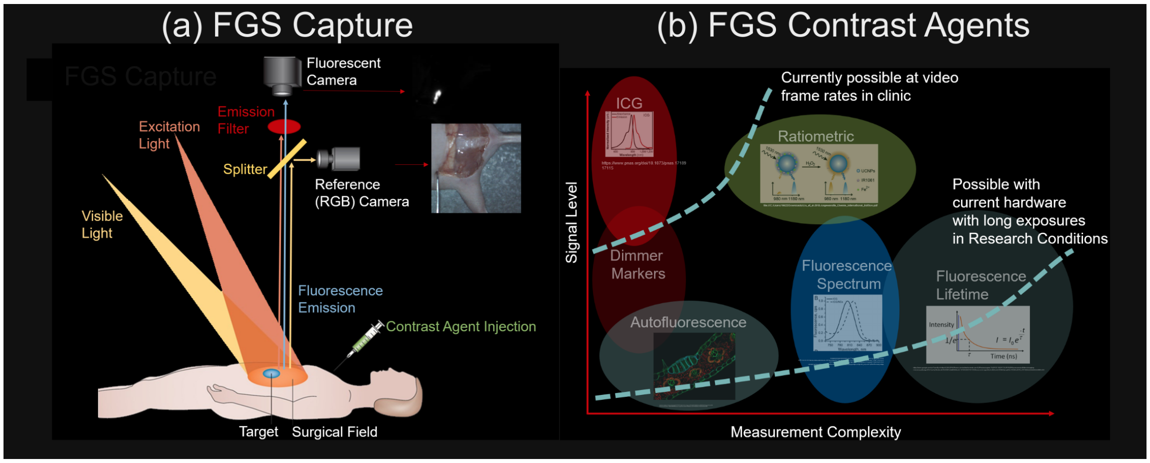


Figure 1.1: **Intro to FGS:** (a) shows an over view of the FGS capture system. (b) shows an overview of the landscape of different contrast agents used in FGS with respect to the signal level and measurement complexity. Currently, clinical use is restricted to the top left corner whereas all techniques are studied in research conditions. Transitioning the dimmer signals to video frame rates is one goal of video denoising in FGS. (Figure modified from images from: Vahrmeijer et al. (2013); Carr et al. (2018); Liu et al. (2018); Pegg et al. (2021); Gustafson et al. (2013); Instruments (2023))

(Lakowicz (2006)), specificity, and photo-bleaching to ensure hardware devices can produce a useful real-time video. For example, one promising contrast agent is the naturally occurring auto-fluorescence (Betz et al. (1999); Demos et al. (2004)) or that auto-fluorescent lifetimes (Marcu (2012); Weyers et al. (2022)) have been proposed to deal with low specificity of cancer targeting drugs (Belykh et al. (2018); Alfonso-García et al. (2022)), but low signal levels make clinical translation difficult, because of this current attempts rely on point scanning systems (Weyers et al. (2022); Noble Anbunesan et al. (2023)) that will inherently limit system resolution or require slow capture times so may not be suitable for all surgery types. One of the primary goals of FGS denoising is to increase the viability of these currently unused contrast agents.

Why Denoise in FGS?

There are a number of answers to *"Why should we pursue FGS denoising?"* The most simple low-risk answer is *"Denoising will make the output video look better, so it will be more useful to surgeons."* However, this is only a statement about the technology we can create, and while this dissertation focuses on exploring what and how to create this technology, the reason it is worth creating is not just to make FGS systems produce better quality video. Video denoising is a software solution to what is conventionally considered a hardware or drug development problem. Software has a fundamental advantage in scalability, flexibility, iteration cycle speed, so bringing this to FGS will bring economic benefits that may increase the viability of the technology and could drive investment by expanding what is addressable by FGS.

Video denoising allows better use of currently available hardware that may not be optimized for in-development drugs; this may lower the bar for what is considered an acceptable signal level of a drug cutting down on the number of iteration cycles required to find a working drug. The simplest measurements with the highest signal levels are what is currently used in clinics, other techniques are restricted to controlled research environments although they promise great improvements to surgeries. Currently, these more advanced or low signal techniques do not produce useful enough video to be used at video frame rates in clinic. Video denoising promises to increase their viability without expensive improvements to capture hardware or incredibly expensive drug development. One of the most economically promising techniques is the use of autofluorescence which may be able to replace drug based contrast agents. Drug development is an incredibly time intensive and expensive process with a high failure rate, so if autofluorescence can remove the need for this process, FGS as a whole will become more economically viable. However, autofluorescence is fundamentally a dim signal, so to be useful either hardware will need to be optimized or possibly invented to capture enough photons, or video denoising methods will need to be developed to make better use of the currently existing hardware. The video denoising path is potentially much less expensive and may be able to to generalize to different spectral bands whereas any improvements in hardware will likely be constrained just to improving visualization of one marker. This ability to generalize, where denoising techniques developed for one contrast agent will improve denoisers used for other contrast agents is one of the properties that makes video denoising a worthwhile pursuit. Fundamentally, this allows the technology to snowball, as more data is captured the video denoising models will be improved so low signal contrast agents will be viable expanding the number of surgeries done and the data collected. Snowballing is a crucial property for driving investment because it enables the hockey stick returns that are expected for many investment firms. Drawing investment will be necessary to fully realize the promise of FGS in clinical settings, and one route may be through the promise of snowballing using algorithmic methods.

2 MOTION ADAPTIVE DEBLURRING WITH SINGLE-PHOTON CAMERAS

When imaging dynamic scenes with a conventional camera, the finite exposure time of the camera sensor results in motion blur. This blur can be due to motion in the scene or motion of the camera. One solution to this problem is to simply lower the exposure time of the camera. However, this leads to noisy images, especially in low light conditions. In this chapter we propose a technique to address the fundamental trade-off between noise and motion blur due to scene motion during image capture. We focus on the challenging scenario of capturing images in low light, with fast moving objects. Our method relies on the strengths of rapidly emerging single-photon sensors such as single-photon avalanche diodes (SPADs).

Light is fundamentally discrete and can be measured in terms of photons. Conventional camera pixels measure brightness by first converting the incident photon energy into an analog quantity (e.g. photocurrent, or charge) that is then measured and digitized. When imaging in low light levels, much of the information present in the incident photons is lost due to electronic noise inherent in the analog-to-digital conversion and readout process. Unlike conventional image sensor pixels that require 100's-1000's of photons to produce a meaningful signal, SPADs are sensitive down to individual photons. A SPAD pixel captures these photons at an instant in time, with a time resolution of hundreds of picoseconds. Each photon detection can therefore be seen as an instantaneous event, free from any motion blur. Recently, single photon sensors have been shown to be useful when imaging in low light, or equivalently, imaging at high frame rates where each image frame is photon-starved (Ma et al. (2020)).

The data captured by a SPAD camera is thus quite different than a conventional camera: in addition to the two spatial dimensions, we also capture data on a high

This chapter is based on work done for the following publication (Seets et al. (2021)): Seets, Trevor, Atul Ingle, Martin Laurenzis, and Andreas Velten. "Motion adaptive deblurring with single-photon cameras." In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1945-1954. 2021.

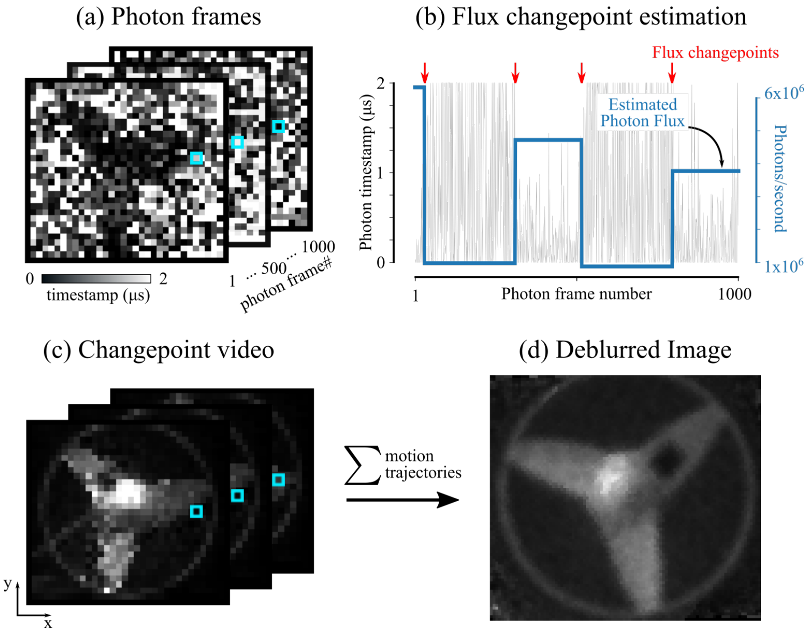


Figure 2.1: **Overview of our adaptive motion deblurring method.** (a) The photon frames captured from a rotating fan contain timestamps for the first detected photon in each frame. Note the smaller value of timestamps in brighter regions and vice versa. (b) Flux changepoints obtained after changepoint detection and flux estimation at pixel location highlighted in cyan in (a). Note the rapidly varying photon timestamps due to heavy-tailed nature of the raw timestamp data. (c) A changepoint video (CPV) maintains sharp edges while providing motion cues to estimate inter-frame motion trajectories; the temporal flux profile of the highlighted pixel in this CPV is the piecewise constant function shown in (b). (d) Integrating photons along estimated spatio-temporal motion trajectories generates a sharp image with high signal-to-noise ratio.

resolution time axis resulting in a 3D spatio-temporal tensor of photon detection events. We exploit this novel data format to deal with the noise-blur trade-off. Our key observation is that the photon timestamps can be combined dynamically, across space and time, when estimating scene brightness. If the scene motion is known *a priori*, photons can be accumulated along corresponding spatio-temporal trajectories to create an image with no motion blur. A conventional camera does not have this property because at capture time high frequency components are lost

(Raskar et al. (2006)). So even with known scene motion, image deblurring is an ill-posed problem.

Our method relies on dynamically changing exposure times, where each pixel can have its own set of exposure times. An overview of our approach is shown in Fig. 2.1. We propose using statistical changepoint detection to infer points in time where the photon flux at a given pixel changes from one steady state rate to another. This allows us to choose exposure times that adapt to scene motion. Changepoints allow us to track high contrast edges in the scene and align "high frequency" motion details across multiple image frames. We show that for the case of global motion (e.g., rotation) we can track the varying motion speeds of different scene pixels and combine photons spatially to create deblurred images. We also show that for the case of local scene motion, our method can provide robust estimates of flux changepoints around the edges of the moving objects that improves deblurring results obtained from downstream motion alignment and integration.

The locations of changepoints can be thought of as a spike-train generated by a neuromorphic event camera (Gallego et al. (2020)), but with three key differences: First, unlike an event camera, our method preserves original intensity information for each pixel. Second, the numbers and locations of events, chosen by our algorithm, adapt to each pixel, without the need for a hard-coded change threshold. Third, direct photon measurements are inherently more sensitive, less noisy, and have a higher time resolution than conventional cameras. Current cameras create an analog approximation of the quantized photon stream (usually a charge level in a capacitor) and then measure and digitize that analog quantity. This process introduces noise and does not take advantage of the inherent quantized properties of the photons. We therefore expect photon counting methods to have higher sensitivity, accuracy, and temporal resolution than analog cameras, when imaging in low light scenarios.

Single-Photon Sensitive Cameras for Machine Vision? Currently, single-photon sensitive image sensors are quite limited in their spatial resolution compared to conventional complementary metal–oxide–semiconductor (CMOS) and charge-coupled device (CCD) image sensors. However, single-photon image sensing is a rapidly developing field; recent work has demonstrated the feasibility of making megapixel resolution single-photon sensitive camera arrays (Morimoto et al. (2020); Gnanasambandam et al. (2019)). Moreover, silicon SPAD arrays are amenable to manufacturing at scale using the same photolithographic fabrication techniques as conventional CMOS image sensors. This means that many of the foundries producing our cellphone camera sensors today could also make SPAD array sensors at similar cost. Other current limitations of SPAD sensors are the small fraction of chip area sensitive to light (fill factor) resulting from the large amount of additional circuitry that is required in each pixel. Emerging 3D stacking technologies could alleviate this problem by placing the circuitry behind the pixel.

Limitations Our experimental demonstration uses a first generation commercial SPAD array that is limited to a 32×32 pixel spatial resolution. More recently 256×256 pixel commercial arrays have become available (Dhulla et al. (2019)). Although current SPAD cameras cannot compete with the image quality of commercial CMOS and CCD cameras, with the rapid development of SPAD array technology, we envision our techniques could be applied to future arrays with spatial resolution similar to that of existing CMOS cameras.

Contributions:

- We introduce the notion of *flux changepoints*; these can be estimated using an off-the-shelf statistical changepoint detection algorithm.
- We show that flux changepoints enable inter-frame motion estimation while preserving edge details when imaging in low light and at high speed.
- We show experimental demonstration using data acquired from a commercially available SPAD camera.

2.1 Related Work

Motion Deblurring Motion deblurring for existing cameras can be performed using blind deconvolution (Levin et al. (2008)). Adding a fast shutter (“flutter shutter”) sequence can aid this deconvolution task (Raskar et al. (2006)). We push the idea of a fluttered shutter to the extreme limit of individual photons: our image frames consist of individual photon timestamps allowing dynamic adaptation of sub-exposure times for the shutter function. Our deblurring method is inspired by burst photography pipelines used for conventional CMOS cameras. Burst photography relies on combining frames captured with short exposure times (Hasinoff et al. (2016)), resulting in large amounts of data that suffer from added readout noise. Moreover, conventional motion deblurring methods give optimal performance when the exposure time is matched to the true motion speed which is not known a priori.

Event-based Vision Sensors Event cameras directly capture temporal changes in intensity instead of capturing scene brightness (Gallego et al. (2020)). Although it is possible to create intensity images from event data in post-processing (Bardow et al. (2016)), our method natively captures scene intensities at single-photon resolution: the "events" in our sensing modality are individual photons. The notion of using photon detections as "spiking events" has also been explored in the context of biologically inspired vision sensors (Zhu et al. (2020); Afshar et al. (2020)). We derive flux changepoints from the high-resolution photon timestamp data. Due to the single-photon sensitivity, our method enjoys lower noise in low light conditions, and pixel-level adaptivity for flux changepoint estimation.

Deblurring Methods for Quanta Image Sensors There are two main single-photon detection technologies for passive imaging: SPADs (Laurenzis (2019); Ingle et al. (2019)) and quanta image sensors (QIS) (Fossum et al. (2016)). Although our proof-of-concept uses a SPAD camera, our idea of adaptively varying exposure times can be applied to QIS data as well. Existing motion deblurring algorithms for QIS (Gyöngy et al. (2017); Gyongy et al. (2018); Ma et al. (2020)) rely on a fixed integration window to sum the binary photon frames. However, the initial step of picking the size of this window requires *a priori* knowledge about the motion speed and scene brightness. Our technique is therefore complementary to existing motion deblurring algorithms. For example, our method can be considered as a generalization of the method in Istvan et al. (2015) which uses two different window sizes. Although we use a classical correlation-based method for motion alignment, the sequence of flux changepoints generated using our method can be used with state-of-the-art align-and-merge algorithms (Ma et al. (2020)) instead.

2.2 SPAD Image Formation Model

SPADs are most commonly used in synchronization with an active light source such as a pulsed laser for applications including LiDAR and fluorescence microscopy. In contrast, here we operate the SPAD passively and only collect ambient photons from the scene. In this section, we describe the imaging model for a single SPAD pixel collecting light from a fixed scene point whose brightness may vary as a function of time due to camera or scene motion.

Pixelwise Photon Flux Estimator

Our imaging model assumes a frame-based readout mechanism: each SPAD pixel in a *photon frame* stores at most one timestamp of the first captured ambient photon. This is depicted in Fig. 2.1(a). Photon frames are read out synchronously from the entire SPAD array; the data can be read out quite rapidly allowing photon frame rates of 100s of kHz (frame times on the order of a few microseconds).

Let N_{pf} denote the number of photon frames, and T_{pf} be the frame period. So the total exposure time is given by $T = N_{pf}T_{pf}$. We now focus on a specific pixel in the SPAD array. In the i^{th} photon frame ($1 \leqslant i \leqslant N_{pf}$), the output of this pixel is tagged with a photon arrival timestamp t_i relative to the start of that frame.* If no photons are detected during a photon frame, we assume $t_i = T_{pf}$.

Photon arrivals at a SPAD pixel can be modeled as a Poisson process (Hasinoff (2014)). It is possible to estimate the intensity of this process (i.e. the perceived brightness at the pixel) from the sequence of photon arrival times (Laurenzis (2019); Ingle et al. (2019)). The maximum likelihood brightness estimator $\widehat{\Phi}$ for the true

*In practice, due to random timing jitter and finite resolution of timing electronics, this timestamp is stored as a discrete fixed-point value. The SPAD camera used in our experiments has a 250 picosecond discretization.

photon flux Φ is given by Laurenzis (2019):

$$\widehat{\Phi} = \frac{\sum_{i=1}^{N_{pf}} \mathbf{1}(t_i \neq T_{pf})}{q \sum_{i=1}^{N_{pf}} t_i} \tag{2.1}$$

where q is the sensor's photon detection efficiency and **1** denotes a binary indicator variable.

This equation assumes that the pixel intensity does not change for the exposure time T. The assumption is violated in case of scene motion. If we can determine the temporal locations of intensity changes, we can use still use Eq. (2.1) to estimate a time-varying intensity profile for each pixel. In the next section we introduce the idea of *flux changepoints* and methods to locate them with photon timestamps.

Flux Changepoints

Photon flux at a given pixel may change over time in case of scene motion. This makes it challenging to choose pixel exposure times *a priori*: ideally, for pixels with rapidly varying brightness, we should use a shorter exposure time and vice versa. We propose an algorithm that dynamically adapts to brightness variations and chooses time-varying exposure times on a per pixel basis. Our method relies on locating temporal change locations where the pixel's photon flux has a large change: we call these *flux changepoints*.

In general, each pixel in the array can have different numbers and locations of flux changepoints. For pixels that maintain constant brightness over the entire capture period (e.g. pixels in a static background), there will be no flux changepoints detected and we can integrate photons over the entire capture time T. For pixels with motion, we assume that the intensity between flux changepoints is constant, and we call these regions *virtual exposures*. Photons in each virtual exposure are aggregated to create a piecewise constant flux estimate. The length of each virtual exposure will depend on how quickly the local pixel brightness varies over time, which in turn, depends on the true motion speed.

An example is shown in Fig. 2.1(b). Note that the photon timestamps are rapidly varying and extremely noisy due to a heavy-tailed exponential distribution. Our changepoint detection algorithm detects flux changepoints (red arrows) and estimates a piecewise constant flux waveform for the pixel (blue plot). In the example shown, five different flux levels are detected.

Changepoint Detection

Detecting changepoints is a well studied problem in the statistics literature (Basseville et al. (1993); Tartakovsky et al. (2014)). The goal is to split a time-series of measurements into regions with similar statistical properties. Here, our time series is a sequence of photon timestamps at a pixel location, and we would like to find regions where the timestamps have the same mean arrival rate, i.e., the photon flux during this time is roughly constant.

Using the sequence of photon arrival times $\{t_i\}_{i=1}^{N_{pf}}$, we wish to find a subset $\{t_{l_1}, \ldots, t_{l_L}\}$ representing the flux changepoints. For convenience, we let the first and the last flux changepoint be the first and the last photons captured by the pixel ($l_1 := 1$ and $l_L = N_{pf}$).

Offline changepoint detection is non-causal, i.e., it uses the full sequence of photon timestamps for a pixel to estimate flux changepoints at that pixel. In Suppl. Note 2.6 we describe an online changepoint detection method that can be used for real-time applications.

For a sequence of photon timestamps, we solve the following optimization problem (Truong et al. (2020)) (see Suppl. Note 2.6):

$$(l_i^*)_{i=1}^{L} = \arg\min_{l_1,..,l_L} \sum_{i=1}^{L-1} \left[-\log(\widehat{\Phi}_i) \sum_{j=l_i}^{l_{i+1}} \mathbf{1}(t_j \neq T_{pf}) \right] + \lambda L \tag{2.2}$$

where $\widehat{\Phi}_i$ is the photon flux estimate given by Eq. (2.1) using only the subset of photons between times t_{l_i} and $t_{l_{i+1}}$. Here λ is the penalty term that prevents overfitting by penalizing the number of flux changepoints. For larger test images we use the BottomUp algorithm (Truong (2017 (accessed june 20, 2020)) which is approximate but has a faster run time. For lower resolution images we use the

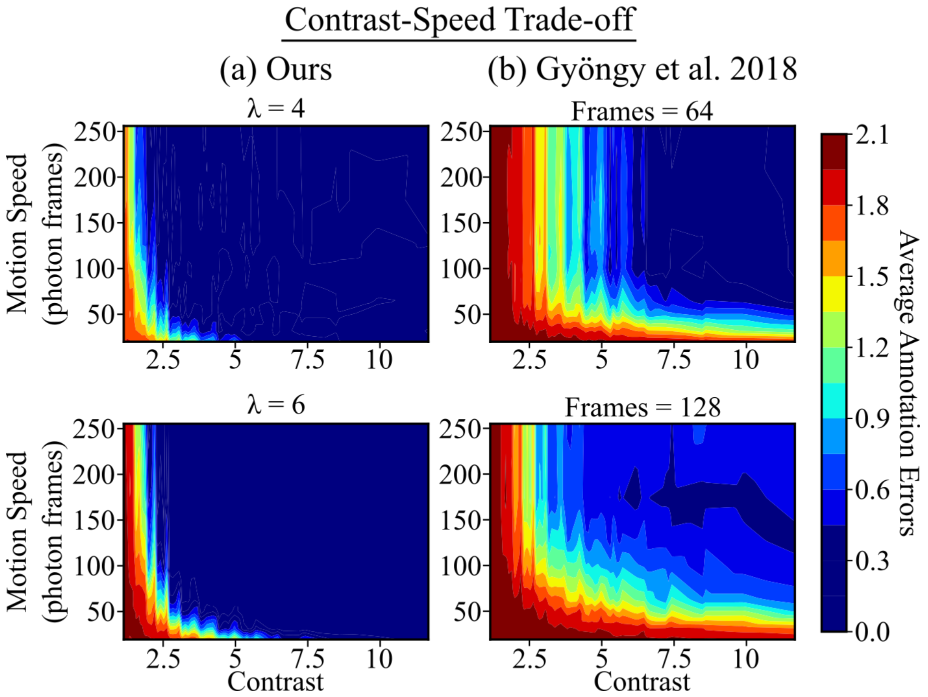


Figure 2.2: **Contrast-Speed Trade-off.** We simulate a single SPAD pixel measuring a pulse-shaped photon flux signal to analyze the contrast-speed trade-off when detection motion. Our method (a) based on PELT changepoint detection adapts to a wider range of contrast and speed combinations than the comparison method (b) from Gyongy et al. (2018).

Pruned Exact Linear Time (PELT) algorithm (Killick et al. (2012)) which gives an exact solution but runs slower. See Suppl. Section 2.9 for results using the BottomUp algorithm.

Flux Changepoints for QIS Data The cost function in Eq. (2.2) applies to photon timestamp data, but the same idea of adaptive flux changepoint detection can be used with QIS photon count data as well. A modified cost function that uses photon counts (0 or 1) is derived in Suppl. Note 2.6.

Single-Pixel Simulations

There is a trade-off between contrast and motion speed when trying to detect changepoints. For example, it is harder to detect the flux change with a fast moving object with a lower contrast with respect to its background. To evaluate this trade-off, we simulate a single SPAD pixel for 800 photon frames with a time varying flux signal with a randomly placed pulse wave. We use the pulse width as a proxy for motion speed, and vary the contrast by varying the ratio of the pulse height. We measure the absolute difference between the number of changepoints detected by our algorithm and the true number of flux changes (which is exactly 2 for a single pulse). We call this "annotation error."

Fig. 2.2 shows the annotation errors for different values of contrast and motion speeds for two different algorithms. For each set of contrasts and motion speeds we display the average number of annotation errors over 120 simulation runs. We use the PELT algorithm to detect flux changepoints. For comparison, we also show the fixed windowing approach used by Gyongy et al. (2018)(Agresti and Coull (1998)). The changepoint detection algorithm is able to adapt to a wider range of contrasts and speeds.

2.3 Pixel-Adaptive Deblurring

Changepoint Video

Using methods described in the previous section we locate flux changepoints for each pixel in the image. The changepoints for each pixel represent when a new virtual exposure starts and stops; photons within each virtual exposure can be used to estimate photon flux using Eq. (2.1). We call this collection of piecewise constant functions over all pixels in the array the *changepoint video* (CPV).

The CPV does not have an inherent frame rate; since each pixel has a continuous time piecewise constant function, it can be sampled at arbitrarily spaced time instants in $[0, T]$ to obtain any desired frame rate. We sample the CPV at non-uniform time intervals using the following criterion. Starting with the initial frame sampled at $t = 0$, we sample the subsequent frames at instants when at least 1% of the pixel values have switched to a new photon flux. This leads to a variable frame rate CPV that adapts to changes in scene velocity: scenes with fast moving objects

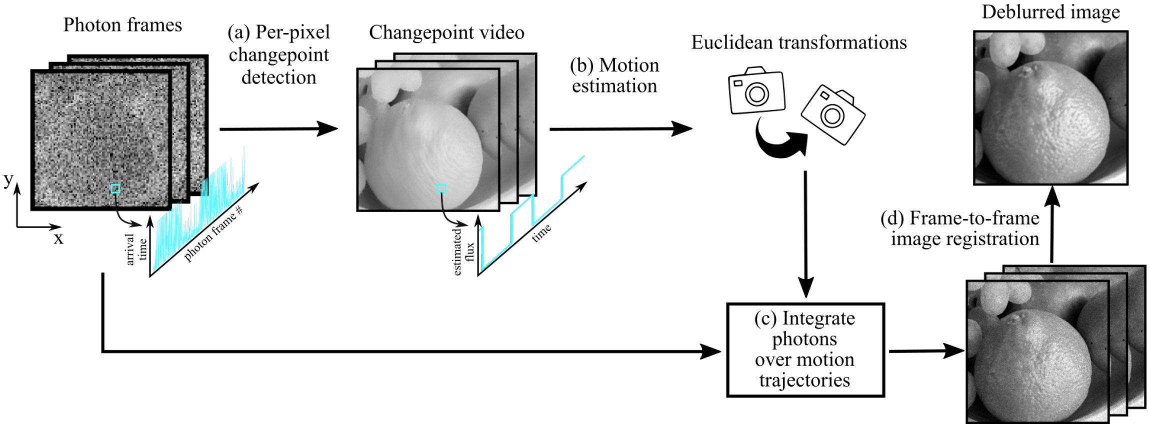

Photon frames
(a) Per-pixel changepoint detection
Changepoint video
(b) Motion estimation
Euclidean transformations
Deblurred image
y
x
arrival time
photon frame #
estimated flux
time
(d) Frame-to-frame image registration
(c) Integrate photons over motion trajectories

Figure 2.3: **Changepoint video deblurring**. Photon timestamps for each pixel location in the 3D timestamp tensor are analyzed for changepoints (a) to generate a changepoint video. Pixels may have different numbers of changepoints, and hence, dynamically changing frame-rate over time. We estimate motion between successive frames of the changepoint video (b) and estimate motion parameters. Finally, using this motion estimate, we sum photons (c) from the photon frames along motion trajectories resulting in a deblurred video. Frames in the deblurred video are registered and summed (d) to obtain the final deblurred image. (Original images from FreeImages.com)

will have a higher average CPV frame rate.

The CPV preserves large brightness changes in a scene. For example, the edges of a bright object moving across a dark background remain sharp. However, finer texture details within an object may appear blurred.

Spatio-Temporal Motion Integration

We use global motion cues from the CPV and then combine the motion estimates with the photon frames to create a deblurred video. The overall flowchart is shown in Fig. 2.3.

We use a correlation-based image registration algorithm to find the motion between consecutive frames in the CPV. For each location $\mathbf{p} = [x, y]$ in the i^{th} CPV frame, we assume it maps to a location $\mathbf{p}'$ in $(i+1)^{\text{st}}$ CPV frame. We assume that

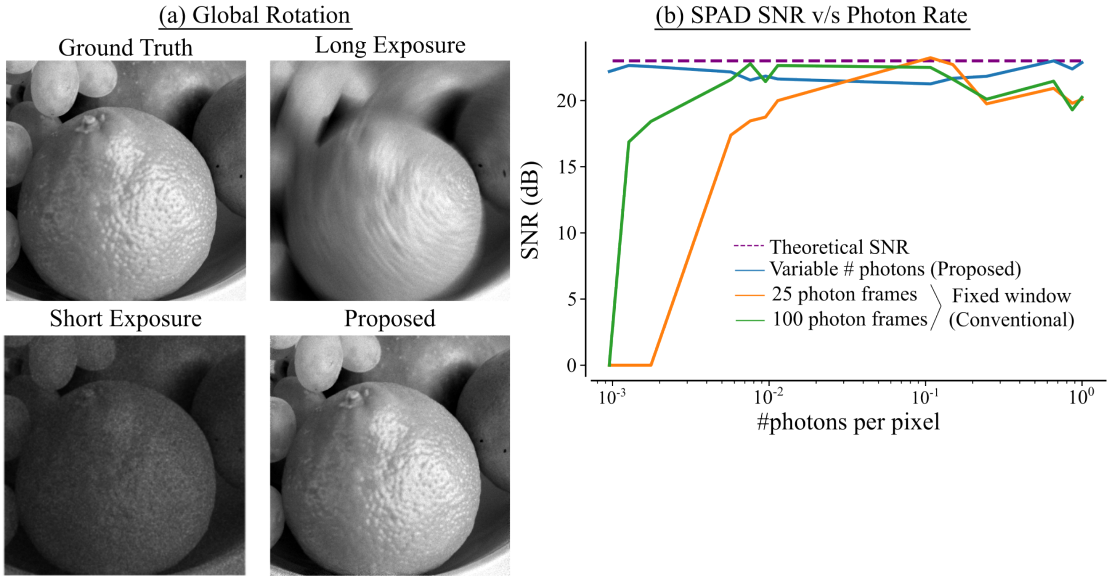


Figure 2.4: **Comparison of SNR for different deblurring window sizes.** (a) A ground truth video of a rotating clementine used for creating simulated SPAD photon frames. A long exposure image is quite blurry while a short exposure image is very noisy. Our deblurring algorithm strikes a balance between noise and blur to get a sharp high-quality image. (b) By varying the brightness of the clementine scene, we compare the simulated SNR using our method compared to a conventional method with fixed windows. Notice the proposed method stays above 20 dB at all photon rates, while the fixed photon window SNRs decrease in the low photon count regime. (Original clementine image from FreeImages.com)

the mapping is a linear transformation:

$$A\mathbf{p} = \mathbf{p}'. \tag{2.3}$$

We constrain A to represent planar Euclidean motion (rotation and translation). Let θ be the amount of rotation centered around a point $[r_x, r_y]$ and let $[\tau_x, \tau_y]$ be the translation vector. Then A has the form:

$$A = \begin{bmatrix} \cos\theta & \sin\theta & r_x(1-\cos\theta)+r_y\sin\theta+\tau_x \\ -\sin\theta & \cos\theta & r_y(1-\cos\theta)-r_x\sin\theta+\tau_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.4}$$

We apply the enhanced correlation coefficient maximization algorithm (Evangelidis and Psarakis (2008)) to estimate the transformation matrix A for consecutive pairs of CPV frames $i \to i+1$. A sequence of frame-to-frame linear transformations generates arbitrarily shaped global motion trajectories. We aggregate the original photon frame data along these estimated spatio-temporal motion trajectories.

We assume that the rotation center, $[r_x, r_y]$, is the middle of the image and a change in rotation center can be modeled as a translation. We solve for θ, τ_x, and τ_y which we linearly interpolate. Then using the interpolated motion parameters and Eq. (2.3), we align all photon frames corresponding to the time interval between CPV frames $i \to i+1$ and sum these frames to get a photon flux image by using Eq. (2.1) at each pixel. This generates a motion deblurred video with the same frame rate as the CPV, but with finer textures preserved as shown in Fig. 2.3.

If the final goal is to obtain a single deblurred image, we repeat the steps described above on consecutive frames in the deblurred video, each time decreasing the frame rate by a factor of 2, until eventually we get a single image. This allows us to progressively combine photons along spatial-temporal motion trajectories to increase the overall signal to noise ratio (SNR) and also preserve high frequency details that were lost in the CPV.

Our method fails when motion becomes too large to properly align images, especially at low resolutions. It can also fail when not many flux changepoints are detected, this will occur mainly due to a lack of photons per pixel of movement. In the worst case, if not enough changepoints are detected, the result of the algorithm will look similar to a single long exposure image.

The method of aligning and adding photon frames is similar to contrast maximization algorithms used for event cameras (Gallego et al. (2018)). However, unlike event camera data, our method relies on the CPV which contains both intensity and flux changepoints derived from single-photon timestamps.

Handling Multiple Moving Objects To handle multiple moving objects on a static background, we implement a method similar to Gyongy et al. (2018) and combine it with our CPV method. We cluster the changepoints for different objects using a density-based spatial clustering algorithm (DBSCAN) (Ester et al. (1996)). For each cluster, we then create a bounding box, isolating different moving objects. We then apply our motion deblurring algorithm on each object individually, before stitching together each object with the areas that are not moving in the CPV. The clustering step also denoises by rejecting flux changepoints not belonging to a cluster. Depending on the application this rejection step may remove important information of small objects; when these objects are important likely operating on the changepoint video for an end computer vision task may be more robust.

Simulations

Starting with a ground truth high resolution, high frame rate video, we scale the video frames to units of photons per second and generate photon frames using exponentially distributed arrival times. We model a photon frame readout SPAD array with 8000 bins and bin width of 256 ps.

We first simulate a rotating clementine by applying successive known rigid transformations to an image and generating SPAD data by scaling the transformed images between 10^4 and 10^8 photons per second. We rotate the image by 0.1° for every 10 generated photons for a total of 1000 photons. We use the BottomUp algorithm (Truong et al. (2020)) with $\lambda = 5$ for the changepoint detection step. The results are shown in Fig. 2.4(a). Our method captures sharp details on the clementine skin while maintaining high quality.

Fig. 2.4(b) shows quantitative comparisons of SNR for different deblurring methods. The conventional approach to deblur photon data uses a fixed frame rate; we use two different window lengths for comparison. We compute the SNR of the deblurred imaging using the $\ell 2$ (root-mean-squared) distance from the ground truth to and repeat this over a range of photon flux levels. We keep the total number of photons captured approximately constant by extending the capture time for darker flux levels. Our method dynamically adapts to motion and lighting so we are able to reconstruct with high SNR even in photon starved regimes where the SNR of the fixed window methods degrades rapidly.

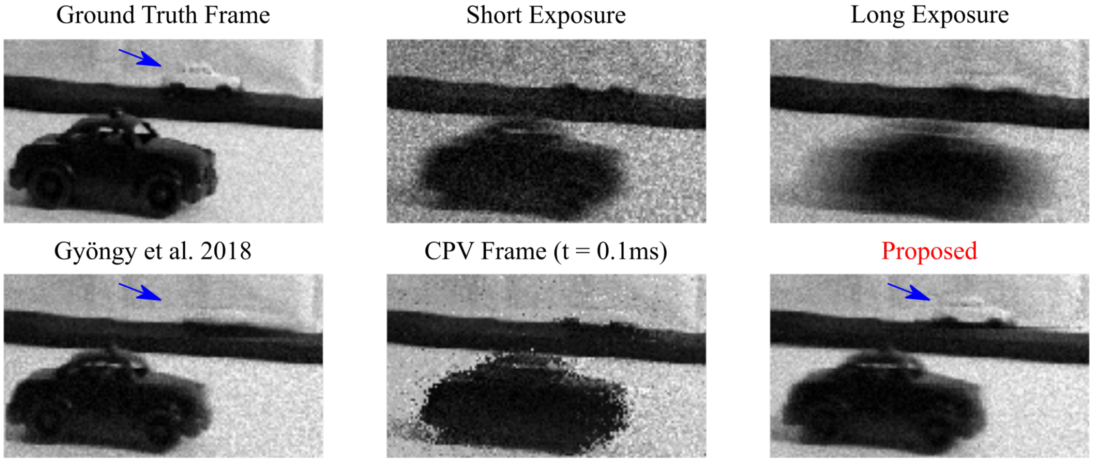


Figure 2.5: **Simulated Motion Deblurring for Multiple Moving Objects.** We simulate SPAD data from video of toy cars, a fast moving black car and slow moving white car. (Top row) A sample frame from the ground truth frame sequence is shown. The short and long exposure images show the results of using integrating the first 75 and 250 photon frames, respectively. Notice that the short exposure preserves the black car while the white car is quite noisy, on the other hand, the long average blurs the black car but preserves details of the white one better. (Bottom row) The method of Gyongy et al. (2018), fails to reconstruct the white car (blue arrow) due to its low contrast. A sample frame from our changepoint video shows both moving cars. Finally, our deblurring algorithm is able to reconstruct both the black and white car with negligible motion blur.

To simulate multi-object motion, we captured a ground truth video of two toy cars rolling down a ramp at different speeds. The video frame pixels are then scaled between 10^5 and 10^6 photons per second and a total of 690 photon frames are generated. A bright slow moving car has a contrast of 1.2 with respect to the background, and moves 48 pixels over the duration of video. The dark car has a contrast of 5.0 with the background, and moves 143 pixels. We use the PELT algorithm (Killick et al. (2012)) with $\lambda = 6$ for the changepoint detection step. We

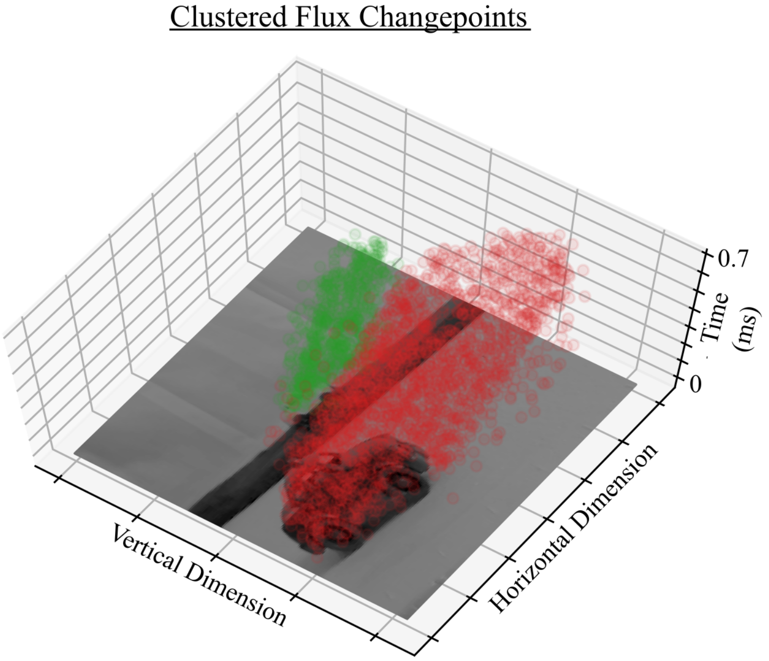
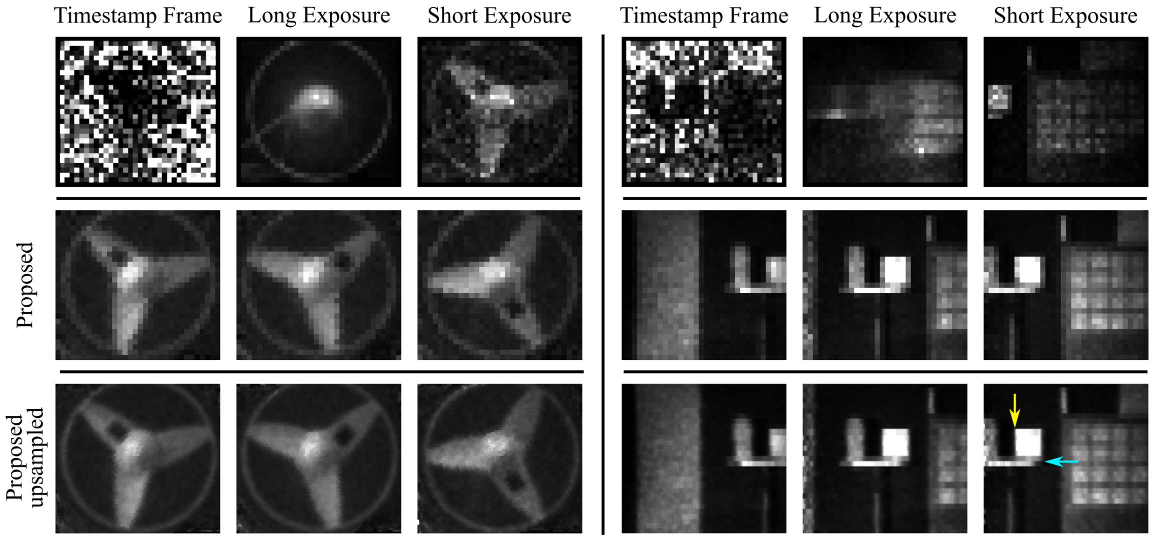


Figure 2.6: **Clustered Flux Changepoints.** Two clusters of flux changepoints are detected using frames from the changepoint video for the toy car scene. These changepoint clusters are used for segmenting the moving cars from the static background.

use $\epsilon = 7.5$ and MinPts $= 40$ in the DBSCAN clustering algorithm. The resulting deblurred images are shown in Fig. 2.5. Observe that the method of Gyongy et al. (2018) blurs out the low contrast white car in the back. Our method assigns dynamically changing integration windows extracted from the CPV to successfully recover both cars simultaneously with negligible motion blur. The changepoint clusters used for segmenting cars from the static background in our method are are shown in Fig. 2.6.

2.4 Experiments

We validate our method using experimental data captured using a 32×32 InGaAs SPAD array from Princeton Lightwave Inc., USA. The pixels are sensitive to near



(a) “Fan” Scene

(b) “Checkerboard” Scene

Figure 2.7: **Experimental Results.** The first row shows a single raw data frame from the photon timestamp tensor; each photon has an associated timestamp with a 250 ps bin resolution. Integration over a long exposure (2 ms for the fan and 0.2 ms for checkerboard scene) this gives a low noise but blurry result. Using a short exposure time (40 µs for both scenes) produces very noisy results. The second row shows a sequence of three frames from the final deblurred video. The third row shows the result with upsampling. Note that in the checkerboard scene due to purely horizontal motion, some vertical edges (yellow arrow) are sharper but not the horizontal edges (cyan arrow).

infrared and shortwave infrared (900 nm–1.6 µm wavelengths). The SPAD array operates in a frame readout mode at 50,000 photon frames per second. Each photon frame exposure window is 2 µs and sub-divided into 8000 bins, giving a temporal resolution of 250 ps per bin (Technologies (2012 (accessed june 20, 2020)).

For this low resolution experimental data, we zero-order hold upsample both the timestamp frames and the CPV before step (b) in Fig. 2.3 in the spatial dimensions. We then use upsampled photon frames for step (c) resulting in sharper images. Upsampling before motion integration allows photons that are captured in the same pixel to land in a larger space during the motion integration step.

Fig. 2.7 shows results from two different scenes. The “fan” scene shows the performance of our algorithm with fast rotation[†]. The optimal exposure time in this case depends on the rotation speed. Our method preserves the details of the fan blades including the small black square patch on one of the fan blades. The "checkerboard" scene shows deblurring result with purely horizontal global motion. Note that our method is able to resolve details such as the outlines of the squares on the checkerboard.

The last row in Fig. 2.7 shows the upsampled results. Benefits of upsampling are restricted to the direction of motion. The "fan" dataset is upsampled 9× compared to the original resolution. The "checkerboard" dataset is upsampled 4×; this is because the motion is limited to the horizontal dimension. Note that some of the details in the vertical edges are sharp, but horizontal edges remain blurry. Conceptually, our upsampling method can help fill in missing information missed due to the low fill factors of our camera, around 10 percent. In our case, each pixel contains the same high frequency information of a camera with 10x the resolution and the same pixel pitch. Our upsampling method retains this high frequency information over the motion of the object, allowing for upsampling of about 3x in each dimension. Note our method can only fill in information that is missed due to the poor fill factor of the detector and may not work as sensors improve to high fill factors.

2.5 Discussion and Future Work

Euclidean Motion Assumption In the case of camera shake, most of the motion will come from small rotations in the camera that result in 2D translations and rotations in the image frames. The short distance translations of shaking camera would cause translations in the frames that are similar in nature, but smaller in magnitude.

[†] The background behind the fan is covered with absorbing black felt material. This allows us to treat the data as having global rotation, because there is hardly any light captured from the background.

Translation of the camera over larger distances would result in parallax while motion within the scene can result in potentially non-euclidean motion. In these cases our model captures scene changes only approximately. It applies to frame to frame motion over short time-scales and limited to regions in the scene. Ideas for extending our model to deal with larger motion will be the subject of future work.

Dealing with Local Motion The techniques presented in this chapter assume multiple moving objects exhibiting euclidean motion, with no occlusions. We can extend our approach to more complex motions. We can use a patch-wise alignment and merging methods to deal with more complex local motion and occlusions (Hasinoff et al. (2016); Ma et al. (2020)).

Deblurring algorithms developed for event camera data can be adapted to SPAD data, because the flux changepoints represent changes in brightness similar to the output of event cameras. Current event camera algorithms are able to recover complex motion in scenes (Gallego et al. (2018); Stoffregen et al. (2019)), and they could be improved with a fusion based approach where image intensity information is also available (Gehrig et al. (2019)).

Data Compression With increasing number of pixels, processing photon frames with high spatio-temporal resolution will be quite resource intensive. Our online changepoint method takes some initial steps towards a potential real-time implementation. The CPV can be used for video compression with variable frame rate: by tuning the regularization parameter of the changepoint detection algorithm a tradeoff between image fidelity and data rate can be achieved.

Acknowledgments This material is based upon work supported by the Department of Energy / National Nuclear Security Administration under Award Number DE-NA0003921, the National Science Foundation (GRFP DGE-1747503, CAREER 1846884), DARPA (REVEAL HR0011-16-C-0025) and the Wisconsin Alumni Research Foundation.

Disclaimer This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied,

2.6 Supplementary Section: Flux Changepoint Detection

Offline Algorithm: Cost Function Derivation

Consider a set of photon time stamp measurements $\{x_i\}_{i=1}^{N}$. Here each x_i is a valid measurement, and in the frame-readout capture mode, described in the main text, this is different than the t_i's. If no photon is detected in a frame we add the frame length to the next detected photon. We do this so each x_i will be i.i.d. and distributed exponentially. We again wish to find a set of change points, $\{x_{l_1}, \dots, x_{l_L}\}$. In general, the optimization problem for changepoint detection is given by Eq. (P2) in Truong et al. (2020):

$$(l_i^*)_{i=1}^{L} = \underset{l_1,..,l_L}{\arg\min} \sum_{i=1}^{L-1} c(x_{l_i \dots l_{i+1}}).+\lambda L \tag{2.5}$$

The summation term represents the likelihood that each segment in between changepoints come from the same underlying distribution, while the regularization term is needed because the number of changepoints are not known *a priori*. For our case $c(\cdot)$ is the negative log likelihood for a set of exponentially distributed measurements. Let $f(x)$ be the exponential density function with rate parameter Φ, and let $\widehat{\Phi}_i$ be the maximum likelihood estimate for Φ for the set of measurements $\{x_{l_i} \dots x_{l_{i+1}}\}$. Note that the maximum likelihood estimator maximizes the log likelihood. To derive $c(\cdot)$, we begin with Eq. (C1) from Truong et al. (2020):

$$c(x_{l_i \ldots l_{i+1}}) = -\max_{\Phi} \sum_{j=l_i}^{l_{i+1}} \log f(x_j|\Phi) \tag{2.6}$$

$$= -\sum_{j=l_i}^{l_{i+1}} \log f(x_j|\widehat{\Phi}_i) \tag{2.7}$$

$$= -\sum_{j=l_i}^{l_{i+1}} \log \widehat{\Phi}_i e^{-\widehat{\Phi}_i x_j} \tag{2.8}$$

$$= -\sum_{j=l_i}^{l_{i+1}} \log \widehat{\Phi}_i + \sum_{j=l_i}^{l_{i+1}} \widehat{\Phi}_i x_j \tag{2.9}$$

$$= -(l_{i+1} - l_i) \log \widehat{\Phi}_i + (l_{i+1} - l_i) \tag{2.10}$$

where the last line comes from the fact that $\widehat{\Phi}_i = \frac{(l_{i+1}-l_i)}{\sum_{j=l_i}^{l_{i+1}} x_j}$. Plugging Eq. (2.10) into Eq. (2.5), the last term sums to a constant N and can be dropped from the optimization. Then we convert to the direct measurments t_i by expanding out where no photons where found to get Eq. (2.2).

QIS: Offline Cost Function

A quanta image sensor (QIS) is another sensor type capable of measuring single photons. Unlike a SPAD, the QIS senor only gives a binary output for each photon-frame corresponding to whether or not a photon was detected. This analysis also applies to binary SPAD arrays because they generate photon frames the same statistics as QIS sensors. Note that we can convert our experimental SPAD data to QIS data by stripping the SPAD data of the timing information. Let $n_i = 0$ if the i^{th} QIS photon-frame detects no photons and $n_i = 1$ otherwise. Let τ_b be the temporal bin width for each photon-frame. Suppose the jot is exposed to a flux of Φ, then the probability of detecting a photon during photon-frame i is:

$$p = P(n_i = 1) = 1 - e^{-q\Phi\tau_b} \tag{2.11}$$

Where q is the quantum efficiency. We can model measuring multiple photon-frames with the QIS jot as a Bernoulli Trial, with probability of success given by Eq. 2.11. For a set of N photon-frames the maximum likelihood estimator, $\hat{\Phi}_{QIS}$, is given by Ingle et al. (2019),

$$\hat{\Phi}_{QIS} = \frac{-1}{q\tau_b} \ln(1 - \widehat{p}) \tag{2.12}$$

$$\widehat{p} = \frac{\sum_{i=0}^{N} n_i}{N} \tag{2.13}$$

This flux estimator should also be used in SPAD sensors under very high fluxes, where their is significant probability of detecting more than one photon in a period equal to the SPAD's time quantization. Similarly, the MLE in Eq. 2.1 can be used in low light conditions for a QIS sensor.

We derive the changepoint cost function in the raw data domain. Following the steps of the earlier derivation, with $f(n_i)$ being the Bernoulli distribution with parameter p:

$$c_{QIS}(x_{l_i \ldots l_{i+1}}) = -\max_{\Phi} \sum_{j=l_i}^{l_{i+1}} \log f(n_j|\Phi) \tag{2.14}$$

$$= -\sum_{j=l_i}^{l_{i+1}} \log f(n_j|\widehat{\Phi}_i) \tag{2.15}$$

$$= -\log(\widehat{p}_i) \sum_{j=l_i}^{l_{i+1}} n_j - \log(1 - \widehat{p}_i) \sum_{j=l_i}^{l_{i+1}} 1 - n_j \tag{2.16}$$

Where $\widehat{p}_i = \frac{\sum_{j=l_i}^{l_{i+1}} n_j}{l_{i+1} - l_i}$.

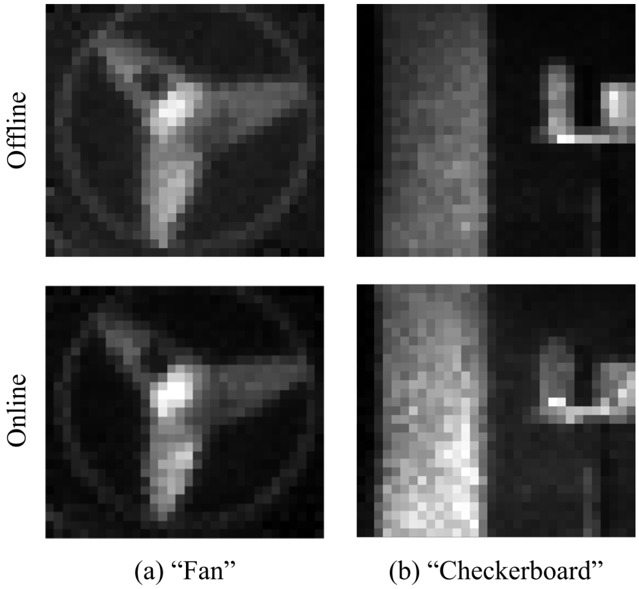


Figure 2.8: **Comparing online vs. offline changepoint detection.** We processed the two experimental datasets using our online and offline changepoint detection algorithms. There is a slight loss of edge details when the online algorithm is used but it can potentially operate on chip without needing to store large amounts of photon data.

Online Flux Changepoint Detection Algorithm

Offline changepoint detection is suitable for offline applications that capture a batch of photon frames and generate a deblurred image in post-processing. In some applications that require fast real-time feedback (e.g. live deblurred video) or on-chip processing with limited frame buffer memory, online changepoint detection methods can be used. We use a Bayesian online changepoint detection method (Adams and MacKay (2007)). This algorithm calculates the joint probability distribution of the time since the last flux changepoint. For exponentially distributed data, it uses the posterior predictive distribution which is a Lomax distribution. We assume that the flux changepoints appear uniformly randomly in the exposure window T. Because detecting a flux changepoint after only one photon is difficult we use a look-behind window that evaluates the probability of the photon 20–40 photon frames into the past as being a flux changepoint. Using a look-behind window

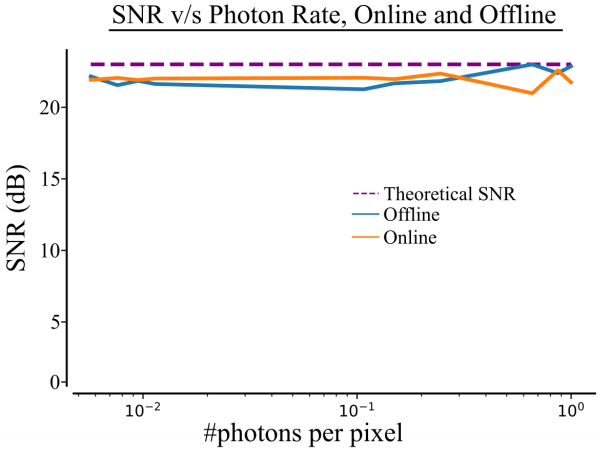


Figure 2.9: **Online vs. Offline Rotating clementine SNR.** We run the same experiment as in Fig. 2.4, with the rotating clementine at many light levels. Note that the resulting SNRs of the two methods are quite similar. Note the dip in the performance of the online method at high flux is likely due to noise.

greatly increases detection accuracy and introduces only minor latency (on the order of tens of microseconds). We also found that it is helpful to use a small spatial window that spreads out flux changepoints in space to increase the density of changepoints. In general, online detection will work better for slower motion as the algorithm learns from past data. We compare online and offline detection in Suppl. Note 2.8.

We use a Bayesian changepoint detection algorithm shown in Algorithm 1 of (Adams and MacKay (2007)). Here we derive the posterior predictive distribution used in Step 3 of their algorithm. We use a $\mathsf{Gamma}(\alpha, \beta)$ prior for Φ. Let $\mathbf{x} := \{x_i\}_{i=1}^{N}$. It can be shown that $\Phi|\mathbf{x} \sim \mathsf{Gamma}(\alpha + N, \beta + \sum_{i=1}^{N} x_i)$. The predictive posterior density is given by:

$$f_{x_{N+1}|\mathbf{x}}(y|\mathbf{x}) = \int_0^\infty f_{x_{N+1}|\mathbf{fl}}(y|\Phi) f_{\mathbf{fl}|\mathbf{x}}(\Phi|\mathbf{x}) d\Phi \tag{2.17}$$

$$= \int_0^\infty \Phi e^{-\Phi y} \frac{(\beta + \sum x_i)^{\alpha+N}}{\Gamma(\alpha + N)} \Phi^{\alpha+N-1} e^{-(\beta+\sum_i x_i)\Phi} d\Phi \tag{2.18}$$

$$= \frac{(\beta + \sum_i x_i)^{\alpha+N}(\alpha + N)}{(\beta + y + \sum_i x_i)^{\alpha+N-1}} \tag{2.19}$$

which is a Lomax density with shape parameter $\alpha+N$ and scale parameter $\beta+\sum_i x_i$. For our data we used a $\mathsf{Lomax}(1, 100)$ in Step 3 and $H(\cdot) = 40$ in Steps 4 and 5 of Algorithm 1 in Adams and MacKay (2007).

For online detection we use code modified from Kulick et al. (2020). Fig. 2.8 shows the results of using the online changepoint detection algorithm. Observe that some of the edge details are better preserved with the offline changepoint algorithm.

We run the same experiment as in Fig. 2.4, where an clementine is rotated at different brightness levels. We show the resulting SNR for online vs offline detection in supplementary Fig. 2.9

2.7 Supplementary Section: SNR Analysis

We test our deblurring algorithm for different motion speeds for the case of rotational motion using the "clementine" dataset in the main text. We measure the SNR by computing the discrepancy between the ground truth flux image and the deblurred result. We do this by temporally downsampling the original photon frames, so the number of photons decrease as the motion speeds up. Suppl. Fig. 2.10 shows how changing the regularization parameter λ in the offline flux changepoint detection algorithm effects the SNR. We find that as long as λ is high enough a good reconstruction SNR stays high. In Suppl. Fig. 2.11 we show that our algorithm converges to the performances of a long exposure capture (with motion blur) if the number of photons per degree of rotation falls below 3.

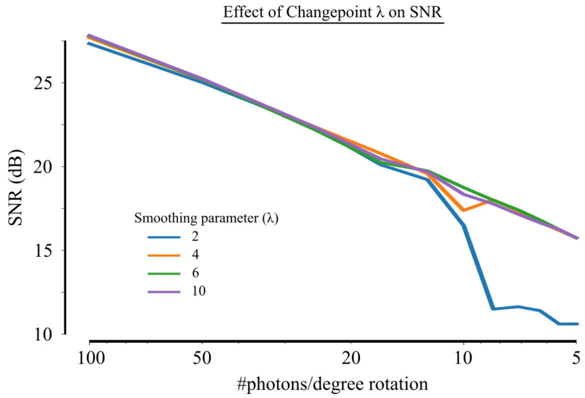


Figure 2.10: **Effect of offline changepoint algorithm regularization parameter λ.** If λ is large enough we get good performance. When λ is too small, many flux changepoints are found, which will cause the CPV to be too noisy to properly align frames.

We run the same experiment as in Fig. 2.4, where an clementine is rotated at different brightness levels. We test our offline QIS changepoint detection method by removing the timing information and only considering a binary output. Our adaptive changepoint method helps at low light levels, see Fig. 2.12.

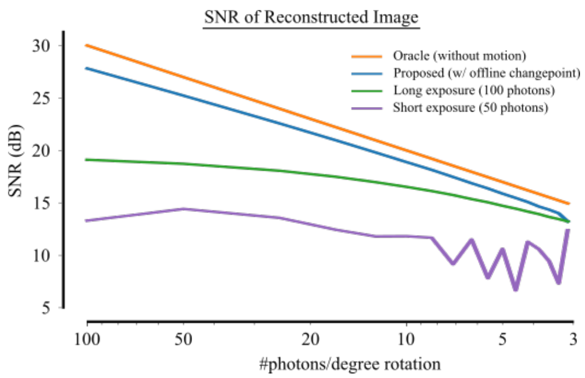


Figure 2.11: **Effect of number of photons captured per degree of rotation.** Our algorithm is unable to find flux changepoints at speeds of 3 photon frames per degree. When no flux changepoints are found we just get a long exposure image. We also see that the SNR for a blurry image (long exposure) or a noisy image (short exposure) is worse than the proposed deblurring method until our method converges to the long exposure image.

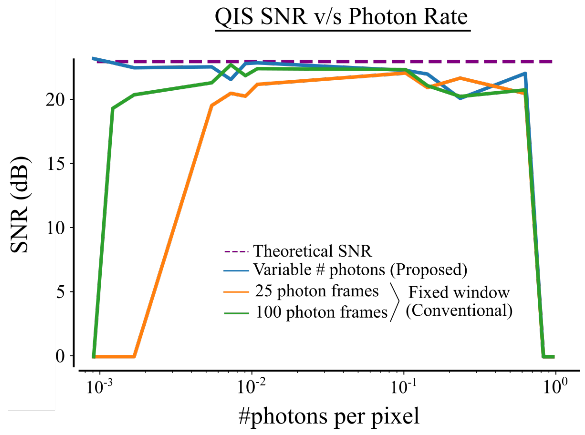


Figure 2.12: **QIS, SNR vs. Brightness.** We run the same experiment as in Fig. 2.4, with the rotating clementine at many light levels. We remove timing information from the raw data to create QIS binary data frames and run the same deblurring experiment. Again our adaptive changepoint method is helpful in low light scenarios. Note that the sharp drop on the right is due to saturation of a QIS sensor.

2.8 Supplementary Section: Additional Simulation Results

This section contains some additional simulated results. A second scene with two toy cars is displayed, we use the same parameters as the toy car scene in the main text for frame generation, changepoint detection and deblurring. For this scene the dark car moves 90 pixels and has a contrast 3.3. The bright car has a contrast of 1.2 and moves 30 pixels. Our results are shown in supplementary Fig. 2.13 and the clustered changepoints are shown in supplementary Fig. 2.14. Again, our method is able to deblur both moving cars.

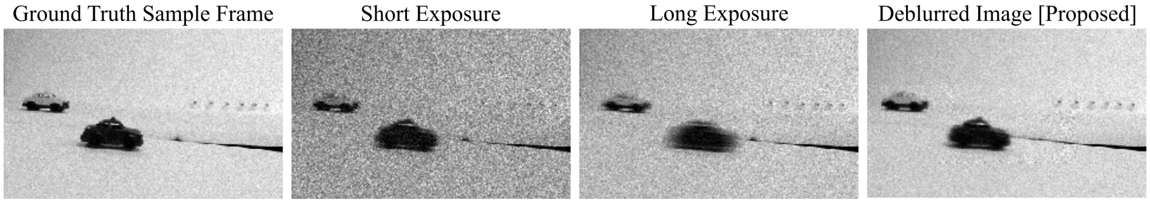


Figure 2.13: **Simulated Multiple Objects.** We simulate SPAD data from a 240 fps phone video of two rolling toy cars, a fast moving dark car and slow moving bright car. From left to right, the ground truth image shows the result of generating the same number of photon frames from the first frame of the video sequence. The short and long exposure images show the results of using only the first 75 and 250 photon frames, respectively. Notice that the short exposure preserves the dark car while the bright car is quite noisy, on the other hand, the long average blurs the dark car but preserves details of the bright one better. Finally, our deblurring algorithm is able to reconstruct both the dark and bright car.

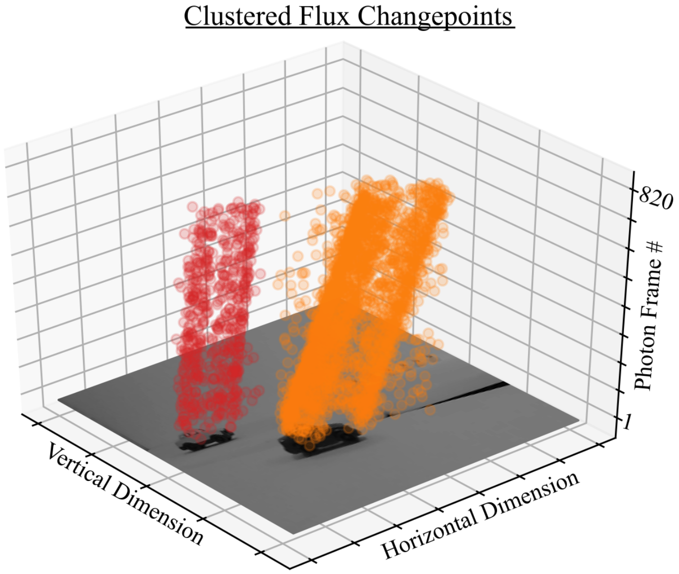


Figure 2.14: **Clustered Flux Changepoints.** Displayed are the 2 flux changepoint clusters found for the scene in Supplementary Fig. 2.13. We only display half of the flux changepoints in each cluster for visualization purposes.

We simulate a simple pixel art scene to demonstrate the advantage of a changepoint video rather than burst frames. Notice in supplementary Fig. 2.15 that the changepoint video frame is able to capture a wide range of motion speeds and contrasts that a single fixed frame cannot capture.

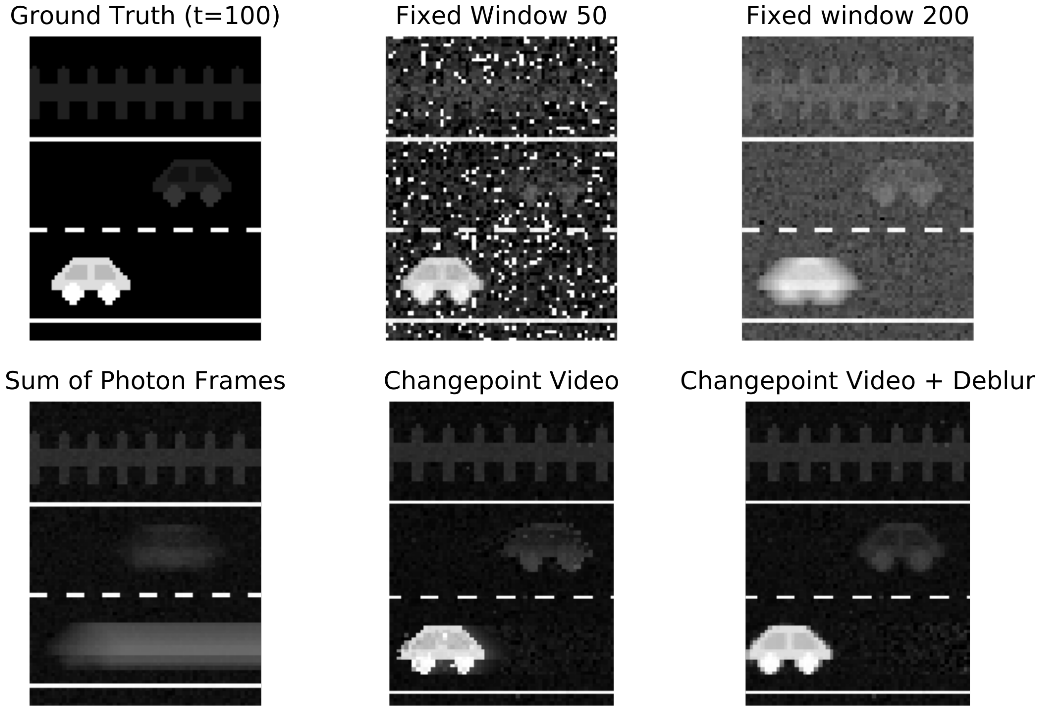


Figure 2.15: **Pixel Art Multiple Objects.** We simulate a scene where a bright car moves quickly to the right and a dim car moves slowly to the left. Notice that in the short averaging window, the dim car is lost in the noise while in the long averaging window the bright car is blurred. The sum of all photon frames maintains the background quite well. The changepoint video frame adapts to motion in each pixel and captures both the bright car, the dim car, and the background. Notice that the changepoint video frame loses some of the structure of the dim car due to noisy changepoints. We combine the adaptive changepoint video with a deblurring algorithm to deblur both cars.

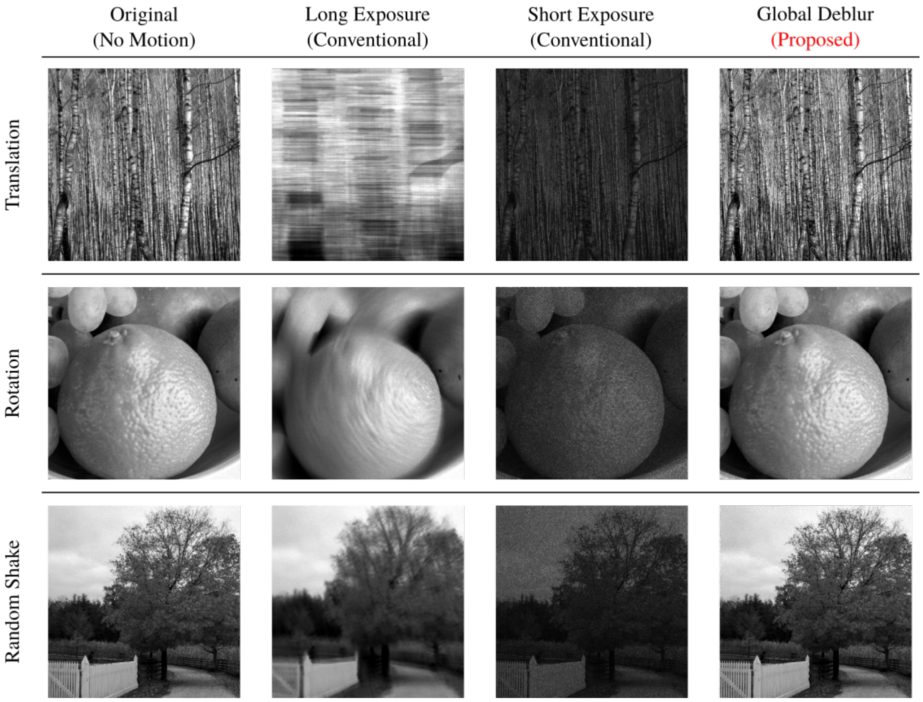


Figure 2.16: **Simulated motion deblurring results for different types of global motion.** From left the right the columns show, a ground truth image that shows the result if the same number of photons are sampled from the original image but with no motion. A long exposure where all photon frames are summed, the result is a blurred image. A short exposure image shows the combination of the first 20 photons of the timestamp data, notice the edges are sharp but noise dominates. The result of the global motion deblur algorithm is shown in the last column. (Original images from FreeImages.com)

Global Motion Results

We start with a ground truth high resolution image, successively apply rigid transformations (using known rotations and translations), and generate photon frames using exponentially distributed arrival times. We reassign these high spatial resolution timestamps a lower resolution 2D array to simulate a low resolution SPAD pixel array.

We model a photon frame readout from a SPAD array with 8000 bins and bin width of 256 ps. Images are scaled so that the true photon flux values ranges

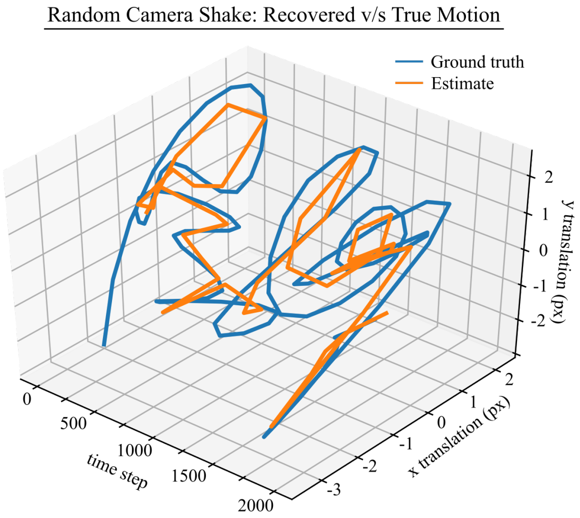


Figure 2.17: **Comparison of estimated and true motion trajectories.** This plot shows the true and estimated motion trajectories for the random shake case in Fig. 2.16. The recovered motion tracks the ground truth motion quite well.

between 10^4 and 10^8 photons per second. We then iteratively transform the flux image according to known motion parameters, and downsample spatially to a resolution of 425×425 before generating photon timestamps.

For the horizontal translation blur, we moved the image 1 pixel to the right blur, we rotate the image 0.1 degree for every 10 generated photons for a total of 1000 photons. To emulate random camera shake, we create random motion trajectories by drawing two i.i.d. discrete uniform random variables between -3 and 3 and use that as the number of pixels to translate along the horizontal and vertical directions. We generate 20 photons per translation for a total of 2000 photons. We use the BottomUp algorithm (Truong et al. (2020)) with $\lambda = 5$ for the changepoint detection step. In practice we found that the results were not very sensitive to the choice of λ and values between 2 and 12 produced similar results.

We generate photon events from an exponential distribution. We transform the flux image, then down-sample to simulate objects with more detail than pixel resolution. We then generate 10-20 photons from the down-sampled flux image. Continuing this we get a 3-d tensor of photons representing global motion of the original image.

Supplementary Fig. 2.16 shows simulated deblurring results for three different motion trajectories. The top row shows a case of horizontal translation: conventional long/short exposures must trade off motion blur and shot noise. Our deblurring method reproduces sharp details, such as vertical lines of the tree stems. The second row shows a case of rotation: note that different pixels of the scene now undergo different amount of motion per unit time. Our method reconstructs fine details of the texture of the clementine peel. The bottom row shows random camera shake with unstructured motion which is much more complex than the first two cases which have constant smooth motion. Our technique is able to correct for this global motion by approximating the overall motion trajectory as a sequence of small translations and rotations. Supplementary Fig. 2.17 shows the comparison between the true motion trajectory and the trajectory estimated as part of our deblurring algorithm.

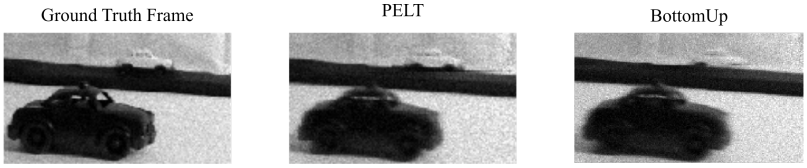


Figure 2.18: **Toy Car Simulated Scene.** Here are the results using the toy car simulated scene. The parameters used are the same as in the main text. Note that the BottomUp algorithm is able to detect and deblur both cars; however, it seems to produce a slightly noisier and blurrier result. For this scene 210 by 300 with 690 photon frames per pixel, on our unoptimized system with 30 parallel cores, it takes the PELT algorithm two minutes to run while the BottomUp takes approximately one minute.

2.9 Supplementary Section: Comparison of BottomUp and PELT

In this section we run some of the same simulation experiments from the main text but with the BottomUp algorithm instead of the PELT algorithm. In Suppl. Fig. 2.18 we compare the ability of both algorithms on the toy car scene, BottomUp seems to produce slightly more noisy and blurry results. In Suppl. Fig. 2.19 we re-run the contrast vs. speed simulations and find that BottomUp does comparably well with slightly more false positives.

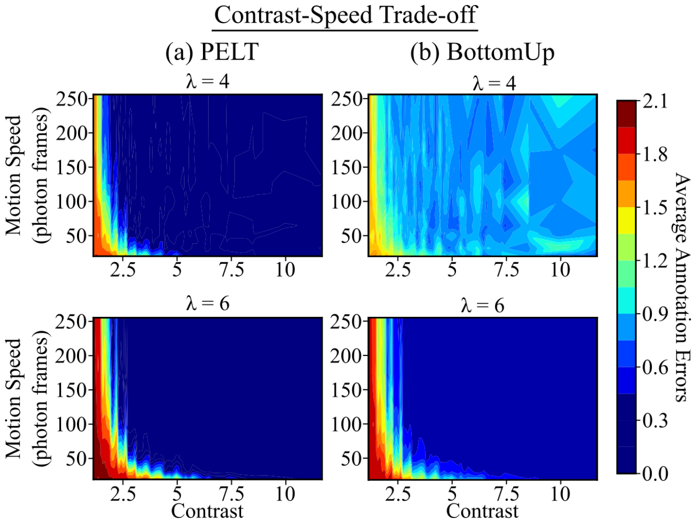
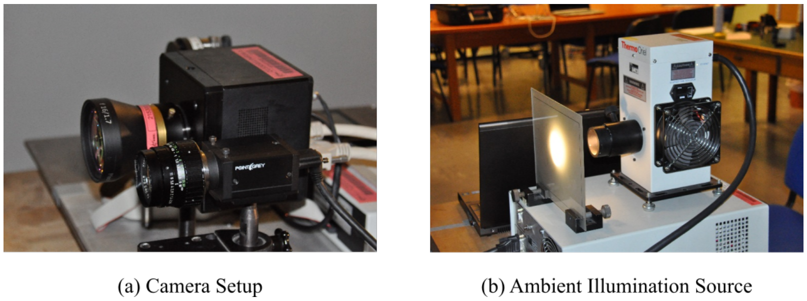


Figure 2.19: **BottomUp vs. PELT - Contrast vs. Speed.** We repeat the contrast vs. speed simulation as done in the main text but with the BottomUp algorithm. Note that the BottomUp algorithm does comparably well for $\lambda = 6$. For $\lambda = 4$, the BottomUp algorithm is able to detect more difficult objects at the expense of false positives during easier scenarios.

2.10 Supplementary Section: Experiment Setup

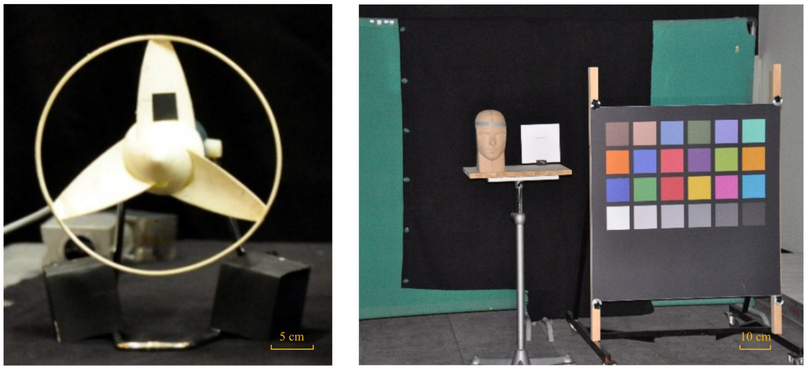
For the experimental investigation we used a 32×32 InGaAs SPAD array from Princeton Lightwave (PL GM-APD 32 x 32 Geiger-Mode Flash 3-D LiDAR Camera) and an RGB camera (VIS/BW point gray Grashopper 3, GS3-U3-23S6M-C) capturing the same field-of-view. The SPAD camera samples photon events with a depth of 10^{13} bins and 250 ps/bin. Further, during the experiments we used a frame readout rate of 50 kHz. The InGaAs sensor is sensitive in near infrared (NIR) to shortwave infrared (SWIR) wavelengths ranging from 900 nm to 1.6 µm.

During the measurements we investigated two different type of scene setups: a "fan" and a "checkerboard" scene, as depicted in Suppl. Fig. 2.20. In the first scene, the fan consists of three blades mounted on a central cone and is enclosed by a circular frame with a diameter of 18 cm. One blade was marked with a black piece of paper. The fan scene was used to investigate rotational motion. The second scene consists of an artificial head, a white plate and a colored checkerboard. Colors appear at a different gray levels in the SWIR wavelength images. This second scene was used to investigate random motion due to a horizontally shaking camera.


(a) Camera Setup

(b) Ambient Illumination Source

Figure 2.20: **Hardware setup.** (a) Our hardware setup consists of a Princeton Lightwave SPAD array (PL GM-APD 32×32 Geiger-Mode Flash 3-D LiDAR Camera) and an RGB camera (VIS/BW point gray Grashopper 3, GS3-U3-23S6M-C) capturing the same field-of-view. (b) Ambient illumination is provided by a diffuse light source (broadband arc lamp ThermoOriel Model 66881).



(a) “Fan” scene

(b) “Checkerboard” scene

Figure 2.21: **Experimental Scenes.** (a) The “fan” scene consists of a small fan with a black square patch on one of the fan blades. (b) The “checkerboard” scene consists of a large color checkerboard and a mannequin head.

2.11 Supplementary Section: Changepoints for Compression

Raw Data: Consider a frame readout SPAD camera which uses b_t bits to time-tag each photon. It reads out a set of frames $V(x, y, t) \in [0, 2^{b_t}]^{H \times W \times T}$, where H, W, T are the height, width and total number of photon frames, respectively. V will take a total of

$$b_V = b_t HWT \tag{2.20}$$

bits of data. Note that a binary SPAD or QIS sensor will have $b_t = 1$ and is an important point of comparison because currently available large single photon cameras are binary.

Block Sums: One compression of V uses block sums,

$$V(x, y, t)_S = \sum_{l=\lfloor \frac{t}{S} \rfloor}^{\lfloor \frac{t}{S} \rfloor + S} V(x, y, l) \tag{2.21}$$

which simply adds together every S frames reducing the total data by a factor of S. V_S will use

$$b_{V_S} = b_t HW \frac{T}{S} \tag{2.22}$$

bits of data. While this compresses data it also losses temporal resolution and does not adapt to the scene.

Changepoints: In a changepoint representation of V, changepoints of the form $e = \{x, y, t, \phi\}$ are read out. Each changepoint tells us that at pixel x, y at time t, the photon flux was ϕ before t. The number of bits required for for N_e changepoints is,

$$b_e = N_e(\log_2(WHT) + b_e) \tag{2.23}$$

assuming H, W, T are powers of two, and where $b_e \leqslant \log_2(T)$ is the required bit depth for representing photon flux if we only care about temporal resolution at the photon frame rate of our camera. N_e is a function of the scene and is the difficult

part for analyzing changepoint compression rates; for example, in a stationary scene each pixel only requires 1 changepoint to represent the entire scene whereas fast moving scenes may require a changepoint per pixel every few frames. Later, we analyze the behavior of N_e for simple scenes.
Overflow changepoints: If T is large it may be better to encode changepoints based on the time difference between them. In this case we will use a maximum of T_o bits to encode time. For every pixel after 2^{T_o} frames an overflow marker can be sent indicating a rollover in the timing circuit. However, we can do better by sending only a single overflow marker for the whole array, that will also simultaneously reset all internal clocks for each pixel. The bits used in this modality is

$$b_o = (\frac{T}{2^{T_o}} + N_e)(\log_2(WH) + b_e + T_o). \tag{2.24}$$

Analyzing N_e - Global Motion

Lets consider a scene with global motion at constant velocity v $(\frac{px}{frame})$, without loss of generality assume v is in the positive x direction. Then our scene can simply be written as, $I(x,y,t) = I(x+vt,y,0)$. We assume $I(x,y,0)$ is piece-wise constant at a moderate scale.

Fix a y, let N_y be the number of constant pieces in row y, and $N = \sum N_y$. At $t = 0$, let the i^{th} piece start at x_i and have length p_i with intensity I_i, then at a fixed y,

$$I(x,y=y,0) = \sum_{i=0}^{N_y} I_i \delta(x_i \leqslant x < x_i + p_i) \tag{2.25}$$

Let us further assume our camera pixels focused at a small point so we approximately sample along the integers of $I(x,y)$, and also assume the pieces are moderately sized so $p_i \geqslant 1$. Let us now find N_e for this problem.

The i^{th} piecewise component will require v changepoints per photon frame because the leading edge will create a changepoint at rate v but the lagging edge's changepoint will be the same as an changepoint from the leading edge of a different component so we only need to count the leading edge changepoint for each piece. Therefore in this case, $N_e = NTv$. It is reasonable to assume $N < WH$ and $v < 1$, because otherwise our spatial or temporal sampling is not high enough for our scene. Then our changepoints compression ratio is,

$$\frac{b_V}{b_e} = \frac{b_t HWT}{N_e(\log_2(WHT) + b_e)} \tag{2.26}$$

$$= \frac{b_t HW}{Nv(\log_2(WHT) + b_e)}. \tag{2.27}$$

As expected there is more compression for fewer pieces moving at a slower velocity.

Let the smallest object in the scene be p pixels long, then $N \leqslant \frac{HW}{p}$ and $N_e \leqslant \frac{HWTv}{p}$. Then the compression achieved by the changepoints representation is bounded by,

$$\frac{b_V}{b_e} \geqslant \frac{b_t p}{v(\log_2(WHT) + b_e)}. \tag{2.28}$$

Larger objects and slower objects are more easily compressed by changepoints where the compression rate is quite large.

Analyzing N_e - Local Motion

Single Convex Object

Lets consider a scene with a single moving convex object moving that moves d pixels, with area A and circumference c. Then the maximum number of changepoints required is bounded by $N_e \leqslant 2dc$, which is achieved for a line of length c moving at constant velocity $\frac{d}{T}$ perpendicular to its extent.

In general, given a convex object's binary mask, $s(x, y)$, centered at $(0, 0)$, the bound for the number of changepoints required is proportional to the largest 1 dimensional projection of the object and is achieved when the object is moving in the direction parallel to the maximal projection direction. Let l be the maximum projection length then the number of changepoints required for this object is bounded by,

$$N_e \leqslant 2dl. \tag{2.29}$$

Multiple Convex Objects

We can bound the scenario of multiple objects by using the previous result along with the assumption that objects never occlude each other.

In particular if we have K objects each with maximum projection length l_i, and each moving a total distance d_i,

$$N_e \leqslant \sum_{i=1}^{K} d_i l_i. \tag{2.30}$$

3 OFDVDNET: A SENSOR FUSION APPROACH FOR VIDEO DENOISING IN FLUORESCENCE GUIDED SURGERY

In medical imaging modalities such as fluorescence image-guided surgery (FGS), thermal imaging, imaging of Cerenkov radiation in radiotherapy, or Raman scattering, there are many photon-starved or low contrast signals that are important to image at video frame rates. For example in FGS, a patient is injected with a fluorescent compound that binds to a specific tissue type. Then the tissue is illuminated with an excitation light source during surgery and the fluorescent marker emits a weak fluorescent signal that can be picked up by a fluorescent camera. Existing procedures use fluorescent markers to visualize tumors (Ishizawa et al. (2009); Zhang et al. (2017)), blood vessels (Li et al. (2010)), lymph nodes (Kitai et al. (2005); Frumovitz et al. (2018)), necrotic tissue (Xie et al. (2015); Zajac et al. (2022)), or nerves (Gibbs-Strauss et al. (2011)). A surgeon will use the fluorescent camera feed to make real-time intraoperative decisions, where having high fluorescent signal is key. Fluorescence is much weaker than standard light sources and this low signal is compounded by the need for short exposure times for real-time video. Additionally, some useful fluorescent dyes and intrinsic fluorescence compounds often have low photon yields or low specificity leading to noisy videos that may lack the required sensitivity for clinical decision-making. For example, the auto-fluorescence from the inferior parathyroid can be weak during thyroidectomy (Kim et al. (2016)) making it hard to detect the target. The low signal levels in FGS can be improved computationally through video denoising methods that will be able to take into account past captured frames to denoise a current frame as well as temporal and spatial information. Denoising methods will increase signal-to-noise levels in FGS to allow surgeons to lower the injection dose, allow for longer imaging time periods, and allow use of lower contrast and low sensitivity fluorescent agents even

This chapter is based on work done for the following publication (Seets et al. (2024b)): Trevor Seets*, Wei Lin*, Yizhou Lu, Christie Lin, Adam Uselmann, and Andreas Velten. "OFDVDnet: A Sensor Fusion Approach for Video Denoising in Fluorescence-Guided Surgery." In Medical Imaging with Deep Learning. 2023. *co-first authors

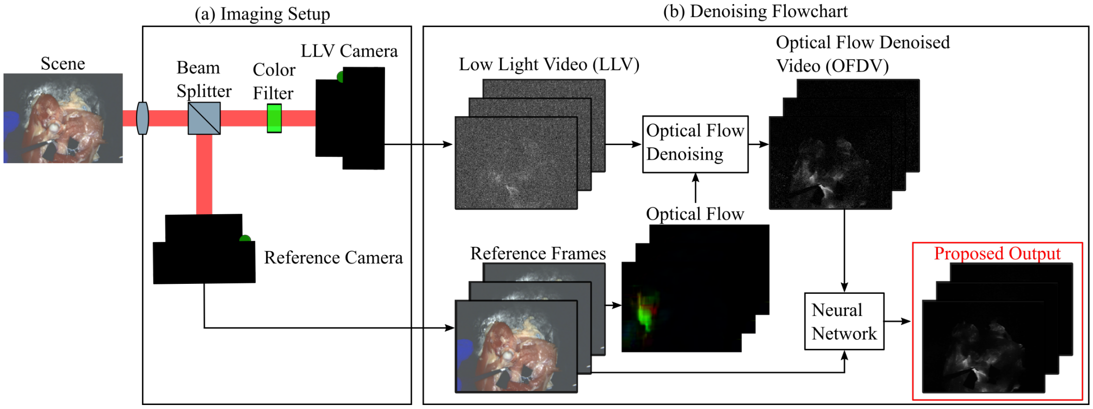


Figure 3.1: **Overview:** (a) Our imaging setup consists of two co-registered cameras: the reference camera measures a conventional intensity image of the scene while the low light video (LLV) camera measures a noisy fluorescence video. (b) To denoise the LLV, we first find optical flow throughout the video using the reference frames. We then apply the optical flow to the LLV to align frames in time, we merge the aligned frames to create the OFDV. The OFDV and reference frames are then fed into a video denoising neural network that will produce our final denoised video.

auto-florescence.

Due to the low brightness of fluorescent markers, fluorescence signals are flooded by background and ambient illumination. Hardware solutions have been developed to increase signal by filtering out ambient light. One method, called transient lighting (Velten et al. (2020)), switches on and off the room light in short intervals between the exposure of fluorescence camera frames. The blinking light is unnoticeable to the human eye due to the flicker-fusion threshold making the surgical room appear well-lit while providing a dark environment for fluorescent capture. Another method is wavelength filtering which blocks photons from the illumination light source and photons outside of the fluorescence emission band. Both wavelength filtering and transient lighting lend themselves to a two-camera approach where one camera captures the low light video (LLV) of fluorescence, and another co-located camera captures a good quality RGB video called the reference video (RV).

In this work, we aim to recover a denoised video from a noisy low light video

(LLV) of a faint fluorescent compound with noise standard deviations one magnitude higher than what current methods consider. To address this extreme noise, we propose a sensor fusion approach where the RV provides motion and structural cues that act as a guide when denoising the LLV. Due to the low signal level of the fluorescent compound we consider using a shot noise-limited camera such as a single photon avalanche diode (SPAD) to capture the LLV.

We identify 3 useful properties for FGS denoising, spatial, temporal, and RV correlation properties, many existing algorithms were only designed to use 2 of these factors. Video denoising methods, such as the non-learning based V-BM4D (Maggioni et al. (2012)) and learning based FastDVDnet (Tassano et al. (2020)), DVDNet (Tassano et al. (2019)), and VNLNet (Davy et al. (2019)) are meant to denoise videos without a RV so use only temporal and spatial properties. These algorithms rely on explicit or implicit feature mapping to properly denoise frames, which is difficult with the extreme noise considered in this paper. Additionally, convolutional neural network (CNN) approaches do not scale well to take as input a large number of video frames due to GPU memory limitations. A second class of denoisers are guided image denoisers such as guided filtering and joint bilateral filtering that were designed for images with a guide image so make use of spatial properties and RV spatial correlations but no temporal information. A third category of denoisers are align and merge techniques (Hasinoff et al. (2016)) which are often used to denoise videos generated by single photon cameras (Seets et al. (2021); Ma et al. (2020); Istvan et al. (2015); Gyöngy et al. (2017); Gyongy et al. (2018)) and exploit temporal correlation; however, at extreme noise levels, alignment is difficult. To utilize all three properties, we propose OFDVDnet which draws inspiration from both align and merge techniques and deep learning methods in a two-stage approach. First, we align the LLV with motion extracted from the clean RV to compress the long temporal correlations in the LLV followed by a video denoising CNN based on FastDVDnet.

An overview of OFDVDnet is shown in Fig 3.1, first, our LLV and RV data is collected using two co-registered cameras. We use the RV to compute optical flow and an occlusion mask between successive frames (Fig 3.2). The optical flow and

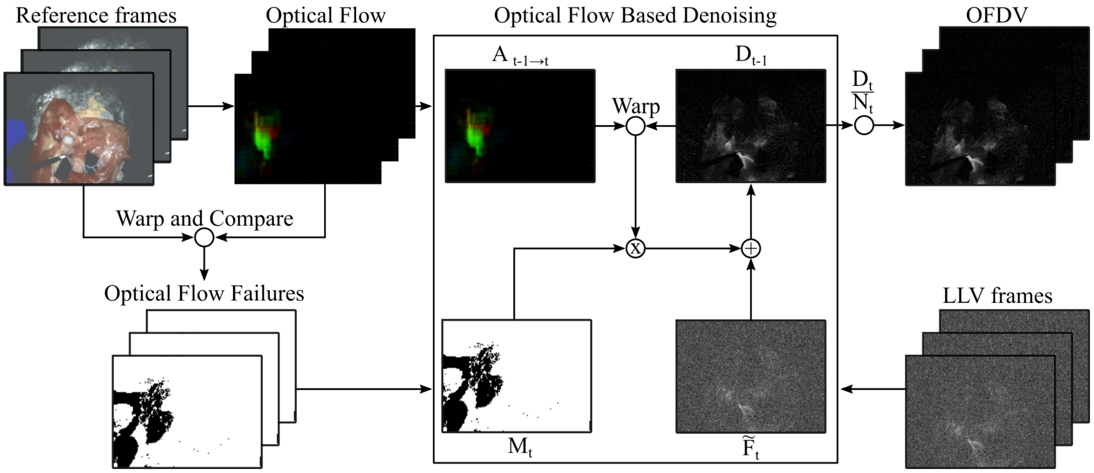


Figure 3.2: **Optical Flow Denoising:** First, we calculate optical flow from the RV frames. We then calculate an optical flow failure mask by warping the RV frames forward in time and comparing the warped frames to the true RV frame at that time. We then iterate over every frame and create a running sum of the LLV video. In one iteration we warp the previous running sum result forward in time 1 frame with the optical flow. We then reset pixels in the running sum according to the optical flow failure mask. The current LLV frame is then added to the result which becomes the next running sum frame and finally the OFDV.

mask are then applied to the LLV to create a denoised motion-compensated estimate of the LLV, called the optical flow denoised video (OFDV). The OFDV incorporates information from many distant frames. One main advantage of the OFDV is it is a compressed representation of many different frames that can be easily loaded into GPU memory. Finally, we feed the ODFV and the RV frames into a video denoising neural network inspired by FastDVDnet to create a final denoised output.

We capture a RV and LLV aligned dataset using a commercial FGS imaging system (OnLume Surgical, Madison, WI) in a simulated surgery based on the blue blood chicken model used in micro-surgical training (Albano et al. (2021)) where we inject a fluorescent agent to highlight blood vasculature in chicken thighs. To collect training data we use indocyanine green (ICG) as our fluorescent agent to capture low-noise video and simulate much weaker fluorescence giving us noisy and ground truth pairs for training.

3.1 Method

Dataset

We capture data with a transient lighting-enabled (Velten et al. (2020)), clinical wide-field FGS imaging system (OnLume Surgical, Madison, WI). OnLume's system uses two camera sensors for both the reference and fluorescent cameras. In a surgical training model (Albano et al. (2021)), we inject the near-infrared fluorescent agent, indocyanine green (ICG), via syringe into the femoral artery of four chicken thighs to simulate vascular surgery. We prepared varying doses of ICG up to the clinical guidelines of 2.5 mg/mL to generate fluorescent videos with visual contrast and low noise that can be treated as ground truth and simulate much lower fluorescence. In future work, we would like to detect markers that have much fewer photons than our captured fluorescent videos. We capture about 100 minutes of simulated surgical footage with a variety of motion such as cutting, pulling, squeezing, injecting, and working with surgical tools. The 100 minutes of footage is broken up into 590 100-frame long videos. The videos are captured at 15 frames per second at a resolution of 768×1024; however, we downsample the resolution to 192×256 to speed up training times in our experiments.

To simulate our noisy LLV, we scale the fluorescent frames between $[\phi_{\text{back}}, \phi_{\text{sig}} + \phi_{\text{back}}]$ photons per frame, where ϕ_{back} is the number of background photons and ϕ_{sig} is the maximum number of signal photons in a pixel, and then we apply Poisson noise to the scaled frames. We use $\phi_{\text{back}} = 10$ photons accounting for ambient light and sensor dark counts. We use 3 different signal photon levels, $\phi_{\text{sig}} = [1, 5, 20]$, to represent a range of fluorescent strengths. We measure the noise level with the signal-to-background (SBR) ratio, SBR $= \frac{\phi_{\text{sig}}}{\phi_{\text{back}}}$. Giving us three SBR levels, SBR$= [0.1, 0.5, 2]$ or standard deviations of $\sigma = [826, 180, 57]$ on an 8-bit image, see Appendix 3.8.

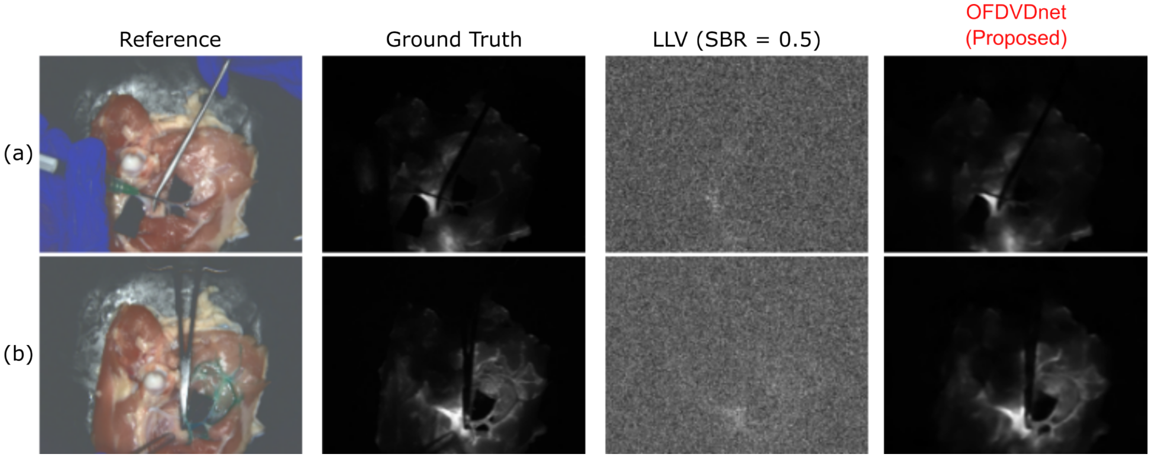


Figure 3.3: **Injection of ICG:** In this figure, we show the RV frame, ground truth fluorescence frame, noisy LLV frame, and our proposed denoised result of a scene before and after ICG injection. (a) At the start of ICG injection, there is a small amount of innate fluorescence near the injection site. (b) After injection, the vascular structure becomes visible in the fluorescence channel. OFDVDnet is able to reconstruct both scenes well, maintaining most of the important vascular structure in (b). Note that at high concentrations the dye appears green, but this color does not necessarily indicate the near infrared fluorescence from the ICG for example see the tip of the syringe in (a).

Denoising Algorithm

Our goal is to denoise the LLV with the help of a co-registered noise-free RV. For our noise model, we consider the case where the LLV was captured with a photon-limited camera such as a SPAD and is dominated by Poisson noise (Hasinoff (2014)). While a pure Poisson noise model is a good approximation for single photon cameras, commercial CMOS cameras may introduce additional noise terms but next-generation CMOS cameras (e.g. Hamamatsu qCMOS) have low enough noise to measure single photons. These new cameras are shot noise limited and generally have a Poisson dark current or dark count rate which can be modeled with a constant photon background rate. We anticipate next-generation cameras to be used in FGS, so we use a Poisson noise model with a constant background to simulate these cameras. We note that a different camera-specific noise model

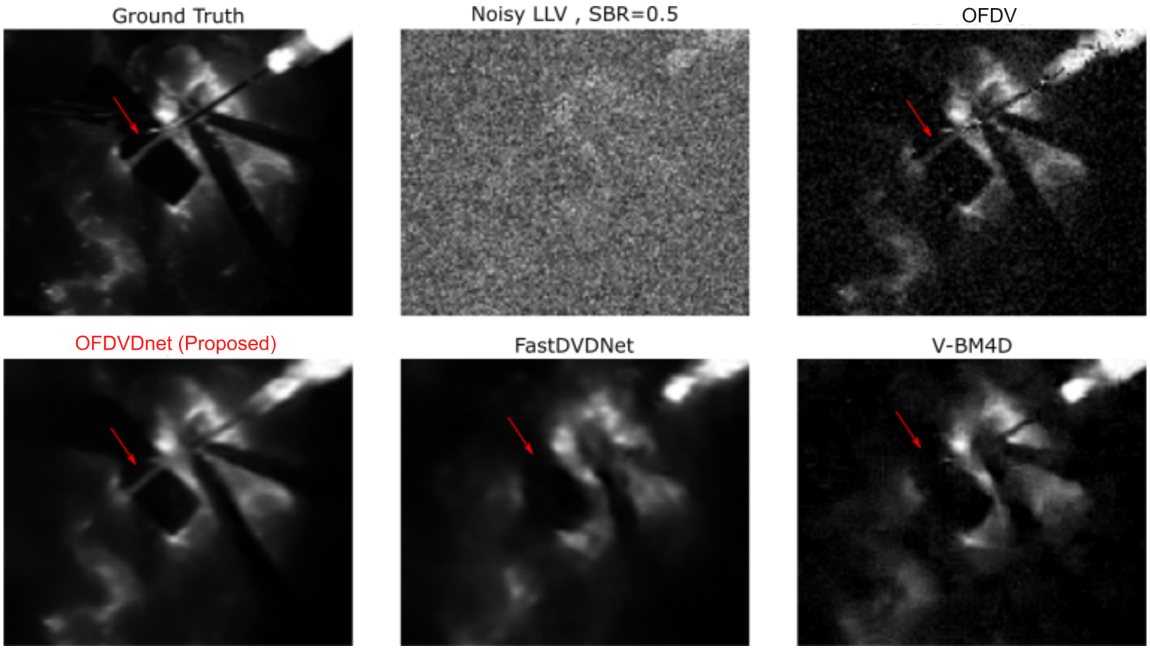


Figure 3.4: **Vessel Detail:** OFDVDnet correctly reconstructs a small vessel (red arrow) while comparison methods remove it entirely.

would be straightforward to implement with our model. At time t, let $\tilde{\mathsf{F}}_\mathsf{t}$ and W_t denote the captured LLV frame and RV frame, respectively. Let, $\tilde{\mathsf{F}}_\mathsf{t} = \mathsf{Poiss}(\mathsf{F}_\mathsf{t})$ where F_t represents the ground truth LLV frame, and Poiss represents Poisson sampling. Because we consider very noisy cases it is difficult to denoise $\tilde{\mathsf{F}}_\mathsf{t}$ alone, so we use the RV as a guide.

We use a two-step denoising method, first our optical flow based denoising exploits the alignment of the LLV and RV to find motion within the scene from the RV and apply it to the LLV. Once the LLV is aligned, frames can be merged to reduce noise to create the optical flow denoised video (OFDV). We warp the LLV both forward and backward in time aggregating information from all frames. However, if a video is needed to be displayed with little latency, then the OFDV could be warped only forward in time resulting in lower latency at the cost of quality. In our second step, a CNN takes the OFDV and RV frames as input to further denoise.

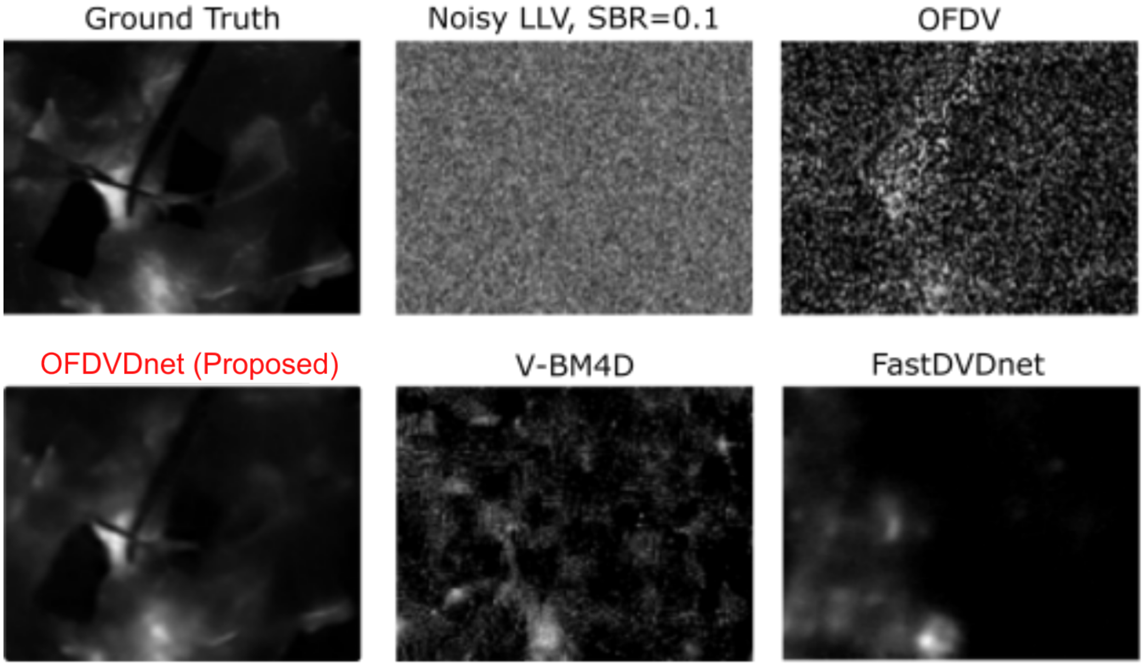


Figure 3.5: **High Noise:** This scene has no occlusions so OFDV averages over all frames giving good results where comparisons fail.

This two-step approach uses frames from distant points in time to denoise a single frame without increasing GPU memory use.

Optical Flow Based Denoising: One technique to denoise $\tilde{F}_t$ is to average 2T frames around time t, $\tilde{F}_{t-T}...\tilde{F}_{t+T}$. Higher T will reduce noise but also increase motion blur. Therefore, we would like to spatially align the frames together before averaging. We use the RV to find optical flow and flow failure masks, which are then applied to the LLV to create the OFDV. The method is shown in Fig 3.2.

First, we use the RV frames to find optical flow warp maps, $A_{t \to t+1}$ that warp a frame at time t to time $t + 1$. $A_{t \to t+1}$ can be found using any optical flow method. We found that Gunnar-Farneback's (Farnebäck (2003)) optical flow algorithm was sufficient for our problem and, on our hardware, ran faster than the CNN based MaskFlowNet (Zhao et al. (2020)). We use $A_{t \to t+1}$ to average the LLV frames along motion trajectories with a running sum strategy. Let D_t^+ be the running sum at time t, then,

$$D^+_t = A_{t-1\to t}(D^+_{t-1}) + \tilde{F}_t \tag{3.1}$$

Ideally, we could estimate a frame at time t as $\frac{D^+_t}{t}$. However, optical flow can fail in a variety of circumstances such as occlusions. To deal with flow failures, for each frame we detect pixels with flow errors and reset the summation for those pixels. Then due to the resets each pixel in D^+_t may represent a sum over a different period of time, so we also record the length of time since the last reset in each pixel given by N^+_t. Therefore, the new LLV estimate at time t is given by $\frac{D^+_t}{N^+_t}$, which represents an average in each pixel since the last optical flow failure. Let M_t be a binary mask which has a pixel value of 0 if $A_{t-1\to t}$ fails to warp correctly, then Eq. 3.1 becomes,

$$D^+_t = M_t \odot A_{t-1\to t}(D^+_{t-1}) + \tilde{F}_t \tag{3.2}$$

$$N^+_t = M_t \odot A_{t-1\to t}(N^+_{t-1}) + \mathbb{1} \tag{3.3}$$

where $\odot$ represents pixel-wise multiplication and $\mathbb{1}$ represents an image of all ones. We generate M_t by detecting optical flow failures by comparing intensity values of successive reference frames; we warp W_{t-1} and compare it to W_t by,

$$M_t = \left| 1 - \frac{A_{t-1\to t}(W_{t-1})}{W_t} \right| < \tau \tag{3.4}$$

where τ is a threshold value, we use $\tau = 0.07$. We then compute, $\widehat{F}^{flow+}_t$,

$$\widehat{F}^{flow+}_t = \frac{D^+_t}{N^+_t} \tag{3.5}$$

$\widehat{F}^{flow+}_t$ is the forward OFDV, we combine $\widehat{F}^{flow+}_t$ with the backward OFDV $\widehat{F}^{flow-}_t$ to create out final OFDV. The backward OFDV is calculated by running the same process reversed in time. Similar to the forward OFDV, we calculate D^-_t, and N^-_t that represent the running sums on a time-reversed video. We can then create the final OFDV, $\widehat{F}^{flow}_t$, by combining the forward and backward estimations as follows,

$$D_t = D_t^+ + D_t^- - \tilde{F}_t \tag{3.6}$$

$$N_t = N_t^+ + N_t^- - \mathbb{1} \tag{3.7}$$

$$\widehat{F}_t^{flow} = \frac{D_t}{N_t} \tag{3.8}$$

where each pixel in D_t and N_t represents the sum and number of aligned pixels between two optical flow failures, respectively. Note that to find D_t and N_t we need to subtract out what is contained in both forward and backward OFDVs. A pixel value in the OFDV is the average value of an aligned LLV between optical flow failures. For example, if optical flow is correctly found for all frames, the OFDV will average along motion trajectories over the entire video, lowering noise and avoiding motion blur. Whereas if there is an occlusion the OFDV will avoid motion blur and only average pixel values between successive occlusion events. This process leads to the OFDV having high temporal consistency from frames being correlated and spatially varying noise levels from different averaging lengths per pixel.

Neural Network: In order to further remove the remaining noise and any warping artifacts in $\widehat{F}_t^{flow}$, we use a CNN denoiser based on FastDVDnet (Tassano et al. (2020)) which takes five noisy frames to denoise the middle frame. We provide the CNN with five consecutive OFDV frames $\widehat{F}_{[t-2:t+2]}^{flow}$, averaging time maps $N_{[t-2:t+2]}$, and RV frames $W_{[t-2:t+2]}$ as input. We train the CNN to reconstruct the middle ground truth frame F_t using 1000 training pairs over 100 videos with mean square error (MSE), see Sections 3.6, and 3.7 for details.

The four neighboring OFDV frames of $\widehat{F}_t^{flow}$, $\widehat{F}_{\{t-2,t-1,t+1,t+2\}}^{flow}$, provide the CNN with additional information on the center frame and reduce flickering. The averaging time maps $N_{[t-2:t+2]}$ act as noise maps to indicate per-pixel noise levels because OFDV pixels have averaged signal over a varying number of frames and thus different noise characteristics. RV frames $W_{[t-2:t+2]}$ let the CNN exploit structural similarities between the RV and OFDV.

Table 3.1: Comparison of PSNR/SSIM/FSIM (higher is better) for LLV frames denoised using OFDVDnet, OFDV, and the comparison methods at 3 noise levels.

SBR	OFDVDnet	OFDV	FastDVDnet
0.1	**29.3**/**.76**/**.88**	10.8/.015/.20	24.3/.48/.83
0.5	**34.0**/**.89**/**.93**	21.5/.22/.52	30.8/.80/.88
2.0	**36.9**/**.92**/**.95**	30.8/.72/.82	35.7/.89/.93

SBR	V-BM4D	Guided Filtering	Joint Bilateral
0.1	19.7/.19/.52	16.4/.19/.69	15.8/.11/.59
0.5	29.9/.61/.86	28.1/.61/.86	26.3/.52/.81
2.0	36.7/.88/.92	33.7/.90/.92	31.5/.85/.90

3.2 Results

Our testing set consists of the middle 96 frames from 100 videos. We compare our results to video denoisers only given the LLV frames, the CNN FastDVDnet (Tassano et al. (2020)) re-trained on our data, and V-BM4D (Maggioni et al. (2012)), a popular block matching and filtering technique. We also compare our technique to two image denoising techniques that use a RV frame to assist in denoising a LLV frame, guided filtering (He et al. (2012)), and joint bilateral filtering (Gastal and Oliveira (2011)). In Fig 3.3, we show an example scene before and after ICG injection into the vessels in a chicken thigh. Before injecting ICG, Fig 3.3(a), fluorescence is only seen near the injection site and in areas of the chicken thigh that are either auto-fluorescent or fluorescent due to the chicken treatment process. As the dye injects into the femoral artery, it perfuses and smaller vascular branches begin fluorescing. OFDVDnet is able to reconstruct most of the details of the vascular system with only slight blurring.

OFDVDnet performs well in high noise when the OFDV can average over a large number of frames allowing use of information in the entire video to create the denoised output. Fig. 3.5 shows an example scene (SBR= 0.1) with no occlusion events and only small motion that can be easily taken care of by the OFDV. OFDVDnet reconstructs most of the fluorescent structure correctly, but struggles to reconstruct the moving syringe fully due to large movement. Both FastDVDnet and V-BM4D fail in this high-noise example.

Table 3.2: Ablation study PSNRs.

SBR=0.5	PSNR
OFDVDnet	34.04
No RV Frames	33.82
No Averaging Time Maps	33.30
Only RV Frames	24.13
Switch OFDV with 5 LLV	31.05
Switch OFDV with 17 LLV	32.74

We found that OFDVDnet could better reconstruct small vessels, one example is shown in Fig. 3.4. In this example, the OFDV contains the blood vessel that has been injected with the marker (red arrow) so the denoising network has the information needed to properly reconstruct this important feature. Whereas the vessel is lost in the noisy LLV frames so the comparison methods completely remove this vessel while showing a reasonable image. This failure mode is arguably worse than the un-processed noisy image since it suggests to the surgeon that the image is accurate but in fact conceals the vein.

By leveraging the correlation between successive OFDV frames, OFDVDnet produced videos with very little flickering. When our network denoises successive frames in a video the inputs are very similar, so the output will have little room to flicker due to noise fluctuations. Video results for OFDVDnet and the comparison methods can be found in Appendix 3.5.

For our image quality metrics (IQM) we use peak signal-to-noise ratio (PSNR), as well as structural similarity (SSIM) (Wang et al. (2004a)), and feature similarity (FSIM) (Zhang et al. (2011)) which better correspond to human interpretation than PSNR. The IQA results are summarized in Table 3.1. OFDVDnet outperforms at all noise levels tested and OFDVDnet's IQMs drop slower with increasing noise when compared to other methods. **Network Ablation Study:** We study the effects of the reference frames, averaging time maps, and OFDV on the performance of our network. We retrain the network for each ablation case at SBR= 0.5. The results are summarized in Table 3.2. First, we tested removing the averaging time maps or the RV frame inputs which both decrease the PSNR of the result. Next, we test using only the RV frames as input which obtains a PSNR of 24.13 indicating strong structural priors, such as visible veins, in the RV.

Finally, we replaced the OFDV input with the corresponding LLV frames which resulted in worse PSNR and significant flickering artifacts. We further studied the case of increasing the number of input LLV frames to 17, which was the maximum allowable due to GPU memory constraints. 17 LLV frames produced better quality results compared to five LLV. However, 17 LLV frames resulted in flickering artifacts and a decrease in PSNR compared to using five OFDV frames while also requiring a substantial increase in the required GPU memory (3x), training time (18x), and evaluation time (4x).

3.3 Conclusions and Discussion

In this work, we demonstrated a guided video denoising method meant for applications in FGS that is able to leverage deep learning in a memory-efficient manner with an explicit align and merge step. We captured and evaluated our method on a new dual-camera FGS dataset. OFDVDnet makes use of three properties spatial, temporal, and RV correlations to the LLV while comparison methods only use two of the three. FastDVDnet (Tassano et al. (2020)) and V-BM4D (Maggioni et al. (2012)) use spatial and temporal properties to provide decent results without the need for the RV, but fall off quickly as the noise level increases showing the importance of the RV at high noise. Guided filtering (He et al. (2012)), and joint bilateral filtering (Gastal and Oliveira (2011)) make use of the RV, but only spatially, and use no temporal information. These methods have the worst image quality metrics and result in flickering in the final video at high noise levels but are computationally the least expensive. Finally, our intermediate OFDV only makes use of temporal information from the RV and its results show the importance of using spatial information to denoise.

While OFDVDnet is able to produce strong results, we identify three key areas that future work can improve upon. First, OFDVDnet relies on the time-consuming computation of optical flow between the RV frames which disallows real-time use. An efficient patch-based approach (Hasinoff et al. (2016)) may be faster using the RV frames as a guide. Second, OFDVDnet struggles when strong motion or occlusion disrupts the averaging of the OFDV leading to higher noise results. Better motion tracking that deals with these cases will be useful in denoising more challenging scenarios. One possible place to improve is in the detection of optical flow failures (Eq. 3.1); for example, the current method relies on relative intensity which will falsely detect a failure under changing lighting conditions this could be improved by using a different failure detection method such as normalized cross-correlation. Finally, our dataset is limited to simulated chicken thigh data, so it is unclear how learned priors will translate to other applications such as oncology. While our dataset is useful for evaluation of new algorithms, new application-specific datasets will be required for learning-based approaches. We hope our dataset and method can be used in further algorithm development for medical imaging applications that require higher signal under scene motion.

Acknowledgments This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1747503 and CAREER Grant No. 1846884. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Support was also provided by the Graduate School and the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin-Madison with funding from the Wisconsin Alumni Research Foundation. This project was also supported in part by grant 1UL1TR002373 to UW ICTR from NIH/NCATS

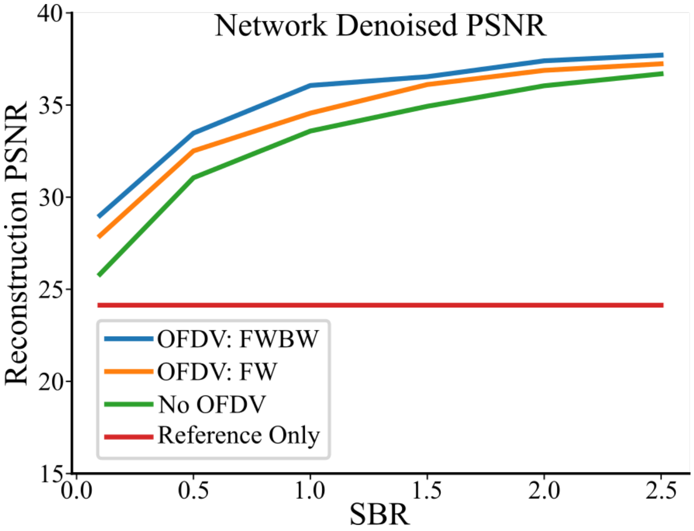


Figure 3.6: **Reconstruction PSNR at different SBR:** This plot shows our neural network reconstruction PSNR for a variety of SBR levels and different inputs into the network.

3.4 Supplementary Section: Additional Ablation Studies

Warp input frames: Explicit motion compensation could benefit video denoising quality (Tassano et al. (2019)). We implemented explicit motion compensation on the input data to our CNN by warping four of the five frames in $\widehat{\mathsf{F}}^{flow}_{[t-2:t+2]}$, $\mathsf{N}_{[t-2:t+2]}$, $W_{[t-2:t+2]}$ to align with their respective middle frames: $\widehat{\mathsf{F}}^{flow}_{t}$, N_t, and W_t using optical flow calculated from the reference frames. Warping did not improve video denoising quality for our case. This results in a PSNR of 33.92.

OFDV forward vs. forward backward OFDV forward(OFDV:FW) warps video frames forward in time, utilizing only information before the current frame in time space. OFDV:FWBW warps video frames both forward and backward in time, utilizing information from both before and after the current frame in time space.We compare the OFDV:FWBW to OFDV:FW by changing the inputs into the neural network at a variety of SBR levels. The results are summarized in Fig. 3.6 along with PSNR curves for the only reference frames and replacing the OFDV with the

LLV. We find the OFDV:FWBW obtains the best results due to its ability to use many frames from both the future and the past. Using the OFDV:FW achieves strong PSNR; however, more flickering is observed in the denoised results when compared to the OFDV:FWBW. This is likely due to stronger input correlations and lower noise in the OFDV:FWBW.

OFDV Ablation Next, we examine how the tunable parameters in our OFDV construction effect the performance of the OFDV construction. Because training the network from scratch takes about 1 week on our hardware we instead choose to run a study on the OFDV PSNR performance. Our OFDV relies on the computation of optical flow based on OpenCV's implementation (Bradski (2000)) Gunnar-Farneback's algorithm (Farnebäck (2003)) which has 6 tunable parameters. We also have a seventh tunable parameter in our detection of optical flow failures, τ, that controls the sensitivity of our optical flow failure detection where higher values lead to fewer failure detection events. We compute the PSNR of the OFDV on one-third of the training set for a range of values of all 7 of these parameters. For each parameter, we calculate the PSNR for a specific value as the maximum PSNR achieved with respect to the other 6 parameters. We found that the 6 optical flow parameters had very little impact on the final OFDV performance changing by less than 0.25dB over the range of tested values (range chosen based on suggestions from OpenCV's documentation). We found τ has a much larger impact on OFDV PSNR and find that larger τ is useful before a limit is reached, these results are shown in Fig 3.7.

3.5 Video Results

We have included 3 videos from the test set at each noise level. In each video we include the reference video, ground truth LLV, noisy LLV, the OFDV, our result, FastDVDNet's result, and V-BM4D's result. A brief description of each included video:

- Video 323: There is slight movement in this video from tool use.

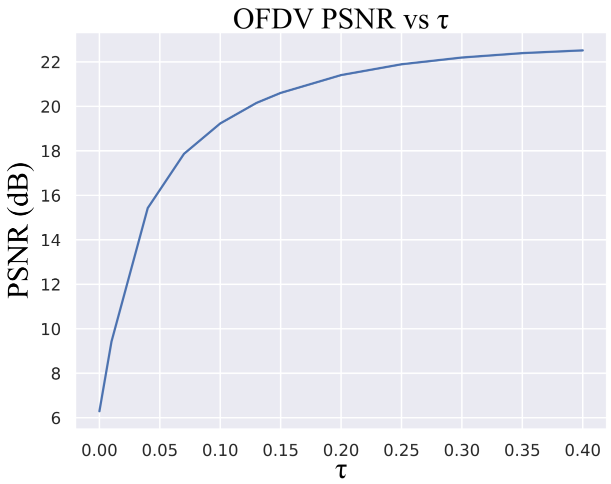


Figure 3.7: **OFDV PSNR vs τ:** This plot shows our OFDV reconstruction PSNR with respect to changing the τ parameter.

- Video 383: In this video there is moderate movement due to pulling tissue.
- Video 499: The camera is bumped in this video leading to large motion in the scene.

We also include a video of denoised results from our ablation study. The result includes 25 test videos from the neural network denoised output with different input configuration. We use the following different inputs: OFDV:FWBW, OFDV:FW, 5 noisy LLV frames, 17 noisy LLV frames, and only reference frames. These results are at $\mathrm{SBR} = 0.5$.

3.6 Supplementary Section: Neural Network Architecture

We use the same general network architecture as FastDVDNet Tassano et al. (2020) for our video denoising network. Suppl. Fig. 3.8 shows a diagram of the architecture. When denoising an OFDV frames $\widehat{F}_t^{flow}$, the network takes four of its

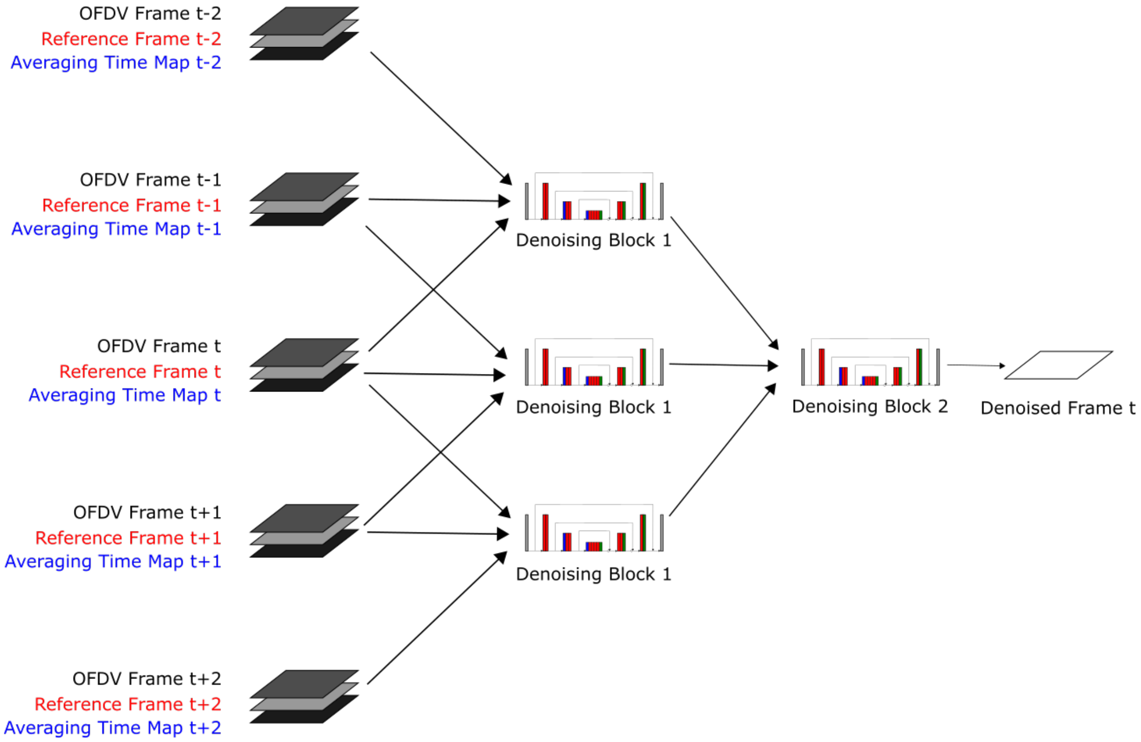


Figure 3.8: **Diagram of the video denoising network**

neighboring OFDV frames $\widehat{F}^{flow}_{\{t-2,t-1,t+1,t+2\}}$ along with their corresponding reference frames $W_{[t-2:t+2]}$ and averaging time maps $N_{[t-2:t+2]}$ as inputs. During the forward pass, adjacent OFDV frames along with their corresponding reference frames and averaging time maps are passed into denoising blocks in groups of three. A diagram of the denoising block is shown in Fig. 3.9.

3.7 Supplementary Section: Training Details

Our CNN is trained on the output frames from the optical flow denoising algorithm or OFDV frames. The training dataset consists of input/ground-truth frame pairs P^k_t:

$$P^k_t = \left((\widehat{F}^{flow,k}_{[t-2:t+2]}, W^k_{[t-2:t+2]}, N^k_{[t-2:t+2]}), F^k_t \right)$$

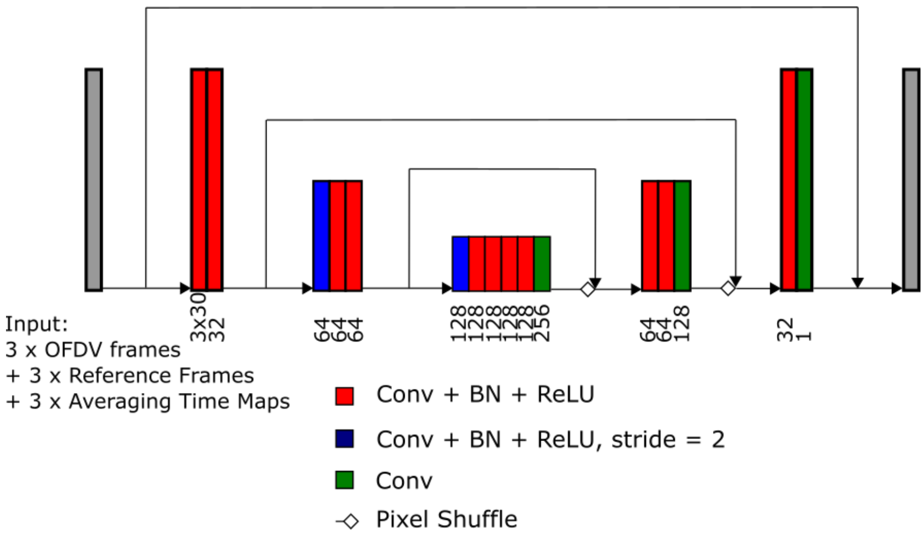


Figure 3.9: **Architecture of the denoising block**

where $k \in \{1, 2, \cdots, n_{tot}\}$.

For a given noise level, the training dataset contains a total number of $n_{tot} = 1000$ input/ground-truth pairs. We sample 1000 ground-truth fluorescent frames from the first 250 out of 590 ground-truth LLVs, and fill in their corresponding OFDV frames $\widehat{F}^{flow,k}_{[t-2:t+2]}$, reference frames $W^{k}_{[t-2:t+2]}$, and averaging time maps $N^{k}_{[t-2:t+2]}$.

We use Mean Squared Error (MSE) as our loss function:

$$L(\theta) = \frac{1}{n_{tot}} \sum_{k=1}^{n_{tot}} \|\widehat{F}^{net,k}_{t} - G^{k}_{t}\|^{2} \tag{3.9}$$

where θ is the set of learnable parameters; $\widehat{F}^{net,k}_{t}$ is the output of the CNN.

ADAM optimizern (Kingma and Ba (2015)) is used to minimize the loss function with all its parameters set to default values and the initial leaning rate set to 10^{-3}. The network is trained for 100 epochs with a batch size of 8. Our CNN is trained separately for three different noise levels, SBR = $[0.1, 0.5, 2]$.

Retraining FastDVDNet: When comparing our method to FastDVDNet (Tassano et al. (2020)) we found it was necessary to retrain FastDVDNet on our dataset and noise levels in order for FastDVDNet to produce reasonable results. We feed

FastDVDNet the raw LLV frames and an estimated noise map. We estimate a constant noise map for each frame by using the average number of photon counts captured at each frame to estimate the noise standard deviation (see Appendix 3.8). We train FastDVDNet using the same training procedure as our method.

3.8 Supplementary Section: Finding Equivalent Gaussian Standard Deviation

Before we generate Poisson noise we scale our images between $[\phi_{\text{back}}, \phi_{\text{sig}}]$ then we add Poisson noise and re-scale the noisy image between [0,255]. For comparison to methods using the standard deviation of additive white Gaussian noise, the comparable standard deviation of our Poisson corrupted images scaled between [0,255] is given by,

$$\sigma = \sqrt{\phi}\frac{255}{\phi_{\text{sig}}} \tag{3.10}$$

where $\phi \in [\phi_{\text{back}}, \phi_{\text{sig}}]$ is the expected numbers of photons for a given pixel. Although, Poisson noise is signal dependent we can get an estimate of the noise in a scene by using $\phi = \frac{1}{2}(\phi_{\text{sig}} + 2\phi_{\text{back}})$, which gives us a standard deviation of $\sigma = [826, 180, 57]$ for our three noise levels of SBR$= [0.1, 0.5, 2]$.

3.9 Supplementary Section: Optical Flow Failures

Figure 3.10 shows five consecutive LLV, RV, OFDV denoised, OFDVDnet denoised and ground truth frames. The red box circles out a region with optical flow failures due to rapid occluder (a pair of tweezers) movements in that area. The rapid movement of occluder causes optical flow to fail repeatedly in short time intervals for those regions around the occluder, leading to less pixels being averaged over time and more noise remaining in those pixels. As shown in Figure 3.10, for each of the five OFDV frames, the area inside the red box has significantly more noise

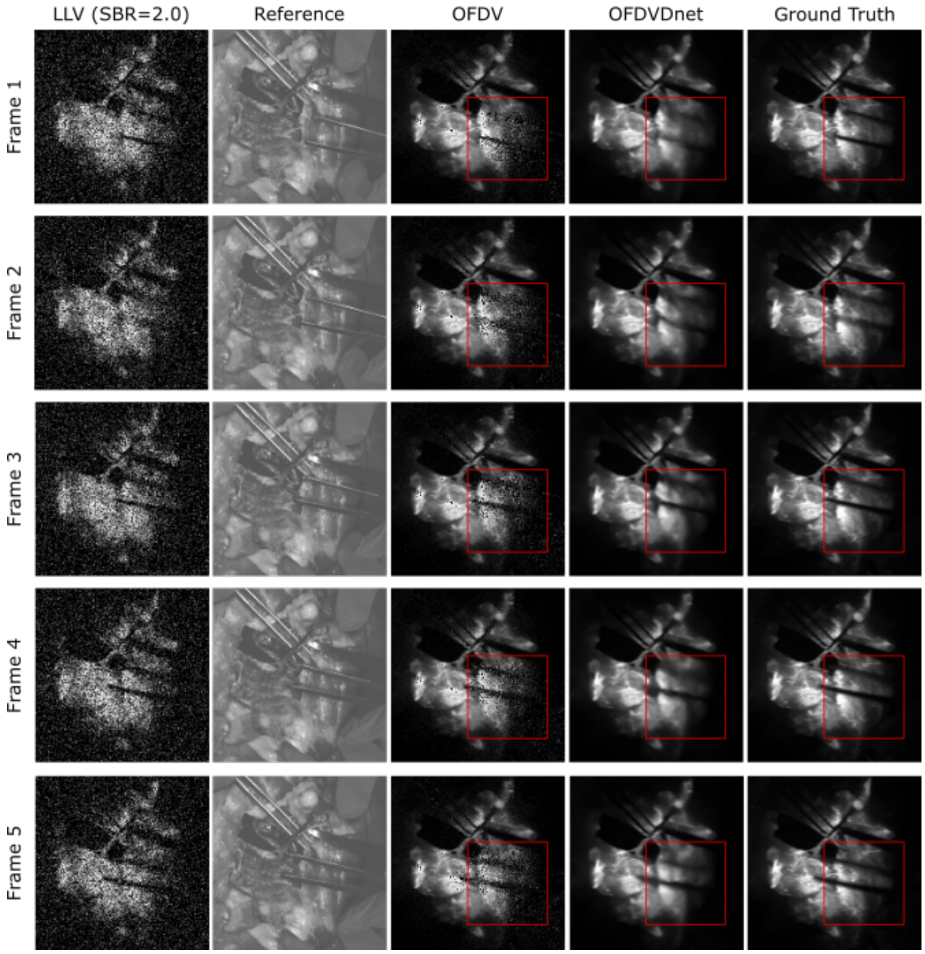


Figure 3.10: **Example case with optical flow failures**

compared to areas outside the box. Because the OFDV frames are more noisy in areas with rapid occluder movements, the OFDVDnet also tend to blur out more details in those areas compared to other areas in the video frames during the neural network denoising step.

4 VIDEO DENOISING IN FLUORESCENCE GUIDED SURGERY

There are 320 million major surgeries performed worldwide every year in which 20 to 30% of patients require re-admittance or have serious postoperative morbidity (Dobson (2020)). Many of these issues are due to difficulty in visualizing or identifying tissues that need removal or that should be avoided; for example, in an estimated 21% of prostate cancer removal surgeries, cancer is found on the margin of removed tissues indicating cancer was likely left in the patient (Orosco et al. (2018)). The current standard of care (SOC) in many surgeries rely heavily on non-quantitative measures such as surgeon perception of tissue under normal lighting or tactile tissue cues. Fluorescence guided surgery (FGS) is a promising technique to improve the SOC by giving surgeons a quantifiable fluorescence video feed that helps identify the state of different tissues leading to an improvement in surgical decision making and an improvement in patient outcomes (Sutton et al. (2023)). FGS relies on a fluorescent contrast agent, either a drug or a naturally occurring fluorophore, that when imaged by an FGS imaging system helps delineate or classify tissues of interest. The most commonly used clinical contrast agents, such as indocynanine green (ICG), are bright and operate in the near infrared while others are dim and exist in the visible light spectrum so are more challenging to capture; these dim agents may not produce enough photons at video frame rates to be clinically viable. In this work, we consider software video denoising as a relatively unexplored and promising path forward to increase the sensitivity of current systems which will increase the number of clinically viable contrast agents.

The operation of an FGS imaging system is shown in Fig. 4.1(a); first the system emits excitation light that excites the contrast agent, then the agent emits fluorescent emission light. The system collects both emission light and reflected excitation light from a scene point then an emission filter removes much of the excitation light. However, filters are imperfect so some light is not filtered out; we call unfiltered

This chapter is based on work that is currently being prepared for a journal publication with authors Trevor Seets and Andreas Velten.

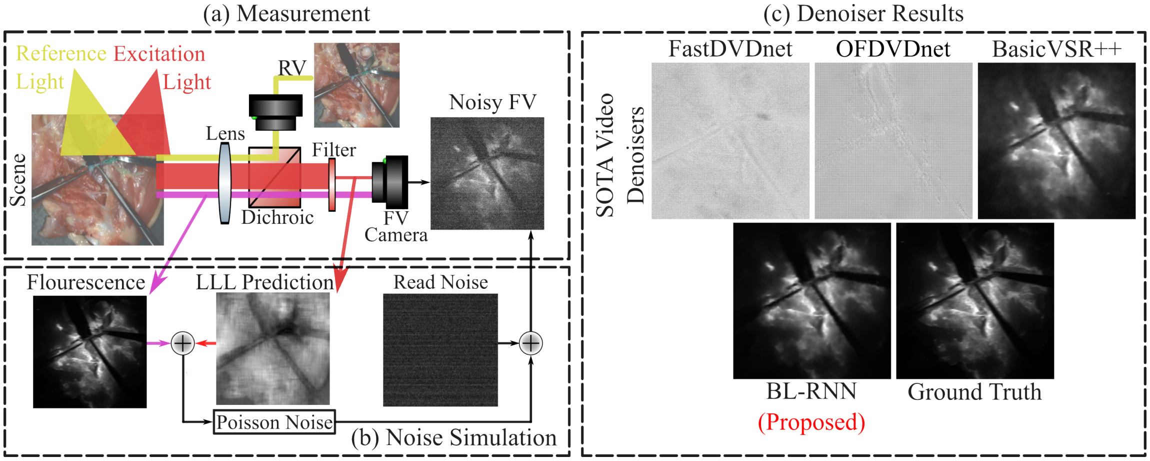


Figure 4.1: **Measurement and Noise:** (a) In the FGS measurement process, excitation laser light and reference light shine onto the scene where the excitation light produces fluorescence at a higher wavelength. Three relevant spectral bands are imaged by the FGS system. The reference band (yellow) is isolated using a dichoric beamsplitter and imaged by the RV camera. The excitation laser band (red) is attenuated by the emission filter leaving the laser leakage light (LLL) which is combined with the fluorescence band (pink) and imaged on the FV camera. (b) In our noise simulation, we combine a clean fluorescence image with an LLL prediction from our LLL-PN. Then we apply Poisson noise and add sampled read noise frames to produce the final noisy FV. (c) shows the results of different denoisers on a noisy FV. State-of-the-art (SOTA) video denoisers struggle while our proposed BL-RNN performs well on this task.

excitation light the **Laser Leakage Light (LLL)** which can be similar in brightness to emission light, and is a core difficulty in improving FGS systems (DSouza et al. (2016); Olson et al. (2019); Pogue et al. (2023)). The LLL and emission photons are added together by the fluorescence camera which outputs the **fluorescence video (FV)**. A secondary **reference video (RV)** with the same field of view is simultaneously captured by a RGB camera using spectral and temporal filtering strategies (Velten et al. (2020)). The goal of a FGS denoiser is to take as input the FV and RV to produce a clean denoised FV while removing LLL. We find conventional video denoisers struggle in FGS denoising and LLL removal, so we develop a new set of baseline methods for FGS video denoising.

FGS video denoising differs from standard video denoising in four key ways. First, the RV provides a helpful source of secondary information for computer vision algorithms such as providing structural and motion cues that can be used to improve denoising performance (Seets et al. (2024b)). We find the RV is key to simulating and removing the LL. Second, the noise levels in FGS may be much higher than in standard video denoising problems potentially requiring long range temporal integration or larger efficient models. Third, a usable FGS denoiser must be real-time capable to fit into the clinical workflow. To remain hardware agnostic, we require methods to be causal where only the past and current frames are available; most conventional video denoising methods are non-causal.

Finally, the noise present in the FV contains the standard shot and camera noise terms, but also an additional LLL noise term. Unlike prior work (Seets et al. (2024b)) which does not consider LLL, we model the LLL term as a spatially varying shot noise term that we predict using our LLL prediction network (LLL-PN). The LLL-PN takes as input a RV frame and outputs a LLL prediction. We note our strategy for dealing with LLL may also be used to deal with noise from naturally occurring fluorescence called auto-fluorescence if it is correlated with the RV. We then use our LLL-PN within a realistic noise model, shown in Fig. 4.1(b), that accurately simulates data seen on a commercial FGS system. We use simulated data to train our denoising algorithms for a wide array of signal and noise levels before testing them on real noisy data.

Surprisingly, we find that state of the art video denoisers when adapted and retrained for this problem struggle, as shown in Fig. 4.1(c). For example, the causal version of BasicVSR++ (Chan et al. (2022a)) has trouble removing the strong LLL in this example leading to signal in the denoised result where there would be none. Surprisingly, we find that NafNet (Chen et al. (2022)), an image denoiser that uses no temporal information, outperforms these video denoisers on this dataset while the opposite is true for conventional video denoising. NafNet also is more efficient to train, taking one third of the training time as BasicVSR++. Training efficiency is extremely important when dealing with data intensive problems such as video denoising, because it allows for practical training of larger models. Motivated by

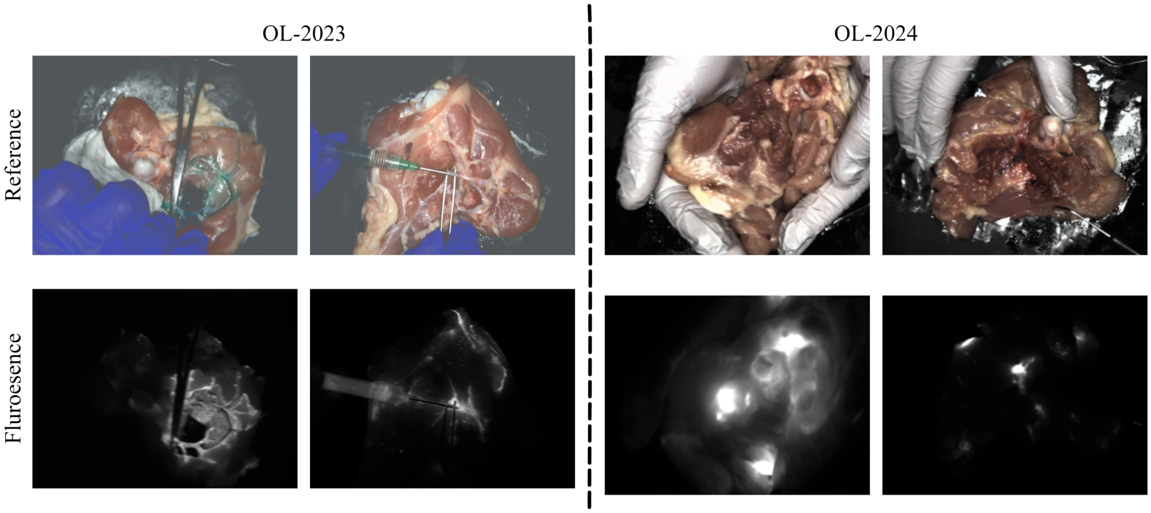


Figure 4.2: **Dataset Example Images:** Here we show two example images for both OL-2023 and OL-2024. OL-2023 focuses on vasculature where as OL-2024 focuses on local fluorescent regions.

these findings we combine NafNet with different temporal propagation techniques from video denoisers to create a strong baseline model for FGS video denoising. We propose a recurrent structure, BL-RNN, with a NafNet backbone that provides robust performance with minimal network complexity and efficient training times. **Contributions:**

- We double the size of existing FGS datasets, including new data necessary for properly simulating noise and real dim signals for testing.
- We propose a novel method for simulating and removing spatially varying LLL in FGS, and simulate a specific commercial camera's sensor noise.
- We propose new network architectures for causal FGS video denoising that incorporates the most effective aspects of state of the art standard video and image denoising methods.

4.1 Dataset

To the best of our knowledge, the OL-2023 dataset (Seets et al. (2024b)) is currently the only publicly available FGS dataset with both FV and RV. OL-2023 contains 100 minutes of mock surgery video using the blue blood chicken surgical model (Albano et al. (2021)). OL-2023 contains a number of surgical actions with a large focus on perfusion imaging with fluorescent injection sites being primarily vascular structures and focusing on slower motion scenarios. We expand the scope of this data with the new OL-2024 dataset with challenging motion scenarios and a focus on cancer and lymphatic surgeries. We follow a similar experimental setup that was used to create OL-2023; we use the OnLume Avata clinical FGS imaging system (OnLume Surgical, Madison, WI), chicken thighs as our mock surgical patient, and high concentrations of indocyanine green (ICG) to generate very bright fluorescence that can be used as ground truth FV without LLL. In addition to the new mock surgical data we also capture a calibration and real noise test set. All videos are at 15 frames per second with a resolution of 768 by 1024 for both the RV and the FV.

OL-2024 and OL-Combined: The OL-2024 dataset contains 130 minutes of new mock surgical video, each video contains aligned RV, R_t^v and low-noise FV, S_t frames. OL-2024 has a focus on simulating surgeries similar to cancer resections or lymphatics while also introducing more challenging motion scenarios. We primarily inject ICG into muscle or fat causing irregular fluorescent blooming with substantial scattering, modeling surgeries with patches of embedded fluorescent tissue. Example images from OL-2023 and OL-2024 are shown in Fig. 4.2. Additionally, unlike OL-2023, videos in OL-2024 are largely contiguous, up 18 minutes, allowing testing of long range dependencies. While these long videos are not the focus of this work we expect them to be important in future work. We combine OL-2023 and OL-2024 into core dataset, **OL-Combined** which we split into a training and testing set.

OL-Calibration: OL-Calibration contains three video subsets used to calibrate our realistic noise model.

- **OL-Dark** contains 5,325 frames (6 min) of dark video that is captured by

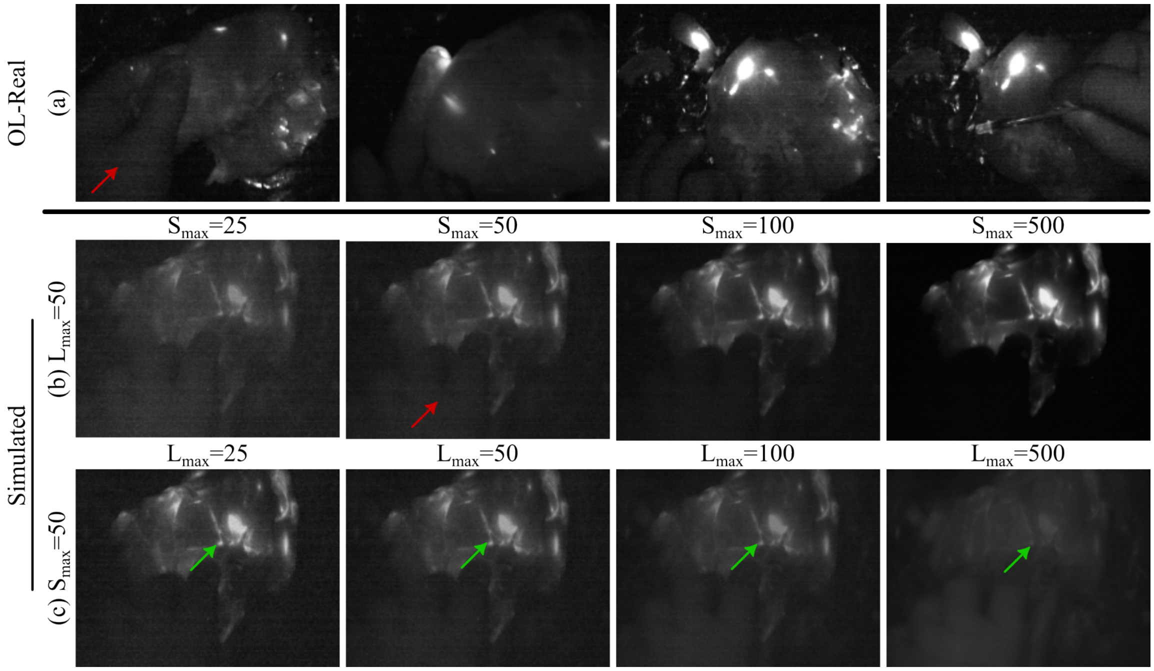


Figure 4.3: **Real and Simulated Data:** (a) frames from our OL-Real test set. (b) simulated frames with increasing signal levels and fixed $L_m = 50$. Qualitatively, $S_m = L_m = 50$ closely matches many of the real data frames. Notice the hand (red arrow) has similar signal levels in both the real and simulated data at these parameters. (c) simulated frames with increasing L_m and $Sm = 50$. Notice that the small fluorescent features (green arrow) fade as L_m increases.

putting a lens cap on the system. This set is used to simulate the sensor's read noise.

- **OL-Phantom** contains 1,830 frames (90 seconds) of a Quel phantom (RCS-ICG-ST01-QUEL03)(Ruiz et al. (2020)) for camera gain calibration. A fluorescence phantom is designed to mimic fluorescence properties of a contrast agent and acts as a fluorescent standard. The Quel phantom is a 3 by 3 grid of calibrated phantom wells of varying concentrations that is used for calibration and testing of FGS systems.

- **OL-LLL** contains 18,576 frames (20 min) of a chicken thigh mock surgery (Albano et al. (2021)) with no fluorescent compound, so the videos only contains ambient, laser leakage light, and read noise. We use this dataset for training our LLL-PN in our noise model.

OL-LLL and OL-Dark are split into training and testing sets.

OL-Real: Finally, we capture the OL-Real dataset which contains about 15 min (13,033 frames) of mock surgery with low doses of ICG that is used to test our models with real dim noisy data. We inject multiple batches of ICG with different dilutions near the noise floor of our system to produce noisy data. Example frames from OL-Real are shown in Fig. 4.3 (a).

4.2 Fluorescence Noise Simulation

In order to train denoising algorithms, we add realistic noise to the OL-Combined FV frames and train our algorithms to remove this noise. We model three key sources of noise in fluorescence data, (1) shot noise originating from the inherent Poisson nature of light (Hasinoff (2014)), (2) camera read noise from imperfections in sensors (Barnes and Allan (1966); Konnik and Welsh (2014); Baer (2006); Snoeij et al. (2006)), and (3) laser leakage light (LLL) coming from excitation light not being fully blocked by the emission filter. The first two sources of noise (1-2) have been extensively studied for general image denoising problems (Wang et al. (2019a); Wei et al. (2020); Zhang et al. (2021)) while the third source of noise is generally treated as a hardware problem and not considered in denoising contexts.

We decompose the noise of our system into read noise (signal independent) and shot noise (signal dependent). We model the LLL term as a spatially varying additive photon count in the shot noise term. It is difficult to capture both the fluorescence and LLL frame simultaneously without hardware changes, so to simulate LLL we approximate it using our LLL-Prediction Network which takes the RV as input and outputs a LLL frame prediction (details in Sec. 4.2). Let S_t be the fluorescent signal of interest, L_t^{LL} be a predicted LLL frame, and R_t be the read noise of the camera at frame t, all scaled between 0 and 1. Then a noisy FV frame, $\tilde{F}_t^v$, is given by,

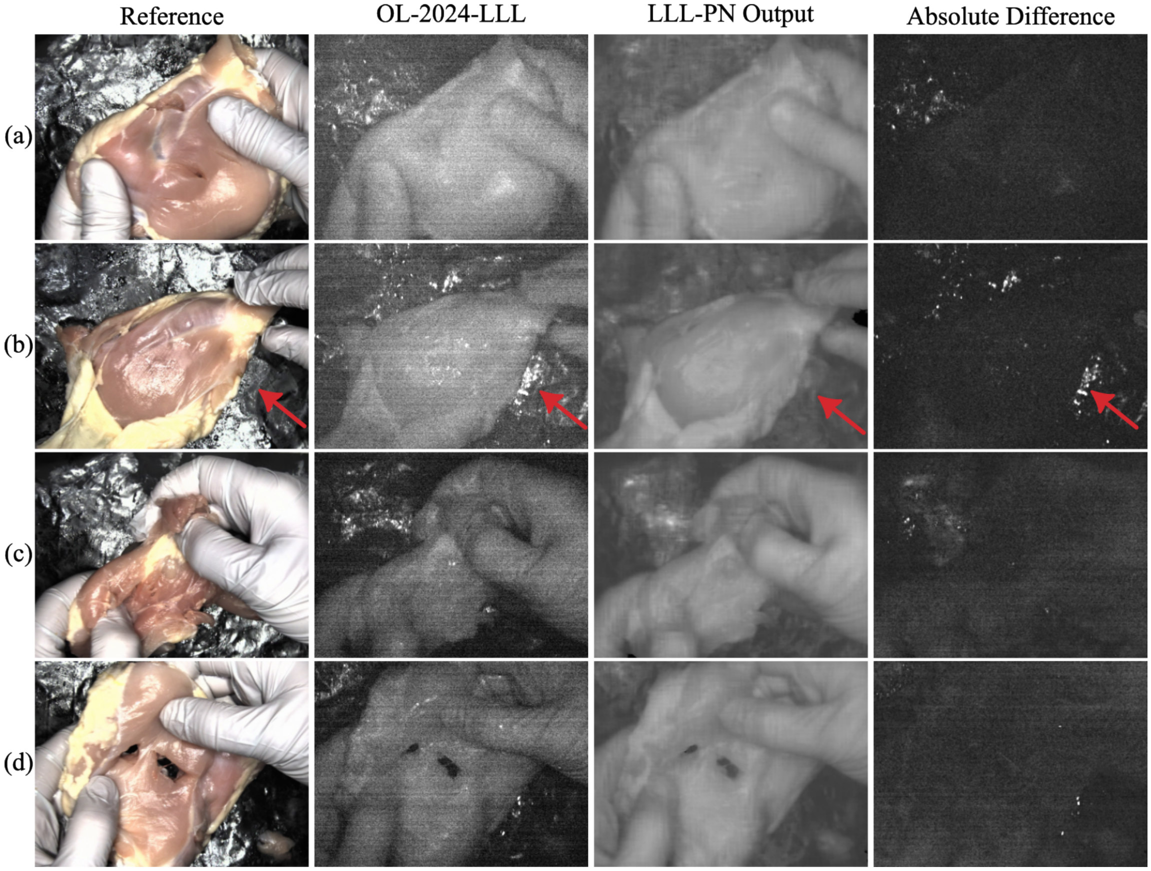


Figure 4.4: **LLL and LLL-PN:** This figure shows examples from OL-LLL and our corresponding LLL-PN predictions. The last row shows the absolute difference between the LLL frame and prediction. The LLL-PN is able to correctly predict most of the large structure leaving shot and read noise. The LLL-PN struggles with features that are uncorrelated to the reference image; for example, specular reflections (red arrow) are particularly difficult.

$$\tilde{F}_t^{\nu} = \frac{1}{KS_m}\text{Quant}(K\text{Pois}(S_mS_t + L_mL_t^{LL}) + \frac{R_t}{R_m}) \tag{4.1}$$

where Pois(λ) is a Poisson random variable with parameter λ, K is the gain of the camera, Quant represents the 12-bit quantizer of the system camera, and (S_m, L_m, R_m) are scaling factors that represent the amount of noise. S_m and L_m represents the maximum number of fluorescence signal and LLL photons in a pixel, respectively. S_m and L_m can vary greatly in different scenes so define the core 2 dimensional noise space we train and test on. R_m controls the amount of read noise and is treated as a data augmentation parameter that will avoid overfitting on specific read noise levels that change with camera settings or ambient temperature. Note Eq. 4.1 is scaled so if there is no LLL or read noise $(L_m = 0, R_t = 0)$, $E[\tilde{F}_t^v] = S_t$ which we find helps in training.

In order to calibrate our noise model to a specific commercial camera we use OL-Calibration. We solve for the gain K of our camera using OL-Phantom and the fact that the Poisson distribution will have mean equal to variance. We find the K value that best fits a Poisson distribution over all Quel phantom wells, see supplement for details. Next we simulate R_t by randomly sampling time-contiguous dark frames from OL-Dark which recent work (Zhang et al. (2021)) has shown to be more accurate than the common Gaussian random variable or other physics based models for read noise. Next, we use a low noise FV from OL-Combined for S_t, and find an estimate for L_t^{LL} using our LLL-PN. In Fig. 4.3, we show images from OL-Real as well as our simulated data with varying S_m and L_m levels. Notice as L_m increases non-fluorescent objects begin to appear in $\tilde{F}_t^v$; for example, hands are a good sign of significant LLL, but also generally the signal is obscured by LLL from tissue.

Laser Leakage Light Prediction Network

Our Laser Leakage Light Prediction Network (LLL-PN) takes as input a RV frame, R_t^v and predicts a corresponding LLL frame, L_t^{LL}, to be used in our noise simulations. Our goal is to train our LLL-PN, $f_{LLL}(R_t^v)$, such that,

$$f_{LLL}(R_t^v) \approx L_t^{LL}. \tag{4.2}$$

We choose NafNet32 for f_{LLL} and train it using OL-LLL which has constant working distance and laser power which may affect the brightness of L_t^{LL}; these are dealt with in our noise simulation through L_m. For most scenes, L_t^{LL} is close in intensity to the read noise of the system making it difficult to capture noise free examples. Because of this we train f_{LLL} with noisy L_t^{LL}, similar in spirit to Noise2Noise (Lehtinen et al. (2018)). Because the RV is uncorrelated to the noise in L_t^{LL}, when trained with an L_1 loss, f_{LLL} learns to predict LLL frames without shot or read noise. Specifically, an L_1 loss is median seeking so f_{LLL} learns to predict the median value of similar patches in the noisy L_t^{LL} within the training set (Lehtinen et al. (2018)). We note that our LLL-PN hinges on the assumption that the RV can be used to predict near infrared reflectivity, which we find to be true in OL-LLL, but will be important to check in human data.

Fig. 4.4 shows example qualitative results, note that when predicted LLL is subtracted from the noisy L_t^{LL}, the result still contains read noise patterns but strong LLL spatial patterns vanish. The LLL-PN struggles to predict specular reflections which is to be expected because the laser and reference lights are not co-located so specular reflections in the FV should be very difficult to predict from the RV alone without 3D scene information. We also find that other 3D dependent structures such as shadows of hands and other tools are difficult to predict for the LLL-PN. f_{LLL} is able to account for 40% of the total energy (L_2 norm) in noisy LLL frames. This coupled with our qualitative results indicate a strong ability to predict LLL. See Suppl. Sec. 4.7 for training details.

4.3 FGS Video Denoisers

The goal in FGS video denoising is to take as input a set of noisy FV and RV frames, $\{\tilde{F}_\tau^v, R_\tau^v\}$, and output a set of denoised fluorescent frames $\{\hat{S}_\tau\}$. We simulate simulate noisy frames using the clean FV in OL-Combined as S_t in our noise model and train the algorithms to recover S_t. Unless otherwise noted the test set used is from OL-Combined.

Conventional denoising methods

There are a number of candidate state of the art (SOTA) models for denoising for both image (Zhang et al. (2023); Chen et al. (2022); Zamir et al. (2022, 2021); Chen et al. (2021); Wang et al. (2022); Tu et al. (2022); Cheng et al. (2021); Zamir et al. (2020)) and video (Tassano et al. (2019); Ostrowski et al. (2022); Tassano et al. (2020); Wang et al. (2019b); Yue et al. (2020); Qi et al. (2022); Van Veen et al. (2021); Huang et al. (2015); Maggioni et al. (2021); Chan et al. (2021a, 2022a); Cao et al. (2023); Xiang et al. (2022); Chen et al. (2016); Arias and Morel (2017); Maggioni et al. (2012); Hasinoff et al. (2016); Xu et al. (2020); Davy et al. (2019); Vaksman et al. (2021); Seets et al. (2024b); Tang et al. (2023)) on conventional datasets that could be used as comparisons for our proposed baselines. However, it would be impractical and expensive to update all methods to properly use the RV, force causality, and retrain them on our dataset. Instead, we choose four models to act as our SOTA comparisons, one image denoiser and three video denoisers. We modify these four methods to incorporate the RV and to become causal. We retrain these methods on OL-Combined.

For the image denoising SOTA comparison method, we choose NAFnet (Chen et al. (2022)). NAFnet is a state of the art image denoiser which combines key elements of other image denoisers with a goal of simplifying the core architecture resulting in a light weight network with strong performance. NAFnet uses a simplified channel attention (Hu et al. (2018)) for global information aggregation, depth wise convolutions (Han et al. (2021); Liu et al. (2022)) for local information, layer normalization (Ba et al. (2016)) to stabilize training and "Simple Gate" as the non-linearity.

We split the video denoisers into three categories based on their temporal propagation strategy and choose one method from each category as our SOTA comparison method. The first category is **sliding window** strategies (Tassano et al. (2019); Ostrowski et al. (2022); Tassano et al. (2020); Wang et al. (2019b); Yue et al. (2020); Qi et al. (2022); Van Veen et al. (2021)) which have a fixed temporal receptive field that is sequentially moved throughout the video sequence. We choose FastDVDNet (Tassano et al. (2020)) due to its strong conventional video denoising performance. FastDVDNet relies on a cascaded approach where successive frames are grouped together as inputs into an initial U-net (Ronneberger et al. (2015)) block whose outputs are fused with temporally adjacent blocks as input into a second U-net.

The second strategy is based on **Recurrent Neural Networks** (RNNs) (Huang et al. (2015); Maggioni et al. (2021); Chan et al. (2021a, 2022a); Cao et al. (2023); Xiang et al. (2022); Chen et al. (2016)). RNNs can pass information forward and backward in time recursively throughout the video, in principle allowing for long range temporal aggregation. We choose BasicVSR++ (Chan et al. (2022a)) because it achieves SOTA conventional denoising performance (Chan et al. (2022b)). BasicVSR++ uses a grid propagation strategy based on flow guided (Chan et al. (2021b)) deformable convolutions (Dai et al. (2017); Zhu et al. (2019)) with second order connections where features are aligned, aggregated and propagated in time.

The third temporal strategy are **Align and Merge** (A&M) methods (Arias and Morel (2017); Maggioni et al. (2012); Hasinoff et al. (2016); Xu et al. (2020); Davy et al. (2019); Vaksman et al. (2021); Seets et al. (2024b); Tang et al. (2023)). A&M strategies in deep learning are extensions of block (Arias and Morel (2017); Maggioni et al. (2012)) or patch matching (Hasinoff et al. (2016)) non-learning based methods. These strategies rely on combining multiple observations of the same object across time to reduce noise variance. These methods align or match regions of the video to deal with motion and then combine these matched regions in a merge step to denoise the video. For this category we choose OFDVDnet (Seets et al. (2024b)) because it was developed for FGS denoising. OFDVDnet uses an explicit align and merge step to aggregate many frames of temporal information to be used as input into downstream CNN without increasing the memory requirements of training large temporal receptive fields.

Modifications to SOTA Methods

We modify the SOTA methods to include the RV and become causal.

RV incorporation: We incorporate the RV in two locations for the SOTA comparison methods. First, we append it in the channel dimension on the input layer allowing the network to fully use the RV information. Second, when possible we use the RV in any relevant alignment module. Using the RV for alignment has been

Table 4.1: **SOTA Results:** State of the art video denoising models retrained on OL-Combined struggle to outperform NafNet32, an image denoiser. Metrics are listed as PSNR/SSIM and a higher value is better for both. Results are for $S_m = L_m = 50$.

Model	PSNR/SSIM
NafNet32	34.8/0.735
BasicVSR++^{C}	30.5/0.639
OFDVDnet^C	2.5/0.000
FastDVDnet^C	6.1/0.020

shown to produce better results (Seets et al. (2024b)) because it has less noise than the FV.

Causality: All SOTA methods are non-causal so need to be modified. We do this by ensuring no future frames can be accessed by the present frame. In FastDVDnet and OFDVDnet we do this by shifting the cascaded U-nets to take in the 4 previous frames along with the current frame instead of using the 2 previous and 2 future frames. BasicVSR++ uses a bi-directional recursive propagation strategy with alternating branches moving information forward or backward in time. In BasicVSR++ we ensure causality by only using switching the direction of the backward propagation branches to forward propagation branches while maintaining the same network size. We indicate these are the causal versions of these SOTA networks with the superscript ^{C}, for example BasicVSR++^{C}.

SOTA Results

We find that the video denoising SOTA methods are unsuitable for this problem. FastDVDnet^C and OFDVDnet^C are unable to deal with the LLL leading to poor performance. While BasicVSR++^{C} is able to produce reasonable results, it occasionally struggles with large motion and LLL removal. Table. 4.1 displays the PSNR/SSIM the SOTA comparison methods at the test values of $S_m = L_m = 50$. FastDVDnet^C and OFDVDnet^C are unable to produce reasonable results at all with a PSNR under 7dB. NafNet outperforms BasicVSR++^{C} by 4.3dB while also using one third of the training resources. This is a surprising result, so we test whether or not this is a property of our data or the models by training our causal version of BasicVSR++^{C} and NafNet32 on Davis (Pont-Tuset et al. (2017)) which is commonly used in video denoising. BasicVSR++^{C} outperforms NafNet32 by a large margin at all noise levels on Davis (details in Suppl. Sec. 4.9), indicating that NafNet32 has unique properties suited to denoising FGS data. This finding motivates the design of our baseline models.

FGS Baseline Models

Because NafNet shows such strong performance on OL-Combined, we choose to combine the core components of Nafnet with temporal propagation ideas of each temporal category to create a set of new baseline models. We choose to create simple baselines when possible because we believe strong simple baselines are important for future comparisons. An overview of our baseline models is shown in Fig. 4.5.

Residual Blocks: For the primary residual blocks in our baseline models we use the NAFblock (Chen et al. (2022)) from NAFnet which contains a channel attention module for global information, layer norms for stable training, and depth wise convolutions to lower parameters in the convolutional layers. We use the popular U-net structure as the primary backbone of all models, represented as $f(\cdot;\Theta)$ in the following subsections.

BL-SW

A sliding window strategy takes as input a fixed number of previous frames and the current frame to denoise the current frame. We use a cascaded U-net structure similar to FastDVDnet. Our sliding window baseline (BL-SW) takes as input 5 frames,

$$\hat{S}_t^k = f_1(\{\tilde{F}_\tau^\nu, R_\tau^\nu\}_{\tau=t-k-2}^{t-k}; \Theta) \tag{4.3}$$

$$\hat{S}_t = f_2(\{\hat{S}_t^k\}_{k=0}^2, \{R_\tau^\nu\}_{\tau=t-4}^t; \Theta). \tag{4.4}$$

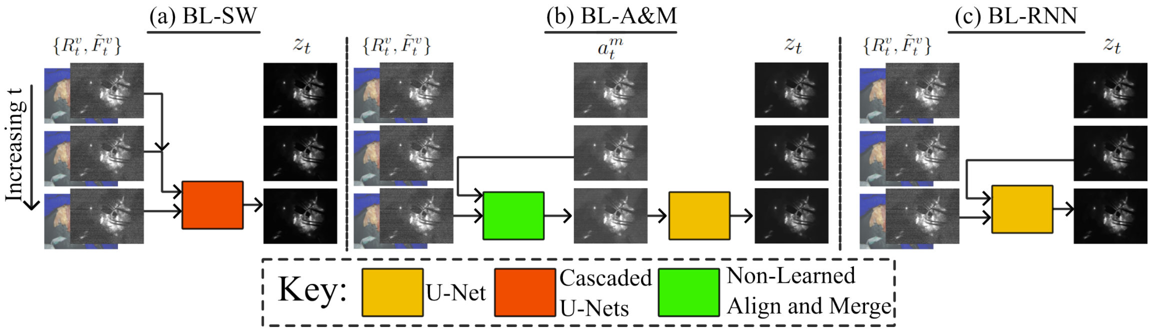

(a) BL-SW
$\{R_t^v, \tilde{F}_t^v\}$
z_t
Increasing t
(b) BL-A&M
$\{R_t^v, \tilde{F}_t^v\}$
a_t^m
z_t
(c) BL-RNN
$\{R_t^v, \tilde{F}_t^v\}$
z_t
Key:
U-Net
Cascaded U-Nets
Non-Learned Align and Merge

Figure 4.5: **Baseline Model Overview:** We propose 3 baseline: (a) SW-BL uses a fixed temporal receptive field, (b) BL-A&M uses a recursive non-learning based align and merge module followed by a U-net block, and (c) RNN-BL uses a simple recursive input.

where $\hat{S}_t$ is the denoised output, and $\hat{S}_t^k$ are intermediate feature frames with double the number of channels as the input. f_1 and f_2 are NafNet24 U-nets. Note FastDVDnet only computes $\hat{S}_t^0$ and $\hat{S}_t^2$, so BL-SW has an extra U-Net forward pass. But $\hat{S}_t^k$'s can be cached from previous t's, so BL-SW does not require much more computation.

BL-A&M

The baseline align and merge (BL-A&M) first uses a non-learned function that temporally aligns and then merges frames together before the output is fed into a downstream network. While there are many candidates for both the alignment and merge functions, we choose to use the A&M strategy from OFDVDnet (Seets et al. (2024b)) because it is the SOTA in FGS denoising. The causal version of the A&M strategy in OFDVDnet, A^M, relies on calculating optical flow between every successive frame, and using a recursively defined motion corrected sum with occlusion rejection to aggregate information across time. Our BL-A&M follows the following formulation,

$$a_t^m = A^M(\{\tilde{F}_\tau^\nu, R_\tau^\nu\}_{\tau=t-1}^t, a_{t-1}^m) \tag{4.5}$$

$$\hat{S}_t = f(a_t^m, \tilde{F}_t^\nu, \tilde{R}_t^\nu; \Theta). \tag{4.6}$$

Where a_t^m is the intermediate temporally averaged frame, equivalent to OFDV-Forward from (Seets et al. (2024b)). We change the downstream network, $f(\cdot;\Theta)$, used in OFDVDnet to match Nafnet32.

BL-RNN

A recurrent strategy uses past computation in a recurrent manner, for this baseline model we use a simple recurrence relation that uses a previous denoised output frame as input into the next denoising block. Our baseline recurrent neural network (BL-RNN) takes the following form,

$$\hat{S}_t = f(\hat{S}_{t-1}, \{\tilde{F}_\tau^\nu, R_\tau^\nu\}_{\tau=t-1}^t; \Theta). \tag{4.7}$$

where $\hat{S}_t$ is the output frame at time t and f is a NafNet32 U-net. We found that for this model replacing the SimpleGate activation functions with ReLu activation functions provide slight performance improvements when LLL is present, so we use ReLu activation functions in our BL-RNN. We experimented with using the recurrent structure from BasicVSR++ with NafNet residual blocks, but found that these networks where unstable and produced large artifacts.

Training Details:

All networks are trained with a cosine annealing training scheme and Charbonnier loss (Charbonnier et al. (1994)) because it stabilizes training by reducing impact of outliers (Lai et al. (2017)). We train the networks for a maximum of 2 weeks on the same hardware (Nvidia A100), we choose 2 weeks as this is how long it takes to train BasicVSR++^{C} following the original training scheme (300k iterations).

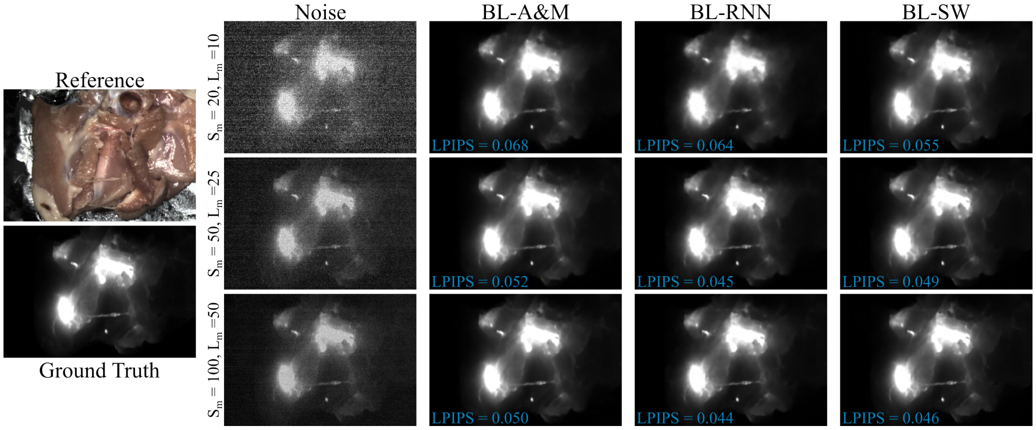


Figure 4.6: **Proposed Baseline Results:** This figure shows the results of our baseline models over different numbers of photons with constant $\frac{L_m}{S_m}$. Notice as signal gets lower the read noise structure becomes more significant and BL-SW outperforms, whereas at higher signal levels BL-RNN gives the best results.

We find FastDVDnet^C, NafNet32, and BL-RNN converge in only 1 week, so we stop training at 1 week for these models to prevent over-fitting. In order to speed up training of the A&M model, we first train the NafNet32 backbone for 300k iteration before training the full model for an additional 100k iterations. In order to augment our data and allow for changes in imaging distance, laser power, and other changes to the physical system, we randomly sample from a variety of S_m, K, L_m, and R_m parameters in our noise simulation during training. We train over $S_m = [10, \frac{1}{2K}]$, $L_m = [0, S_m]$, $\frac{1}{K} = [1200, 2400]$, and $R_m = [4, 8]$. More specifics on data augmentation and training parameters are available in the supplement and detailed configurations will be available with the code.

4.4 Results and Discussion

Changing S_m and L_m

Our noise model is parameterized by two parameters, S_m and L_m which represent the levels of signal and LLL, respectively. This two dimensional noise space leads to difficulty in finding the best model, and we find our baseline models are stronger in different noise regions than others. We evaluate our models throughout a range of S_m and L_m values, we evaluate our models with 3 metrics: PSNR as a per-pixel measure, SSIM (Wang et al. (2004b)) as a perceptual metric, and LPIPS (Zhang et al. (2018)) as a deep learning based perceptual metric which is trained to match human perception. In general, we found LPIPS to most agree with our perceptual ranking of images where PSNR favored over-smoothing which is problematic for small features.

First, we test our models performance for a changing L_m at a fixed $S_m = 50$ as well as a changing S_m for a fixed $\frac{L_m}{S_m}$ ratio, our quantitative metrics are summarized in Table 4.2. In general we find that our baseline models all perform reasonably well, outperforming the SOTA comparison models; however, no model is best for all noise parameters. For example Fig. 4.6 shows an example where our baseline models provide reasonable performance with BL-RNN providing the best LPIPS at high photon counts, but at lower photon counts BL-SW becomes the best performing model. Also notice that in this example the difference between qualitative performance between $S_m = 100$ and $S_m = 50$ for all 3 baselines is very small, this

Table 4.2: **Metric Performance:** These tables show our test metrics at (top) $S_m = 50$ with changing L_m values, and (bottom) at $L_m = \frac{1}{2}S_m$ with changing S_m values. The arrow next to the metrics indicate the direction of better quality. Best results are **bold**.

	PSNR↑ / SSIM↑/ LPIPS↓ at $S_m = 50$				
Model	$L_m = 0$	$L_m = 12.5$	$L_m = 25$	$L_m = 37.5$	$L_m = 50$
BasicVSR++^{C}	37.15/0.83/0.108	35.98/0.79/0.108	34.42/0.74/0.119	32.67/0.68/0.159	30.51/0.64/0.221
NafNet32	43.06/0.95/0.063	40.68/0.89/0.070	38.26/0.82/0.079	36.42/0.77/0.094	34.78/0.74/0.113
BL-SW	**43.50**/**0.94**/0.058	**41.42**/0.89/0.060	39.54/0.83/0.065	37.94/0.80/0.076	**36.44**/0.77/0.092
BL-A&M	38.59/0.89/0.063	38.15/0.88/0.063	37.55/0.87/0.064	36.84/**0.86**/**0.065**	35.92/**0.85**/**0.068**
BL-RNN	41.67/0.92/**0.053**	41.11/**0.91**/**0.053**	**39.88**/**0.89**/**0.056**	**37.96**/0.84/**0.065**	36.05/0.79/0.084

	PSNR↑ / SSIM↑/ LPIPS↓ at $L_m = \frac{1}{2}S_m$				
Model	$S_m = 5$	$S_m = 10$	$S_m = 25$	$S_m = 50$	$S_m = 100$
BasicVSR++^{C}	27.22/0.30/0.571	29.80/0.50/0.430	32.66/0.67/0.223	34.40/0.74/0.119	34.53/0.73/0.119
NafNet32	33.87/**0.84**/0.133	36.09/0.86/0.088	37.74/0.85/0.080	38.25/0.82/0.079	38.73/0.80/0.079
BL-SW	**34.33**/0.80/**0.102**	**36.41**/0.82/**0.078**	**38.44**/0.83/0.069	39.54/0.83/0.065	40.33/0.84/0.062
BL-A&M	31.56/0.70/0.242	34.12/0.79/0.137	36.34/0.85/0.079	37.54/0.87/0.064	38.24/**0.89**/0.059
BL-RNN	32.05/0.75/0.241	35.07/**0.83**/0.128	38.35/**0.88**/**0.067**	**39.97**/**0.89**/**0.055**	**40.71**/0.87/**0.054**

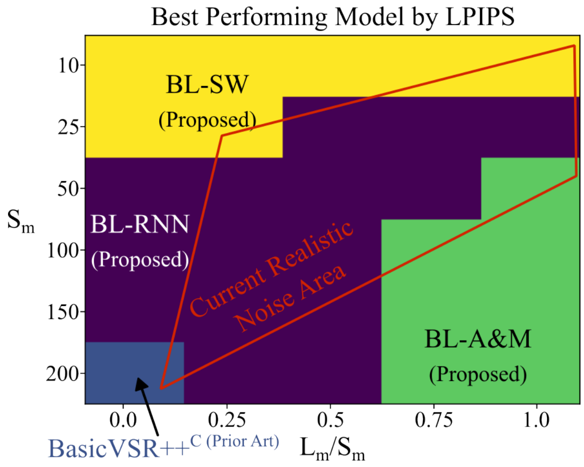


Figure 4.7: **Best Model over S_m and L_m:** This figure shows the model with the lowest LPIPS over a range of different S_m and L_m values. Inside of the red polygon indicates an estimate of realistic noise scenarios that may be seen on current systems in scenes that require denoising.

agrees with the same columns in Table. 4.2 which shows only small changes in LPIPS for the three baseline models. While more photons improve performance, this trend shows there may be a point of diminishing returns when designing drug, device, and denoising model combinations where more photons do not provide large benefits.

Fig. 4.7 provides a summary image showing which model obtains the best LPIPS over a larger range of noise values. We test our models for S_m = [10, 25, 50, 100, 150, 200] and $\frac{L_m}{S_m} = [0, 0.25, 0.5, 0.75, 1]$. BL-SW performs the best at low photon levels where read noise is significant, possibly because BL-SW has access to multiple read noise measurements that are temporally correlated allowing it to better remove the read noise. BL-A&M performs well at high signal and high LLL indicating its robustness to LLL and ability to properly remove it. BL-RNN provides strong performance in the middle noise regions that most closely align with where we expect current FGS systems to operate within for noisy scenes. We find generally, BL-RNN does not over-smooth like the other baselines, likely because it keeps high frequency information to pass on in the recurrent connections. This lack of over-smoothing explains BL-RNN's superior LPIPS score in the middle

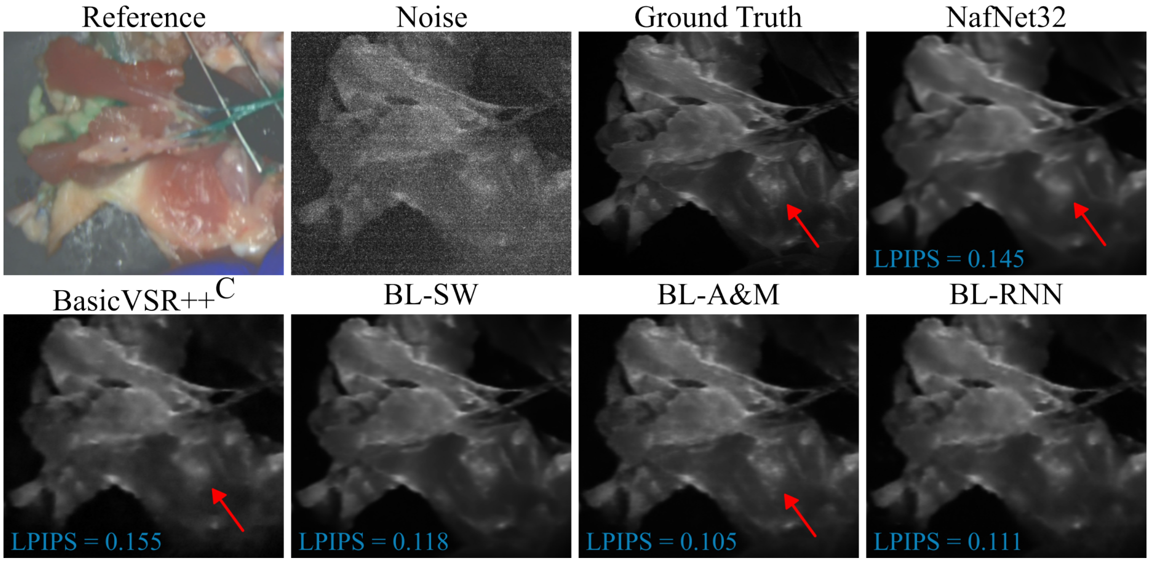


Figure 4.8: **Example Denoised Results:** This figure shows a test case with $S_m = L_m = 50$. In this example our three baseline models all perform reasonable well whereas the comparison models tend to over-smooth (red arrow).

noise regions. Interestingly, BasicVSR++^{C} performs quite well at the test point $S_m = 200, L_m = 0$ which corresponds closely to the noise levels found in general video denoising settings, which reinforces our finding that the changes to the noise in FGS video denoising has strong impacts on performance of models developed solely for conventional video denoising.

Fig. 4.8 shows an example result where the comparison models of NafNet32 and BasicVSR++^{C} tend to oversmooth whereas our baseline models all perform qualitatively very similar and retain small features. We find that in general NafNet32 and BasicVSR++^{C} tend to oversmooth and produce flickering videos.

Robustness to LLL

LLL Robustness is an important property of a strong FGS denoiser for generalizing to real data. We propose to measure LLL robustness through the change in performance as L_m increases, as well as in a new test case where the LLL-PN changes to an new LLL-PN not seen during training.

Table 4.3: **LLL Robustness:** This table shows the results of our LLL robustness tests for 2 different LLL-PNs. The models are trained with f^1_{lll} but tested with either f^1_{lll} or f^2_{lll}. We report the average PSNR, average LPIPS and our robustness measure, $\mathbf{m}_{lll}$ (closer to 0 is better). Best results are bold.

LLL-PN	f^1_{lll}			f^2_{lll}		
Model	$\mathbf{m}_{lll}$ ↑	**$\bar{\text{PSNR}}$** ↑	**$\bar{\text{LPIPS}}$** ↓	$\mathbf{m}_{lll}$ ↑	**$\bar{\text{PSNR}}$** ↑	**$\bar{\text{LPIPS}}$** ↓
BasicVSR++^{C}	-6.64	34.14	0.14	-7.25	33.55	0.14
NafNet32	-8.33	38.64	0.084	-12.74	36.32	0.091
BL-SW	-7.04	**39.77**	0.070	-11.28	37.48	0.070
BL-A&M	**-2.66**	37.41	0.65	**-4.42**	36.67	0.067
BL-RNN	-5.76	39.33	**0.062**	-7.83	**38.28**	**0.059**

Measuring Robustness: We experimentally find that for a fixed S_m the PSNR is roughly linear with L_m. This inspires us to propose using the slope of this line, $\mathbf{m}_{lll}$, in order to measure robustness to LLL levels. A lower value of $\mathbf{m}_{lll}$ indicates that the models performance is robust to changing LLL because it has a small expected change with respect the magnitude of the LLL. Specifically, we assume a model's test PSNR at a fixed S_m can be written as a line,

$$\text{PSNR} \approx \mathbf{m}_{lll}\frac{L_m}{S_m} + b_{lll} \tag{4.8}$$

where b_{lll} is the intercept and $\mathbf{m}_{lll}$ is the slope of the line. We find $\mathbf{m}_{lll}, b_{lll}$ values with least squares regression.

Switching LLL-PN: It is important for a denoising model to be robust to inaccuracies in the LLL-PN used at training time, f^1_{lll}; if f^1_{lll} is not accurate to the true LLL in a scene it will be important for the denoising model to still perform well. We test the robustness of these models by creating a second LLL-PN, f^2_{lll}, by switching the training and testing data used to train f^1_{lll}. We use f^2_{lll} to test the performance of these models under a different LLL-PN than the models were trained on.

We test results for the parameters $S_m = 50$ with $L_m = [0, 12.5, 25, 37.5, 50]$ and for $[f^1_{lll}, f^2_{lll}]$. We report the mean PSNR, and $\mathbf{m}_{lll}$ over this range in Table 4.3. The R^2 value for all linear fits for finding $\mathbf{m}_{lll}$ are above 0.95 indicating a good fit. All models exhibit worse performance under f^2_{lll} indicating a more accurate LLL-PN will be helpful in increasing generalizability of the models. The BL-RNN and BL-A&M changed the least from testing with f^1_{lll} to f^2_{lll} and both had small $\mathbf{m}_{lll}$ indicating these models are the most robust to LLL changes. With BL-A&M providing more robustness at the cost of slightly worse performance when compared to BL-RNN. Likely, BL-A&M and BL-RNN are robust because their inputs contain some processed version of the FV. In the case of BL-A&M the input contains a temporally averaged image so in principle contains averaged results of the LLL potentially reducing the ability for the model to overfit to a specific LLL-PN. Similarly, the BL-RNN uses a denoised output as input allowing it to gain information on the LLL from the previous frame rather than relying solely on the RV and FV, allowing for a multistage approach in estimating the LLL.

What component allows for LLL removal?

The SOTA models for standard video denoising have difficulty in removing the LLL. Here we do an ablation study on our BL-RNN model to test what component is important for LLL removal. We test the following:

- We remove the channel attention modules to test the importance of global information.
- We change the Relu activations to SimpleGate.
- We remove the layer norms.
- We remove the reference video.

We retrain each case using the same training scheme as our full network. We test at $S_m = 50$ and $L_m = [0, 25, 50]$; we report the PSNR difference between the full BL-RNN and the ablation models in Tab. 4.4. We find that the layer norms are the most important aspect of the network and are crucial for stable training, without layer norms we needed to lower the learning rate of the network by a factor of ten for stability, but the network still could not produce strong performance at

Table 4.4: **BL-RNN Ablation Results:** This table shows the PSNR change of our BL-RNN model with different components removed or changed. We test the models at $S_m = 50$ and at three L_m's.

	L_m		
Ablations	0	25	50
No Layer Norm	-22.3	-20.5	-16.7
ReLu → SimpleGate	0.069	-0.464	-0.401
No RV	0.545	-1.992	-1.054
No Channel Attention	-0.553	-1.505	-3.708

convergence. We found that the simplified channel attention module was crucial for denoising at higher L_m indicating that global information is very important for removing the LLL leading to a 3.7dB improvement in PSNR. Similarly, the RV was helpful in removing LLL; although, surprisingly, when there is no LLL the network without no RV performs slightly better but the stark performance drop at higher L_m makes the RV necessary. This behavior could indicate that requiring networks to denoise over a large range of L_m values is difficult and without the RV this network can focus more on denoising the scenario without LLL because its performance in significant LLL will be bounded by its ability to remove it which is challenging without the RV. We also found a slight dip in LLL rmeoval when using SimpleGates instead of ReLus although other components had a larger effect.

Use of Temporal Information

In this experiment, we test the networks abilities to use temporal information in the absence of motion. We conduct this experiment by simulating a no motion scenario by duplicating the first frame of each test video 100 times to create a video with 100 identical frames. We add simulated noise to these videos following our noise model for the case of $S_m = 25, L_m = 12.5$ and $S_m = 10, L_m = 5$. We test the performance of BL-A&M, RNN-BL, and BasicVSR++^{C} to see how well each is able to use temporal information. Because the frames have no motion the BL-A&M

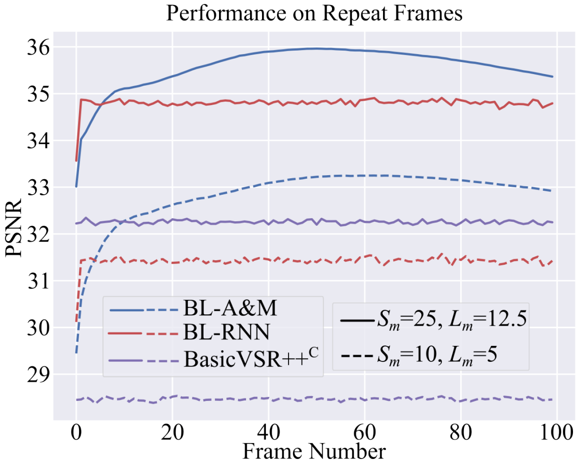


Figure 4.9: **Repeated Frames:** In this experiment, we copy the first frame of each test video 100 times to simulate a scene without motion. We find RNN-BL and BasicVSR++^{C} have constant performance after the first frame, whereas BL-A&M improves for the first 50 frames. After 50 frames the BL-A&M model is outside of its training environment so has a slight decrease in performance.

model will average the prior frames together as input into the downstream network so will be forced to use the temporal information whereas the other models may not learn to use temporal information to its full extent. We plot the average test video performance by frame number in Fig. 4.9.

We find that the recurrent models, RNN-BL and BasicVSR++^{C}, have constant performance across all time points except in the first frame for BL-RNN. This implies these recurrent models do not fully exploit temporal information in the case of no motion. The BL-A&M improves sharply for the first 50 frames before slightly decreasing in performance. The slight decrease in performance is likely due to the fact that the BL-A&M model was not trained with long repeated frame videos so these longer averages are outside of the training environment leading to a performance drop once far outside of examples in the training set. As expected temporal information provides a larger performance improvement for BL-A&M at lower signal levels emphasizing that as models are pushed to work with less noise they will need to make better use of temporal information. However, it seems that current recurrent structures do not necessarily learn to use this information to its

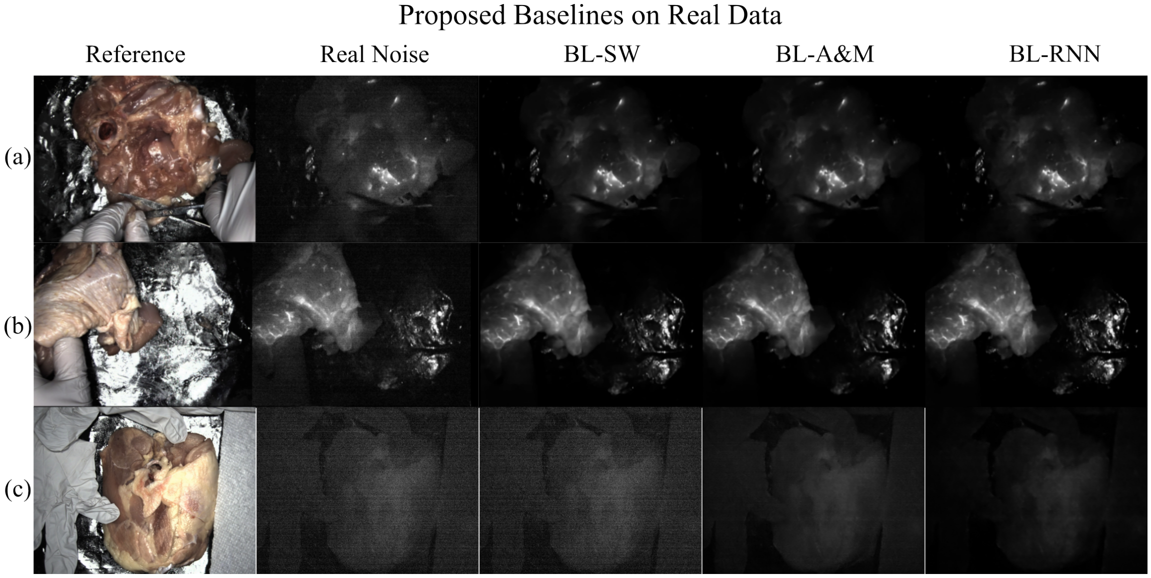


Figure 4.10: **OL-2024 Results:** This figure shows the results of our baseline modes (a-b) on 2 fluorescent scenes from OL-Real. Notice how the hands in (a-b) are correctly removed by our baselines. (c) shows a scene with no fluorescent agent injected, BL-RNN is able to correctly remove most of the LLL in this scenario.

full extent but the performance improvement of BL-A&M.

Interestingly, the BL-A&M model also does not appear to be optimally using the temporal information. It is forced to average temporally, so after 3 frames at the $S_m = 10$ noise level the average should have similar shot noise characteristics as the $S_m = 25$ noise level case, only differing in the read noise. However, the performance at the $S_m = 10$ three frames average is more than 2 PSNR below the $S_m = 25$ first frame performance. This could be because the read noise spatial correlations are important for read noise removal and a simple average of the frames is not a sufficient statistic for this problem leading to reduced efficacy in read noise removal.

Real Test Data

We test our models on OL-Real to determine how well they generalize to real noisy data. Because we do not have ground truth we can not compute metrics on the results; however, we can compare them qualitatively. Example result images are shown in Fig. 4.10(a-b). We find that all our baseline models are able to perform reasonably well on OL-Real. Video results are available in the supplement.

In another test example we input real noisy data that contains no added fluorescence; the results are shown in Fig 4.10(c). For this test, we want a model that correctly finds that no fluorescence is present and outputs a image of all zeros. Note this training example is outside of the training regime of the models so unsurprisingly many of them struggle with this task. Surprisingly, BL-RNN is able to generalize and properly remove most of the LLL present in this image and is the only model that correctly removes the entire LLL coming from the gloves and paper towel on the sides of this scene.

4.5 Future Outlook

Real Time: In this work we did not focus on the real time performance characteristics of these algorithms, which will be a key part of future work. An extensive look at real-time options will need to consider many strategies including: caching, network pruning, half-point precision, image down-sampling, and other hardware optimizations.

Clinical Data: Real clinical data will be a key part of deploying a real clinical system; however, capturing this data is expensive and time consuming so ensuring the correct data is collected is essential. Specifically, it will be important to be able to work with noisy data because ground truth bright fluorescence will not be available. While Noise2Noise (Lehtinen et al. (2018)) and other variants Krull et al. (2019); Batson and Royer (2019) offer promising solutions to training with only noisy data, these solutions assume that the noise is zero-mean and LLL breaks this assumption. An important area of future work will be creating strategies that are able to deal with LLL with a lack of ground truth data; for example, by capturing multiple datasets focused on the LLL and fluorescence separately. Another strategy could try to exploit LLL and RV correlations to find LLL from LLL corrupted FV. We hope the datasets in this paper can provide a starting point for designing and testing these new training and dataset strategies.

Evaluation: Evaluation of different methods is key to understanding performance; however, most current cost functions suitable for denoising are developed for natural images. A FGS specific lost function based on surgeons perceptions will be important to maximizing the value of FGS denoisers. Additionally, tying clinical outcomes to the evaluation will ensure robust and useful performance.

Hardware Optimization: Our work also sets the stage for better posed cost functions in hardware optimization in FGS. Currently, when selecting filters and excitation light sources a number of factors are considered including the tradeoff between LLL and fluorescence yield. It is possible to reduce LLL, but this may also lower the number of fluorescence photons captured by the FV camera. Understanding the exact cost function associated with this tradeoff space is difficult and currently an open problem. By using machine learning algorithm performance over a large simulated noise space it is possible to generate a cost landscape over S_m and L_m based on an objective function (i.e. LPIPS). This cost landscape could be used to rank candidate hardware configurations. This idea is explored in Chapter 5.

4.6 Conclusion

In this work, we considered the difficult problem of FGS video denoising and how it differs from conventional video denoising. Importantly, we developed a realistic noise model that includes complicated read noise, and we proposed a solution for simulating LLL based on the RV. We explored how the unique setting of FGS causes SOTA methods to struggle and proposed strong baseline denoising models to guide future algorithmic development. We also included a number of evaluation settings that models should be compared on to evaluate their performance under different important settings. We hope this work sets the stage for further developments in FGS video denoising methods.

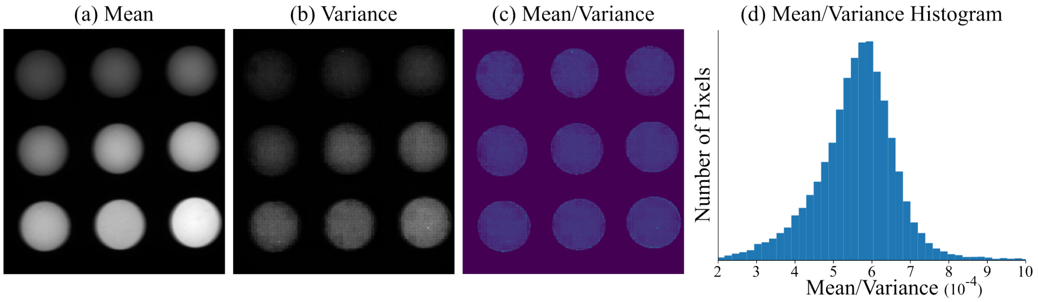


Figure 4.11: **Gain Calibration Images:** This figure shows the (a) mean, (b) variance, and (c) the mean divided by the variance of the Quel calibration phantom from OL-Phantom used to calibrate for our K parameter. (d) shows the histogram of mean divided by variance values for the 9 wells. The mean value of this histogram is our calibrated K value. Notice that the mean divided by variance values are similar in all wells indicating that a Poisson distribution is correct for the brightness level of this phantom.

4.7 Supplementary Section: Calibration and Data Generation Details

Gain (K)

In order to solve for the K in our noise generation model in Eq. 4.1, we do the following calibration procedure. First, we capture a video with 1,830 frames of the Quel calibration phantom Ruiz et al. (2020) (OL-Phantom). The Quel phantom contains 9 wells of varying ICG mimicking fluorescence concentrations. We then take the mean and variance of each pixel across time. We then assume that the Poisson noise term dominates (i.e. read noise is negligible) so our measurements, I, will be Poisson random variables with mean equal to variance. Therefore, K can be estimated as,

$$I = K\mathrm{Pois}(S) \tag{4.9}$$

$$\mathrm{Var}[I] = K^2 S \tag{4.10}$$

$$E[I] = KS \tag{4.11}$$

$$K = \frac{\mathrm{Var}[I]}{E[I]} \tag{4.12}$$

where E is the expectation across time, and Var is the variance across time. We use Eq. 4.12 to estimate a K value for each pixel in a Quel phantom well, we average the estimates to give a final estimate of $K = \frac{1}{1764}$ when I is scaled between 0 and 1.

Quantizer

The OnLume Avata System by default captures 12-bit images and exports 8-bit mp4s. In training, we use a quantizer equivalent to the internal 12-bit quantization to match performance as if the models are run on the system. The quantizer function, Quant, rounds measurements to the nearest multiple of $\frac{1}{2^{12}}$ and clips the values between $[0, 1]$.

Read Noise Scale

The OnLume Avata System by default captures 12 bit images and exports 8 bit mp4s. In order to capture higher bit-depth examples of the read noise, we use the digital gain parameter of the system. Therefore, our read noise samples use a different quantizer than the quantizer used in calibration of the K parameter. We match the scales of the read noise and shot noise terms by dividing the sampled read noise frames by R_m, we sample R_m values between $[4, 8]$ during training and use $R_m = 6.0$ during testing which is calibrated so our scaled read noise frames match the digital gain used to calibrate K.

Param	Min	Max	Test
$\frac{1}{K}$	1200	2400	1763.5
R_m	4	8	6
S_m	10	$\frac{1}{2K}$	Varies
L_m	0	S_m	Varies

Table 4.5: This table shows the camera values used in training and testing.

Data Augmentation

In order to augment our data and not over fit to a given set of camera parameters we randomize different camera parameters over a small range, while testing with our calibrated parameters. Table 4.5 shows the camera values used in training and testing.

LLL-PN Details

Network and Training Details: For our LLL-PN, $f_{\text{LLL-PN}}(I;\theta)$, we use a lightweight version of NafNet Chen et al. (2022). We use the OL-LLL dataset for training which contains video of a mock chicken thigh surgery without fluorescent dyes, therefore the fluorescent frames only contain the LLL term (L^{LL}). We train our LLL-PN parameters, θ, with the following loss function,

$$l_p(I_{\text{ref}}, I_{\text{FL}}, \theta) = ||f_{\text{LLL-PN}}(I_{\text{ref}};\theta) - I_{\text{FL}}||_p \tag{4.13}$$

where I_{ref} is a reference frame, I_{FL} is a fluorescent frame, and $||\cdot||_p$ is the l_p-norm we try $p = 1, 2$. We train our network with the Adam optimizer Kingma and Ba (2015) ($\beta_1 = 0.9, \beta_2 = 0.999$) with initial learning rate of 10^{-4} and batch size of 10 for 100 epochs with a cosine annealing learning rate decay. I_{ref} and I_{FL} are the reference and fluorescence frames respectively.

Choice of loss: We try both the l_1 and l_2 loss to train our LLL-PN. Because we are training in the Noise2Noise paradigm, it is important to consider the sources of noise in our training data. Because our read noise exhibits a strong flicker noise that is non-zero mean, we expect the l_1 loss to work better as it is a median seeking

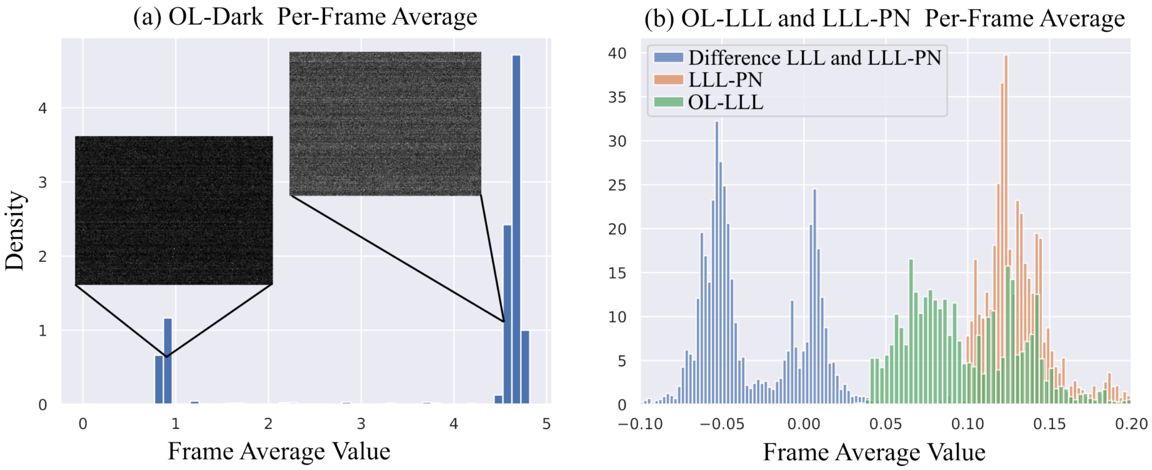


Figure 4.12: **Flicker Noise:** (a) shows the histogram of the per-frame average values of the OL-Dark dataset. The read noise pattern of our camera sensor exhibits strong flicker noise leading to a bimodal distribution, one example from each peak is shown. (b) shows the per-frame average of the OL-LLL dataset (green), the LLL-PN (orange), and the difference between the two (blue). Due to the bimodal flicker noise our LLL-PN predicts the "high" mode of the flicker; however, we are able to obtain the bimodal distribution when subtracting out our background indicating strong performance with subtracting out the LLL and leaving the read noise.

loss whereas l_2 is an average seeking loss Lehtinen et al. (2018) and will predict an average of the flicker which is not dependent on the LLL. We evaluate l_1 and l_2 errors of the networks trained using both training losses; the results are shown in Table 4.6. We find that the l_1 training loss provides better performance so we use this network as our final LLL-PN. Note that the calculated error metrics will include noise energy from read noise and Poisson noise from the background, so the minimum error will never reach 0 as our LLL-PN can only remove the bias term associated with the background.

Flicker noise: We find the median seeking L_1 loss performs better because our read noise exhibits strong flicker noise producing a bimodal noise distribution with different additive bias terms, shown in Fig. 4.12(a). By using an L_1 loss, the LLL-PN predicts the additive bias term that is most present in the training data (the brighter flicker). Fig. 4.12 shows histograms of the OL-LLL test set per-frame average image values for the noisy LLL, $f_{LLL}(R^{\nu}_{t})$, and the difference between the predicted and noisy values. Notice that, as expected, the bimodal distribution of the flicker noise is much more visible in the difference histogram; although, the bimodal distributions are wider which can be accounted for by shot noise and any mistakes made by the LLL-PN.

Training Loss	l_1	l_2	Fluorescent Frame Norms
Errors	0.0621/0.0762	0.0643/0.0785	0.100/0.127

Table 4.6: **LLL-PN loss function comparison:** we compare the errors (l_1/l_2) of our LLL-PN for different loss functions. We also include the noisy fluorescent frame l_1 and l_2 norms to show how much background signal is subtracted.

4.8 Supplementary Section: Training Parameters:

The values we use for training all models are shown in Tab. 4.7, full config files will be available with the code. One key challenge in video denoising is balancing training time with the number of parameters. We find this is even more true in FGS denoising because of the large noise parameter space that needs to be sweeped across requiring many parameters to deal with this large number of potential noise levels. Our BL-RNN model is able to train quickly to convergence while maintaining a large number of parameters which is not true for other models such as BasicVSR++^{C}.

Table 4.7: **Training Details:** In each epoch we go through every 100 frame non-overlapping partition (768 sets) in the training set, for each 100-frame set in a batch we pull out Num T frames for the full batch used in one iteration of training. Total time is the amount of hours it takes to train the model with a single A100 GPU. Iterations is the number of gradient decent steps done in total, and total seen is the number of total frames seen during training. BL-SW and FastDVDnet are trained using 5 frames from every video but only denoise the last frame of this group. To train Bl-A&M we first train the NafNet32 backbone, the values after the + sign in the table indicate the where the pre-training parameters of NafNet in Bl-A&M for differ from the NafNet32 full training.

Model	Seconds/epoch	Epochs	Num T	Batch	Total time (hr)	Iterations	Images Seen
BL-RNN	90	3k	4	15	75	153k	9.2M
BL-SW	90	6k	5 (1)	14	150	324k	22.7M
Bl-A&M	800	1.2k+1.6k	5	8	267+69	111k+302k	4.5M+12.1M
NafNet32	155	3k	10	4	129	567k	22.7M
BasicVSR++^{C}	1100	1000	30	2	306	377k	22.7M
FastDVDnet^{C}	90	6k	5 (1)	14	150	324k	22.7M
OFDVDnet^{C}	950	1.2k	6	8	316.67	114k	5.4M

4.9 Supplementary Section: Causal BasicVSR++ on Davis

We retrain our modified causal BasicVSR++ model on the conventional video denoising Davis dataset following the training scheme from the BasicVSR++ paper Chan et al. (2022a). We also train NafNet32 and NafNet64 on this dataset to compare the results with our choice of image denoiser baseline. Our quantitative results are summarized in Table. 4.8. We find that converting BasicVSR++ to a causal network greatly hurts the performance at large noise levels.

Table 4.8: **NafNet and Causal BasicVSR++ on DAVIS:** Here we report the standard quantitative results (PSNR/SSIM) for video denoising on the DAVIS dataset Pont-Tuset et al. (2017) for our trained models: Causal BasicVSR++$_2$, NafNet32 and NafNet64.

	VBM4D	FastDVDnet	BasicVSR++$_2$	Causal BasicVSR++$_2$	NafNet32	NafNet64
$\sigma = 10$	37.58/-	38.71/0.9672	40.97/0.9786	39.57/0.9721	36.32/0.9455	38.30/0.9623
$\sigma = 20$	33.88/-	35.77/0.9405	38.58/0.966	36.72/ 0.9516	33.14/0.8998	34.86/0.9267
$\sigma = 30$	31.65/-	34.04/0.9167	37.14/0.9560	35.02/0.9328	31.19/0.8547	32.96/0.8957
$\sigma = 40$	30.05/-	32.82/0.8949	36.06/0.9459	33.76/0.9143	29.76/0.8081	31.67/0.8689
$\sigma = 50$	28.8/-	31.86/0.8747	35.18/0.9358	32.74/0.8951	28.55/0.7545	30.70/0.8460
Average	32.39/-	34.64/0.9188	37.58/0.9566	35.56/0.9332	31.79/0.8526	33.67/0.8999

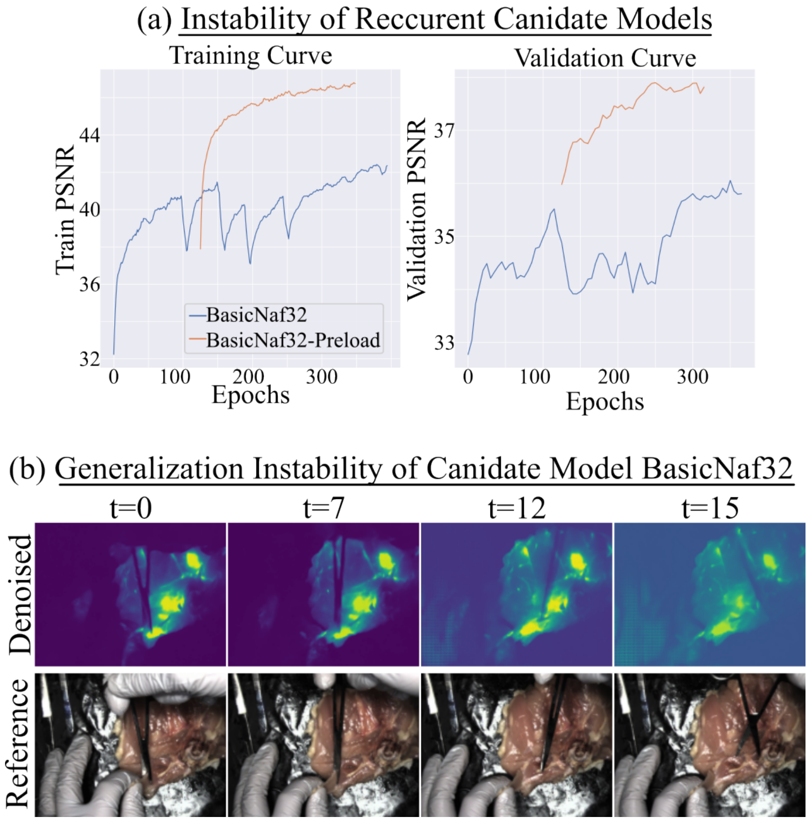


Figure 4.13: **Instability of Recurrent Candidate Models:** (a) Training and validation curves for the BasicNaf32 and BasicNaf32-Preload candidate models. The BasicNaf32 model exhibits training instability after 100 epochs of training (60 hours). While our BasicNaf32-Preload exhibits smoother training. Note BasicNaf32-Preload curves are offset by the pre-loaded image only model training time in equivalent BasicNaf32 epochs. (b) BasicNaf32-Preload has trouble generalizing to the full resolution test set that results in artifacts for later recurrent frames.

4.10 Supplementary Section: Training Instability of Recurrent Candidate Models

We found that training recurrent networks can lead to instability even late in the training process after many hours of training. This makes fast iteration of recurrent structures challenging because time is wasted on tuning the learning rate or other hyper parameters. One of our candidate models was a combination of NafNet and BasicVSR++ called BasicNaf32, which uses the Unet structure of NafNet but the recurrent temporal feature propagation of BasicVSR++. We found that one hyperparameter combination of BasicNaf32 becomes unstable after about 60 hours of training, the training curve is shown in Fig. 4.13(a). A preloading strategy is used to train BasicNaf32-Preload: this strategy first trains the network for 1600 epochs on only images and provides stability to the training process. However, a surprising generalization problem occurs in both models where they are unable to generalize from the trained 256 by 256 images to the full sized 768 by 1024 test images. This instability occurs after about 10 frames in this recurrent structure as shown in Fig. 4.13(b). We found this unstable behavior to occur often within combinations of NafNet and BasicVSR++ because of this finding we chose to use a simpler recurrent structure for our BL-RNN.

4.11 Supplementary Section: No Reference Frame Qualitative Result

We find that the RV is key in removing LLL; an example is shown in Fig 4.14. In this example, the surgeons hand is clearly visible in the BL-RNN network that does not have access to the reference frames whereas the BL-RNN network that uses the reference frames is able to better remove the hand.

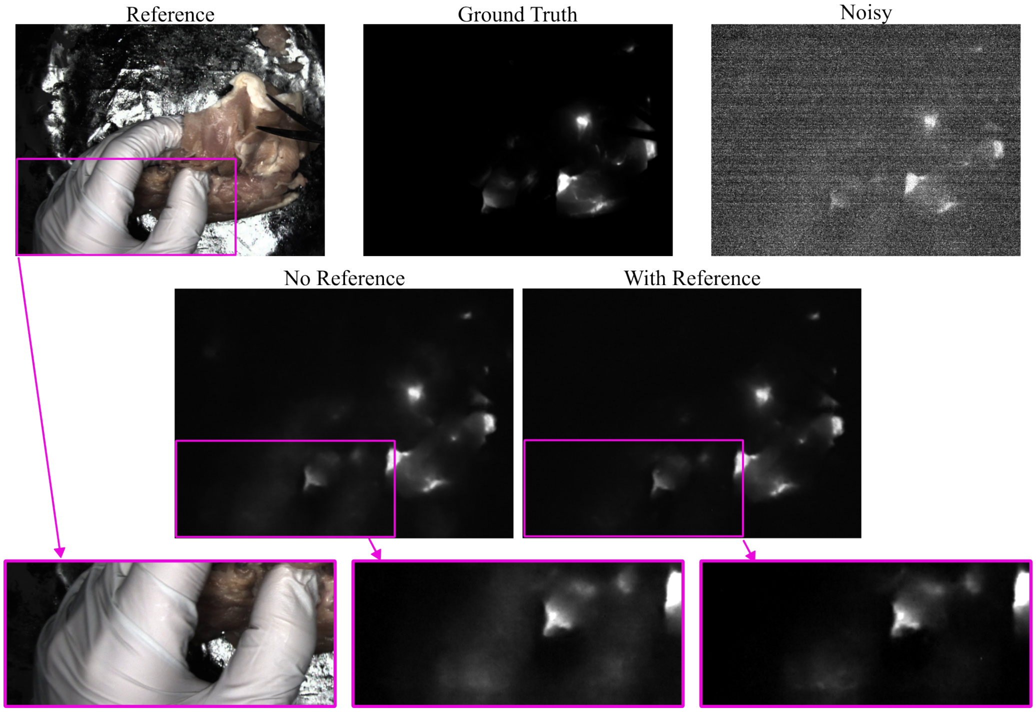


Figure 4.14: **No Reference:** This figure shows a case where the reference frame is key in the denoising process to properly remove the LLL associated with a hand. The noise parameters used for this example are $S_m = L_m = 25$.

5 SOFTWARE DEFINED OPTIMIZATION OF FGS HARDWARE

Hardware choices in general lead to complex trade-off spaces that often do not have well-posed cost functions to evaluate performance of a set of hardware. Instead, engineers and device designers are required to use intuition or heuristics to decide on hardware choices. For example, in fluorescence guided surgery (FGS) hardware is key in designing spectral bands for detection of fluorescence contrast agents. A trade-off exists however between generating and collecting large amounts of fluorescence, S_m, and collection of detrimental laser leakage light (LLL), L_m. LLL is non-specific photons that represents a spatially varying background term (see Chapter 4). In FGS, fluorescence hardware choices can be abstracted into two spectra, the excitation light spectra $f_{ex}(\lambda)$ represents the spectral content of the excitation light, and the emission filter spectra $f_{em}(\lambda)$ which represents the collection efficiency of the system as a function of wavelength, λ. Hardware choices such as the laser bandwidth, center wavelength, filter cutoffs, or other optical elements are used to control $f_{ex}(\lambda)$ and $f_{em}(\lambda)$. Generally, device designers will look at $f_{ex}(\lambda)$ and $f_{em}(\lambda)$ along with the contrast agent spectra to decide hardware. Fig. 5.1 shows an example of the important spectra for this problem, from this data it is possible to extract estimates for the fluorescence generation and LLL throughput. To maximize fluorescence generation it is desirable to have f_{ex} align with the ICG excitation spectra which represents how sensitive the fluorescent agent is at each wavelength. And f_{em} should capture as much of the ICG emission as possible; however, any overlap in f_{ex} and f_{em} represents LLL which needs to be minimized. In general this creates a trade-off between fluorescence signal collection (S_m) and unwanted LLL collection (L_m); however, the exact trade-off between these two quantities is not well understood and is considered a major open problem in FGS (DSouza et al. (2016); Olson et al. (2019); Pogue et al. (2023)). Generally, these are considered a hardware problem; however, by using software denoisers that are trained over a large range of S_m and L_m values it is possible to understand this trade-off space. We propose a optimization cost function based on FGS video denoising network

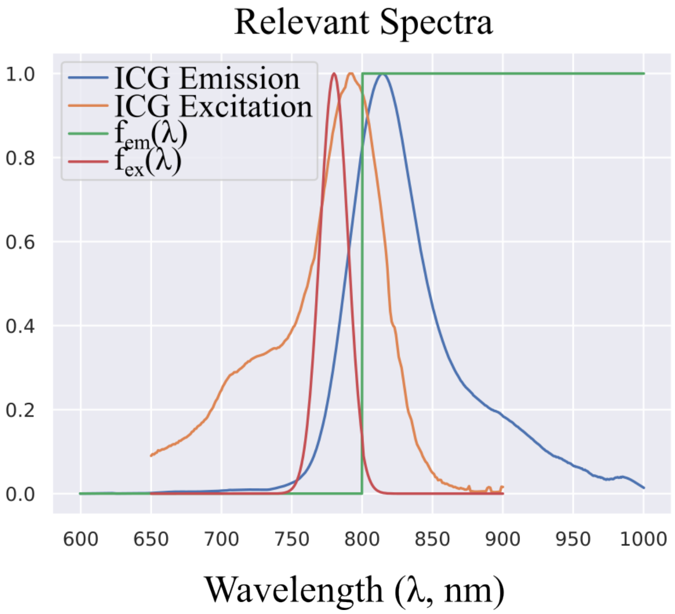


Figure 5.1: **Spectra:** This figure shows 4 important spectra in designing FGS systems. The ICG emission and excitation spectra are functions of the drug or contrast agent used for fluorescence in this case the example spectra are form ICG in albumin. The imaging system also defines 2 spectra through hardware the excitation, $f_{ex}(\lambda)$, is defined by the light source while the emission, $f_{em}(\lambda)$, is defined by optical filters.

performance to evaluate the trade-off of L_m and S_m that can be used to optimize FGS hardware.

Our goal is to create a cost function for the following optimization,

$$\min_{f_{em}(\lambda), f_{ex}(\lambda)} \text{Cost}(f_{em}, f_{ex}, C_{fl}) \tag{5.1}$$

where the optimization runs over a set of hardware components that create the excitation spectrum $f_{ex}(\lambda)$, emission filter spectrum $f_{em}(\lambda)$, and C_{fl} represents a target concentration of fluorescence that determines the fluorescence brightness. The general idea of our approach is to use a video denoiser's (Chapter 4) performance on a test set over a range of L_m and S_m values to define the cost function; then we use a physical model of hardware, and the fluorescence generation process to map hardware choices onto the L_m, S_m space so they can be used as input into our learned cost function.

5.1 Physical Parameters

There are a number of physical parameters required to fully model a fluorescence excitation, emission, and collection pathway. Here we list many of them with a short summary of their physical meaning. We will go into more details on some of these quantities in the following subsections with the goal of obtaining S_m and L_m as functions of other quantities.

1. S_m is the number of detected signal fluorescence photons.

2. L_m is the number of detected laser leakage photons.

3. $f_{ex}(\lambda_{ex})$ is a function over wavelength representing the spectral content of the excitation light source. We assume the following, $\int f_{ex} \leqslant 1$ and $f_{ex} \geqslant 0$.

4. $f_{em}(\lambda)$ is the spectral efficiency of the photon collection pipeline. If a photon of wavelength λ enters the optical path, $f_{em}(\lambda)$ represents the probability of detecting that photon at the fluorescence camera. Note this includes the detection efficiency of the camera. We assume $0 \leqslant f_{em} \leqslant 1$.

5. A_{Fl} represents the linear spectral fluorescence operator that takes as input an excitation spectrum and outputs an emission spectrum. The discrete version of this operator is often called the Emission Excitation Matrix.

6. A_{LL} represents the linear spectral operator accounting for the reflectivity of tissue.

7. $0 \leqslant I_{ex}$ this scalar represents the excitation light power, in units of photons per second.

8. T is the exposure time for a frame of video of the fluorescent camera in seconds.

9. $0 \leqslant C_{fl}$ represents a scale factor on the fluorescence emission that accounts for the expected concentration of the fluorescence agent relative to the photon yield used to define A_{Fl}. This needs to be experimentally calibrated for and depends depends on the biochemistry of a contrast agent. This parameter also acts as a catch-all to other properties effecting fluorescence brightness, such as losses to scattering in depth. We assume fluorescence is operating in the linear regime which is true in most reasonable operating cases.

Note the scalars, I_{ex} and C_{fl}, should be seen from the prospective of a expected scene point in the image space of the camera. For example, the laser power I_{ex} will include distance fall-offs and photon collection efficiency of the imaging lens, i.e. it represents the number of photons per second hitting a pixel imaging a isotropic diffuse white reflector at a given operating distance where the imaging system has a unity spectral pass band (i.e. no filters). This means that changes in lenses and other collection optics will affect the bounds of attainable values for I_{ex}. While exact calibration may be difficult for some of these parameters, without changing optical filters, relative measurements are easy, for example, C_{fl} is proportional to fluorescence concentration.

5.2 A Learned Cost Function

We propose using a cost function based on video denoising network performances in Eq. 5.1. In chapter 4, we trained a number of FGS video denoising models over a range of S_m and L_m parameters. We propose to use the following as the cost function,

$$\mathrm{Cost}(S_m, L_m) = \min_{m \in M} \mathrm{LPIPS}(m, S_m, L_m) \tag{5.2}$$

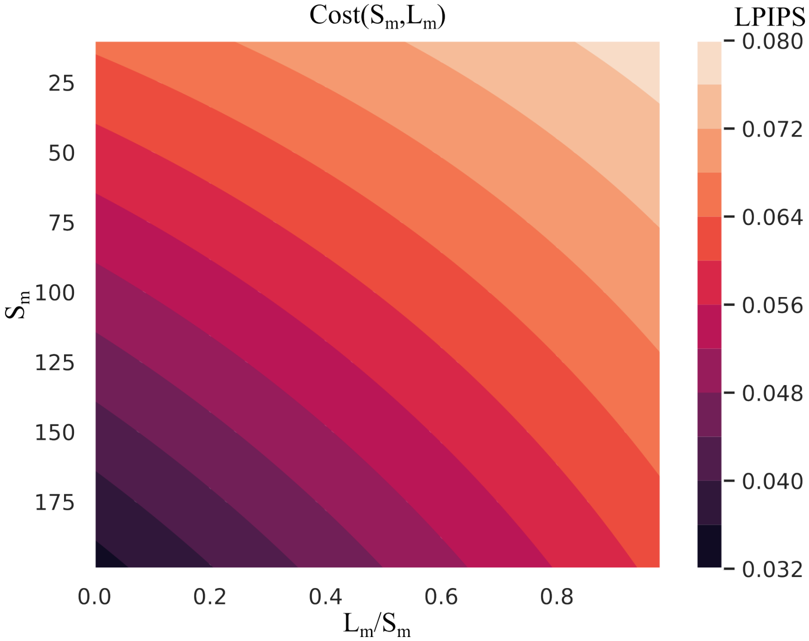


Figure 5.2: **Cost Landscape:** This figure shows contours of the interpolated cost landscape we obtain from the best LPIPS of our trained video denoising models. Lower LPIPS is better.

where $\mathtt{LPIPS}(m, S_m, L_m)$ is the LPIPS obtained on the test set of model m at noise level (S_m, L_m), and M is the set of all models. LPIPS Zhang et al. (2018) is a deep learning based perceptual cost function, that is trained on human perception of deformations and noise. Due to computational cost we use a sparse sampling of $\mathtt{LPIPS}(m, S_m, L_m)$ over (S_m, L_m) along with linear interpolation to approximate this cost landscape. We evaluate our models for $S_m = [10, 25, 50, 100, 150, 200]$ and $\frac{L_m}{S_m} = [0, 0.25, 0.5, 0.75, 1.0]$ which also defines the support of the cost functions we find in this work. Note it is easy to expand the cost function to larger support if needed given enough compute time. Interpolating the LPIPS values, we obtain the cost function shown in Fig. 5.2. Importantly, this cost function is non-linear; however, we find it has a simple parametric approximation.

Parametric Cost Function

We find that $\mathrm{Cost}(S_m, L_m)$ has a simple parametric approximation of the form,

$$\mathrm{Cost}(S_m, L_m) \approx C(S_m, L_m) \tag{5.3}$$

$$C(S_m, L_m) = -c_1 S_m + c_2 L_m + c_3 \frac{L_m}{S_m} + c_4 \tag{5.4}$$

where c_i are constants we fit with least squares, note we have chosen the signs on c_i such that $c_i > 0$ for the values we find with fitting. By parameterize the cost function we are able to investigate properties, e.g. gradients, and we are able to evaluate cost function points quickly.

The values we obtain for these c_i constants are given in Table. 5.1, the squared error between $\mathrm{Cost}(S_m, L_m)$ and $C(S_m, L_m)$ is less than 10^{-30}. Later we need to find S_m and L_m as a function of physical parameters to be able to evaluate our optimization for different hardware configurations.

Gradient of $C(S_m, L_m)$**:** Here we calculate the gradient of $C(S_m, L_m)$ to ensure our parametric form provides intuitive results. We first compute the symbolic solution then plug in our found parameters for c_i. The partial derivatives are given by,

$$\frac{\partial}{\partial S_m} C(S_m, L_m) = -c_1 - c_3 \frac{L_m}{S_m^2} \tag{5.5}$$

$$= -1.611e^{-4} - 0.01285 \frac{L_m}{S_m^2} \tag{5.6}$$

$$\leqslant 0 \tag{5.7}$$

where the last line follows from $0 \leqslant L_m, S_m$. The other partial is given by,

Table 5.1: This table shows our resulting constant values frond from fitting Eq. 5.4 to our interpolated cost function.

c_1	$1.611e^{-4}$
c_2	$7.264e^{-5}$
c_3	0.01285
c_4	0.06637

$$\begin{aligned} \frac{\partial}{\partial L_m} C(S_m, L_m) &= c_2 + \frac{c_3}{S_m} && (5.8) \\ &= 7.264e^{-5} + 0.01285 \frac{L_m}{S_m} && (5.9) \\ &\geqslant 0 && (5.10) \end{aligned}$$

where, again, we use $0 \leqslant L_m, S_m$. As expected increasing S_m or decreasing L_m leads to lower LPIPS and better performance.

The Spectral Fluorescence Operator and S_m

Next we compute S_m as function of $f_{ex}(\lambda_{ex})$. The Spectral Fluorescence Operator (SFO), A_{Fl}, takes as input a excitation spectra, $f_{ex}(\lambda_{ex})$, and outputs the fluorescence emission spectra, $g_{em}(\lambda_{em})$. The SFO describes the process of converting an excitation photon to a spectrally shifted emission photon. Generally, the SFO is measured experimentally using fluorescence spectroscopy. Mathematically, this is a linear operator and can be written as,

$$\begin{aligned} g_{em}(\lambda_{em}) &= A_{Fl} f_{ex}(\lambda_{ex}) && (5.11) \\ &= \int k_{FL}(\lambda_{ex}, \lambda_{em}) f_{ex}(\lambda_{ex}) d\lambda_{ex} && (5.12) \end{aligned}$$

where $k_{FL}(\lambda_{ex}, \lambda_{em})$ is the spectral varying impulse response of a fluorophore.

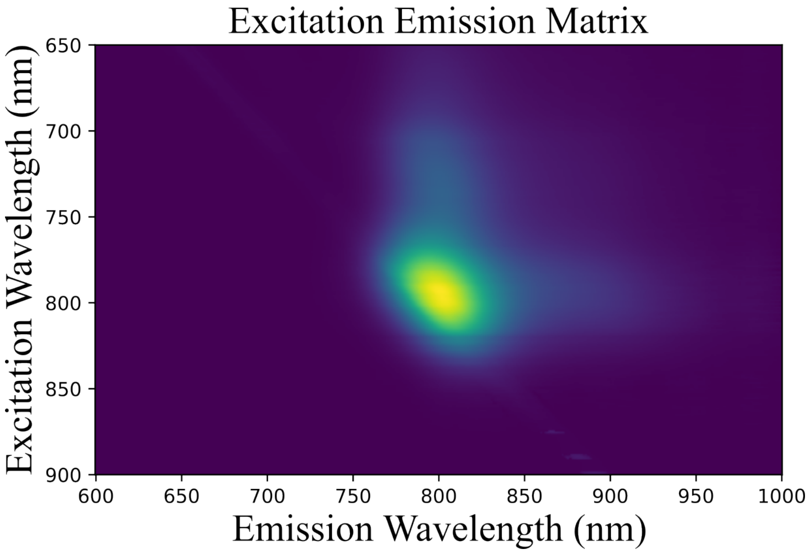


Figure 5.3: **Excitation Emission Matrix:** This shows the EEM of ICG in albumin. The EEM serves as a discrete aproximiation for the continuous SFO. Thanks to Quel Imaging Ruiz et al. (2020) for providing this EEM.

Experimentally, $k_{FL}(\lambda_{ex}, \lambda_{em})$ is measured by sending in a succession of narrow bands of excitation light and measuring the output spectrum. The discrete experimentally measured approximation of this operator is often called the Excitation-Emission Matrix (EEM), Fig. 5.3 shows an example of an EEM for ICG in Albumin courtesy of Quel Imaging.

Commonly, the fluorescence spectra is often used to describe A_{Fl} but often by dropping the one of the spectral dependencies of this operator. The fluorescence spectra is decomposed into the fluorescence emission, $a_{ex}(\lambda_{ex})$, and excitation spectra, $a_{em}(\lambda_{em})$. If only these spectra are available, we may approximate A_{Fl} as $\tilde{A}_{Fl}$,

$$\tilde{A}_{Fl} f_{ex}(\lambda_{ex}) = a_{em}(\lambda_{em}) \int a_{ex}(\lambda_{ex}) f_{ex}(\lambda_{ex}) d\lambda_{ex} \tag{5.13}$$

Notice how this approximation has a constant emission spectra that does not depend on the excitation spectra. While simpler to measure and compute, depending on the application and available hardware shifts in spectra may be relevant requiring use of the full A_{Fl} operator.

Finding S_m: Now we are ready to find S_m as a function of $f_{ex}(\lambda)$ and $f_{em}(\lambda)$. S_m can be written as,

$$S_m = TI_{ex}C_{fl}\int f_{em}(\lambda_{em})g_{em}(\lambda_{em})d\lambda_{em} \tag{5.14}$$

$$= TI_{ex}C_{fl}\langle f_{em}, g_{em}\rangle \tag{5.15}$$

where $\langle\cdot,\cdot\rangle$ is the standard inner product.

The Reflectance Operator and L_m

Similarly, the Reflectance operator, $A_{LL}(\lambda_{em})$, operates on the same domain as $A_{Fl}(\lambda_{em})$ but has a simpler form. It represents the excitation light being reflected off of the tissue that will later effect the LLL. It is simply a multiplication with the excitation spectra and the reflectance spectra of the tissue,$k_{ll}(\lambda)$,

$$L(\lambda) = A_{LL}f_{ex}(\lambda) \tag{5.16}$$

$$= k_{ll}(\lambda)f_{ex}(\lambda) \tag{5.17}$$

where $L(\lambda)$ is the reflected excitation light.

Finding L_m: Therefore, L_m can be written as,

$$L_m = TI_{ex}\int f_{em}(\lambda)L(\lambda)d\lambda. \tag{5.18}$$

Unfortunately, $k_{ll}(\lambda)$ is not necessarily easily available. So instead we bound $k_{ll}(\lambda) \leqslant 1$, giving,

$$L_m \leqslant TI_{ex} \int f_{em}(\lambda) f_{ex}(\lambda) d\lambda. \tag{5.19}$$

$$= TI_{ex} \langle f_{em}, f_{ex} \rangle \tag{5.20}$$

where $\langle \cdot, \cdot \rangle$ is the standard inner product. We can use this bound as a worse case L_m in equation $C(S_m, L_m)$ because because $\frac{\partial}{\partial L_m} C(S_m, L_m) > 0$.

5.3 Expanding the Approximated Cost Function

We now combine our previous results to allow optimization of $C(S_m, L_m)$. Plugging in Eq. 5.20 and Eq. 5.20 into Eq. 5.4,

$$C(S_m, L_m) = C(\Theta) = -c_1 TI_{ex} C_{fl} P_S + c_2 TI_{ex} P_L + c_3 \frac{P_L}{C_{fl} P_S} + c_4 \tag{5.21}$$

$$P_S = \langle f_{em}, A_{Fl} f_{ex} \rangle \tag{5.22}$$

$$P_L = \langle f_{em}, f_{ex} \rangle \tag{5.23}$$

where P_S and P_L represent the fraction of photons converted to detected fluorescence signal and LLL photons, respectively. Θ represents the physical parameters used to calculate the cost function.

We note the conventional wisdom for optimizing FGS hardware looks at both P_S and P_L as the primary metrics for deciding on hardware. However, from Eq. 5.22 we see that both the target concentration and the laser power have an effect on what should be chosen. Eq. 5.22 also ties together P_S and P_L into a signal number of image quality that can be optimized over.

Defining the Noise Floor

The noise floor of an FGS system is often understood as the point where the fluorescence signal is lost within noise in the system either read noise or LLL noise. Generally, this point is found by humans interpreting data or through phantom experiments and it generally does not account for down stream systems ability to remove noise. We propose using the gradients of $C(\theta)$ to define the noise floor of a system and it's down stream processing.

Intuitively, the noise floor occurs at the fluorescence concentration where no more information from the scene can be extracted even with increasing exposure times. From this intuition we propose defining the noise floor, C_{nf}, as the $C_{fl} = C_{nf}$ such that $\frac{\partial C}{\partial T} = 0$. We compute this point as follows,

$$\frac{\partial C}{\partial T} = -c_1 I_{ex} C_{fl} P_s + c_2 I_{ex} P_L = 0 \tag{5.24}$$

$$\rightarrow C_{nf} = \frac{c_2 P_L}{c_1 P_s}. \tag{5.25}$$

And from Eq. 5.24 it is easy to see that decreasing C_{fl} leads to a larger gradient and worse performance, i.e. $C_{fl} < C_{nf} \rightarrow \frac{\partial C}{\partial C_{fl}} \geqslant 0$.

We see that C_{nf} increases with P_L and decreases with P_s as expected. Notice C_{nf} can be decomposed into two terms, one term is hardware defined, $\frac{P_L}{P_s}$, and one term is defined $\frac{c_2}{c_1}$ through processing which incorporates the unavoidable noise in this problem. In our experimental case we find that C_{nf} lands outside the training conditions of our models. Note,

$$\frac{L_m}{S_m} = \frac{P_L}{C_{nf} P_s} \tag{5.26}$$

$$= \frac{c_1}{c_2} \tag{5.27}$$

$$= 2.2 \tag{5.28}$$

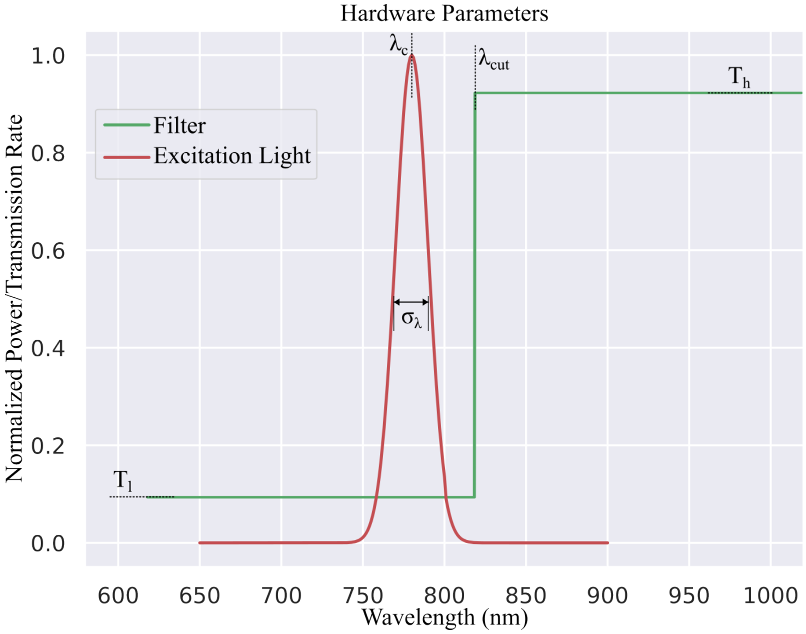


Figure 5.4: **Hardware Parameters:** This figure shows an overview of the 5 hardware parameters we consider.

which lies outside of the support of $C(\Theta)$.

We note that our training of these models are limited in support along the noise parameters, we find that our C_{nf} for our fitted values lies outside of the support of our cost function indicating we have not tested at a level that would be considered a noise floor under our definition. We postulate that with more training data and more training time over larger noise parameters $\frac{\partial C}{\partial C_{fl}}$ will converge to a equation that can give the true noise floor if one exists. We also note our definition could be changed to identify when the gradient equals a small number rather than 0 which may give a more robust measure if a level set along T does not exist.

Parametrization of f_{em} and f_{ex}

At this point it is possible to select optimal hardware from a catalog of options by using Eq. 5.22 over a discrete set of options. f_{ex} and f_{em} are defined by light source and filter choices, I_{ex} is defined by the light source power, T is constrained by the need for real time measurement, A_{fl} is defined by the contrast agent, and a useful range for C_{fl} can be found experimentally. While this is useful, we also seek some intuitive understanding of this problem so next we approximate a number of hardware choices with parameterized functions to explore this problem. Fig. 5.4 shows an overview of the parameters in our simplified hardware models. In out simplified model, there is a excitation light source possibly followed by a excitation filter that cleans up the light's spectral content, for example by cutting off long tails that would interfere in the emission band. The emission band is made up of a single emission filter. These 3 pieces of hardware with our simplified parameters define an 8 dimensional space that our hardware configurations exist in.

Filters

Both the excitation and emission path may contain filters. Here we use a simple square function as our filter model with three parameters: the cutoff wavelength λ_{cut}, the low wavelengths transmission, T_l, and the high wavelengths transmission, T_h. With these parameters our filter response, $T_f(\lambda; \lambda_{cut}, T_l, T_h)$ is given by,

$$T_f(\lambda; \lambda_{cut}, T_l, T_h) = \begin{cases} T_l & \lambda \leqslant \lambda_{cut} \\ T_h & \lambda_{cut} < \lambda. \end{cases} \tag{5.29}$$

While more complex filter models exist that include the spectral rising/falling edge width, for simplicity we do not consider these effects in this work.

Excitation Light

We model the excitation light, $I(\lambda, \lambda_c, \sigma_\lambda)$ as a Gaussian distribution with center wavelength λ_c and a spectral standard deviation σ_λ,

$$I(\lambda; \lambda_c, \sigma_\lambda) = \frac{1}{\sigma_\lambda \sqrt{2\pi}} e^{-\frac{1}{2}(\frac{\lambda - \lambda_c}{\sigma_\lambda})^2} \tag{5.30}$$

f_{em} and f_{ex}

The emission side only has one filter so,

$$f_{em}(\lambda) = T_f(\lambda; \lambda_c^{em}, T_l^{em}, T_h^{em}) \tag{5.31}$$

where $\lambda_c^{em}, T_l^{em}, T_h^{em}$ are the filter parameters for the emission filter.

The excitation path has the excitation light followed by a filter,

$$f_{ex}(\lambda) = T_f(\lambda; \lambda_c^{ex}, T_l^{ex}, T_h^{ex}) I(\lambda; \lambda_c^{ex}, \sigma_\lambda^{ex}) \tag{5.32}$$

where $\lambda_c^{ex}, T_l^{ex}, T_h^{ex}$ are the excitation filter parameters and λ_c^{ex}, σ_λ^{ex} are the excitation light parameters.

5.4 Results

We are now able to produce a number of results using our hardware parameterization and cost function to better understand how hardware effects our denoised results. In general in hardware design, price and performance are balanced to find the best hardware choices. Our goal will be to explore the implications of our cost function on hardware design. Note in our cost function (Eq. 5.22) once we find f_{em} and f_{ex}, we are left with a two dimensional cost function of C_{fl}, the brightness of the fluorescent object, and TI_{ex}, the exposure time multiplied by the laser power. We find that some hardware configuration will have different optimal values depending on C_{fl} and TI_{ex}. Because we have 8 hardware parameters to test for filter and laser design rather than searching for optimal configurations, which may not be available for purchase, we investigate trends that may be of interest and are worth considering when choosing hardware configurations.

Filters

The simplest setup consists of a laser excitation source and an emission filter with no excitation filter. In this subsection we investigate the emission filter settings by fixing

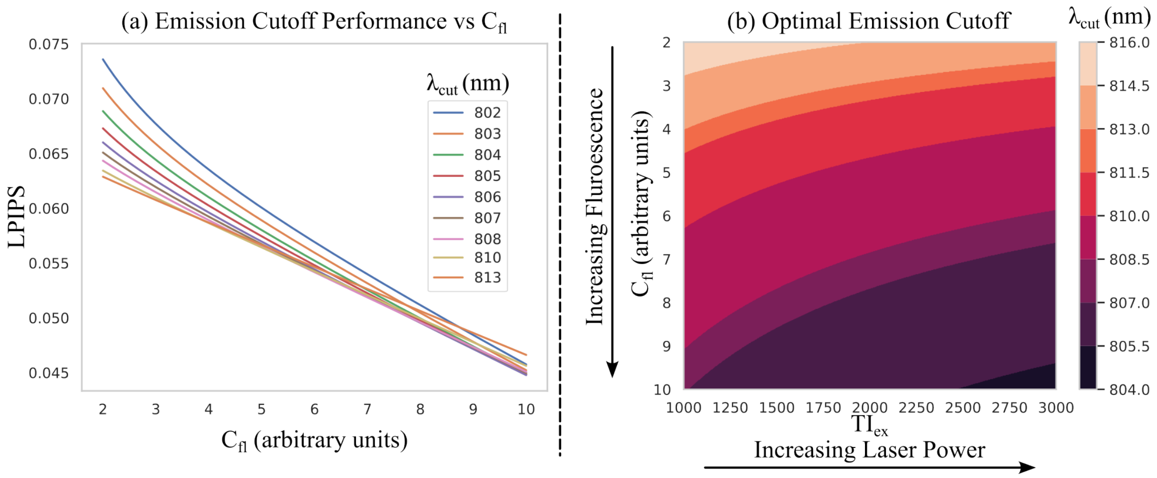
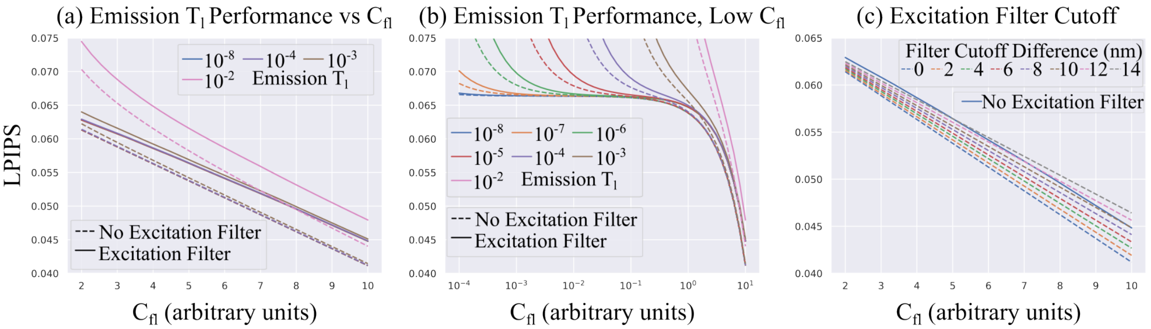


Figure 5.5: **Emission Filter Cutoff Results:** (a) shows the performance of different emission filter cutoff wavelengths,λ^{em}_{cut}, at different fluorescent brightness values, C_{fl}, for a fixed $TIex = 2000$. (b) shows the best performing cutoff value over C_{fl} and TI_{ex}, We find that the best performing λ^{em}_{cut} changes with both laser power and fluorescence concentration.

the other parameters at $\lambda_c = 780$, $\sigma_\lambda = 10$, $T^{ex}_l = T^{ex}_h = 1.0$, $T^{em}_l = 10^{-6}$, $T^{em}_h = 1$ unless otherwise noted.

Emission filter cutoff: In this experiment, we find the optimal emission filter cutoff as a function of C_{fl} and TI_{ex}. We find the optimal λ^{em}_{cut} for each C_{fl} and TI_{ex}. Fig. 5.5(a) shows how performance changes for a many λ^{em}_{cut} for a fixed TI_{ex} and changing C_{fl}. Fig. 5.5(b) shows the landscape of optimal λ^{em}_{cut} values, in general we find that the optimal emission cutoff value will depend both on C_{fl} and TI_{ex}. This is a surprising result because generally in hardware design of these systems, these factors are considered largely independently from each other. This underscores the non-linear relationship between S_m and L_m that comes from the denoising process.

Emission filter stop band transmission: We investigate the effect of T^{em}_l over a range of C_{fl} values with fixed $TIex = 2000$, at each C_{fl} we report the performance of the best performing λ^{em}_{cut}. Fig. 5.6(a-b) shows the results of this experiment over a number of T^{em}_l. We find that $T^{em}_l = 10^{-4}$ is a sufficient filter in this case over most of the C_{fl} range; filters with lower transmission (higher optical densities) do



(a) Emission T_l Performance vs C_{fl}

(b) Emission T_l Performance, Low C_{fl}

(c) Excitation Filter Cutoff

Figure 5.6: **Emission and Excitation Filter Results:** (a) shows performance for changing the emission filter stop band transmission, T_l^{em}. Dashed lines indicate the use of an excitation filter. (b) shows performance of the experiment in (a) on a log scale C_{cl} axis for visualization of small C_{fl}. (c) shows the effects of non-ideal cutoff distances between the emission and excitation filter $\lambda_{cut}^{em} - \lambda_{cut}^{ex}$. As the cutoff gap increases, the value of the excitation filter is diminished.

not significantly improve performance. As C_{fl} gets small, Fig. 5.6(b) shows that there is minimum C_{fl} that is cable of being measured before performance is greatly reduced. This minimum is a function of T_l^{em}. So low T_l^{em} is important for very high sensitivity systems but may not be important over most common operating ranges. This is important because pushing to low T_l^{em} can greatly increase the cost of filters, often requiring custom designs, and it appears in most C_{fl} ranges these expensive filters are not necessary. Importantly, this is a result of the video denoiser being used; by using a video denoiser that is trained to remove LLL we can allow some LLL in our hardware because our software can compensate for it.

Excitation filter: Excitation filters may be used to reduce the amount of excitation light in the pass band of the emission filter (LLL); however, it also reduces the amount of excitation light entering the scene so reduces fluorescence photons. To test the importance of an excitation filter, we add an excitation filter to the last experiment with $T_l^{ex} = 10^{-6}.T_h^{ex} = 1.0$, and $\lambda_{cut}^{ex} = \lambda_{cut}^{em}$, The dashed lines in Fig. 5.6(a-b) shows result for this experiment. We find that an excitation filter in general improves the performance over all C_{fl} by a moderate amount. However, this assumes an optimal excitation and emission filter match of $\lambda_{cut}^{ex} = \lambda_{cut}^{em}$, what happens when we increase the distance between the cutoffs which may be necessary due to limited filter availability? Fig. 5.6(c) shows the result of increasing $\lambda_{cut}^{em} - \lambda_{cut}^{ex}$, we choose $T_l^{em} = 10^{-4}$ and use the best performing λ_{cut}^{em} for each fixed value of $\lambda_{cut}^{em} - \lambda_{cut}^{ex}$. The gain of using an excitation filter drops sharply as the distance between the cutoff increase.The drop-off is more significant at larger C_{fl} indicating that increasing photon output is more important than limiting LLL at high C_{fl}. The excitation filter actually hurts performance at high C_{fl} if the distance between the cutoff is only 10 nm. Although, in general the importance of the excitation filter will be a function of how much overlap there is between the excitation light and emission filter, so becomes much more important for larger σ_λ or as λ_c approaches the emission filter.

Excitation Light

The primary excitation light source is parameterized by the spectral width, σ_λ, and the center wavelength, λ_c. In this set of experiments, we test how the choice of light source will effect the performance when using optimal filter choices. First we find the performance landscape of σ_λ and λ_c when using an excitation filter, and optimizing the cutoff value for both filters; we fix $C_{fl} = 5, T_h^{ex} = T_l^{em} = 10^{-6}$ and $T_l^{ex} = T_h^{em} = 1$. Fig. 5.7(a) shows the found landscape, generally, a smaller spectral width and a center wavelength near the excitation peak of ICG (about 790 nm) performs the best. Interestingly, we find that a narrow-band source with λ_c an offset from the excitation peak will perform similarly to a wider band source aligned with the excitation peak; this may be important when a discrete number of light sources are available. We also find that the filter cutoffs, λ_{cut}, depend heavily on the choice of laser, Fig. 5.7(b) shows the optimal filter cutoff value over σ_λ and λ_c. Next, we test the importance of the excitation filter, whose purpose is to clean up the excitation light source by removing long tails in the emission band. Fig. 5.7(c) shows the results from removing the excitation filter. We find that the excitation filter is not necessary with narrow-band light sources; however, as the band width of the light source increases the excitation filter becomes crucial to maintain performance.

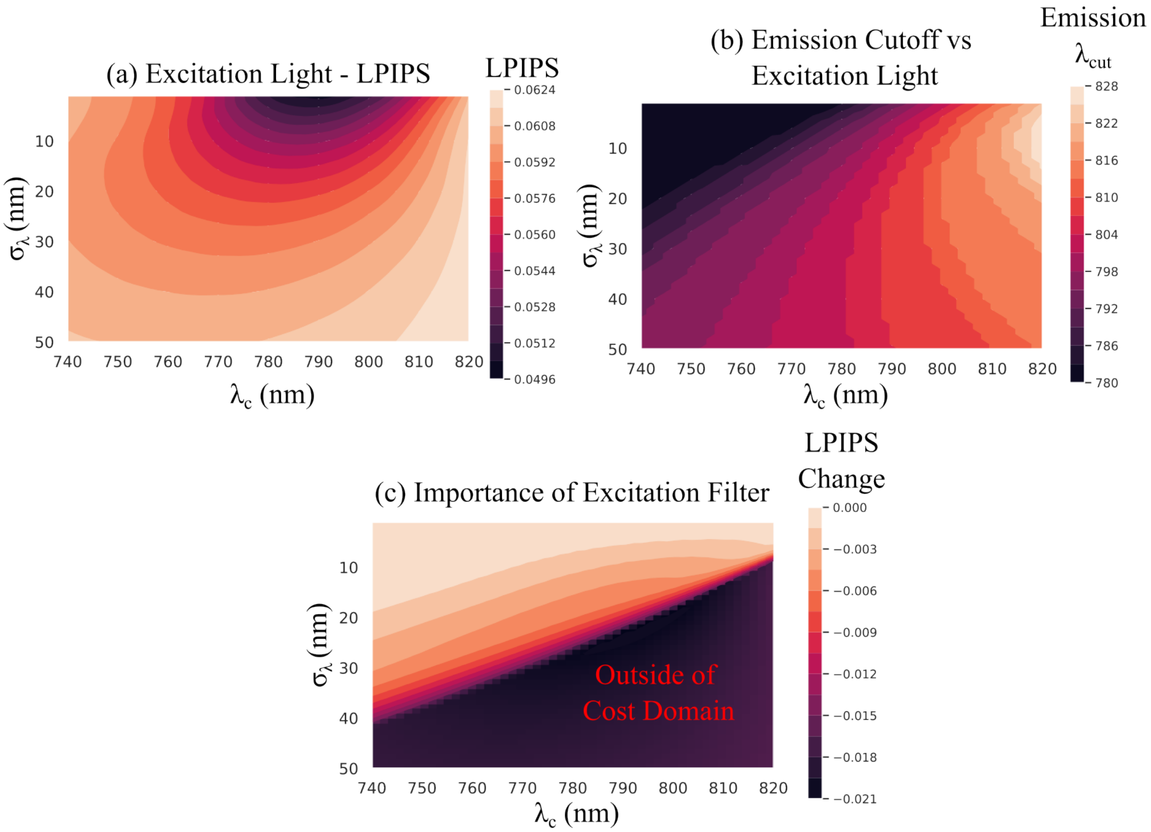


Figure 5.7: **Excitation Light Optimal Parameters:** This figure shows the results of changing the light source parameters. (a) shows the performance at different excitation light spreads, σ_λ, and center wavelengths, λ_c. (b) shows how the optimal cutoff filter value chagnes with the excitation light parameters. (c) shows the change in LPIPS if the excitation filter is removed. At small σ_λ the excitation filter has little effect, but at larger σ_λ the excitation filter is vital. Note the lower right of this plot left the domain of our cost function due to excessive very high L_m. All generated at $C_{fl} = 5$, other parameters are optimized.

Finally, we ask the question how the spectral bandwidth and excitation power are related. The motivation of this question is to evaluate whether LED or laser sources should be used as the excitation light source. In general, lasers' will have much narrower bandwidths; however, it may be easier to run multiple LEDs in parallel producing more excitation power for the LEDs, at similar price points. To test this we use the same parameters as our previous experiment (including

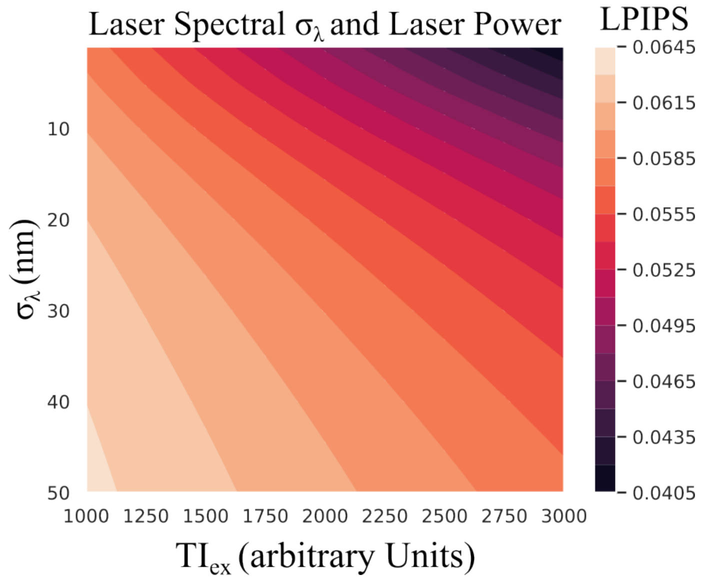


Figure 5.8: **Excitation Light Spread vs. Power:** This figure shows how LPIPS changes with laser spread σ_λ and laser power $\mathtt{TI}_{\mathtt{ex}}$. Tested at $\mathtt{C}_{\mathtt{fl}} = 5$, other parameters are optimized.

excitation filters), but we choose the optimal λ_c for each point and vary $\mathtt{TI}_{\mathtt{ex}}$. Fig. 5.8 shows the result of this experiment. We find that generally, an LED setup with a reasonable $\sigma_\lambda = 40$ will need to about twice as bright as a laser setup with $\sigma_\lambda = 10$.

5.5 Conclusion

In this chapter, we explored how video denoisers can be used to develop a cost function for FGS hardware configurations. In particular, this cost function ties the cost of signal photons and LLL photons into one number that relates to the end human perceptual quality of the video. This gives a well defined cost function that can be used to choose hardware. Importantly, the cost of LLL photons and signal photons are non-linear when a video denoiser is used. Importantly, the video denoiser is able to remove many LLL photons potentially allowing for cheaper hardware to be used.

Limitations: The largest limitation is this method relies on good data in order to generalize. This work is based on video denoisers trained with mock surgical data rather than clinical data. Gathering real data to train denoisers will make this method more accurate. Another problem is the choice of cost function to evaluate perceptual quality. In this work, we used LPIPS which is a deep learning based metric that is trained using human feed back on natural images. This metric could be improved by either fine-tuning, or training a new method based on surgical perceptions of images.

6 WATCHING THE WATCHERS: CAMERA IDENTIFICATION AND CHARACTERIZATION USING RETRO-REFLECTIONS

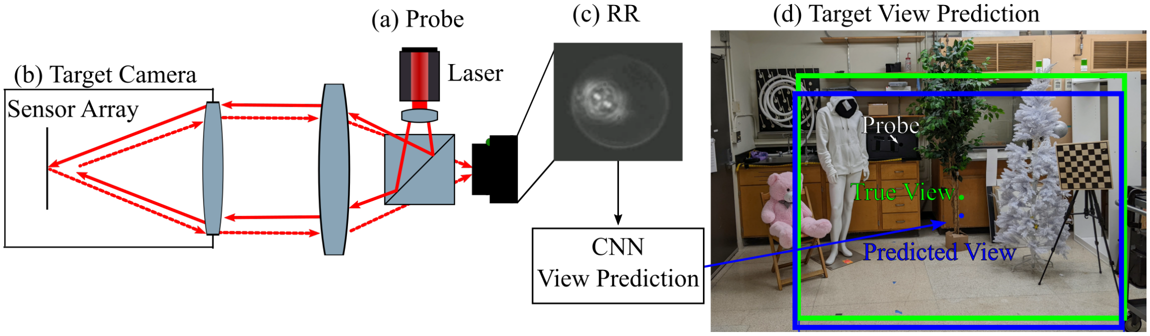


Figure 6.1: **Retro-Reflection Imaging:** In retro-reflection imaging, (a) first the probe emits light into a scene, then (b) a target camera images this light onto a sensor array. Then some light reflects off of the sensor array and returns through the target aperture and lens. This light returns directly to our probe in a retro-reflection (RR), we use a beam splitter to image the RR onto a camera in our probe. (d) The RR contains information about the target camera such as its rotation allowing us to predict the view of a camera using a neural network.

Retro-reflections (RRs) or cat-eye reflections occur when a focused imaging system reflects light directly back to the light source. Because light between the RR and the light source can be collimated and therefore lose little intensity during propagation for distances below the Rayleigh range, RRs can be used for sensing across very large distances even with only moderately sized probe beam widths. Small RRs placed on the moon, for example, can be probed using an eye-safe laser power from a telescope on earth, over 400,000 kilometers away. In recent years, there has been a lot of work using retro-reflections in security and privacy settings to detect hidden spy cameras (He et al. (2018); Li et al. (2015); Qian et al. (2015); Liu et al. (2019c,b); Huang et al. (2021); Svedbrand et al. (2019); Sami et al. (2021)) or optical sights (Mieremet et al. (2008); Lecocq et al. (2003); Auclair et al. (2013)),

This chapter is based on work done for the following publication (Seets et al. (2024a)): Seets, Trevor, Alec Epstein, and Andreas Velten. "Watching the watchers: camera identification and characterization using retro-reflections." Optics Express 32, no. 8 (2024): 13836-13850.

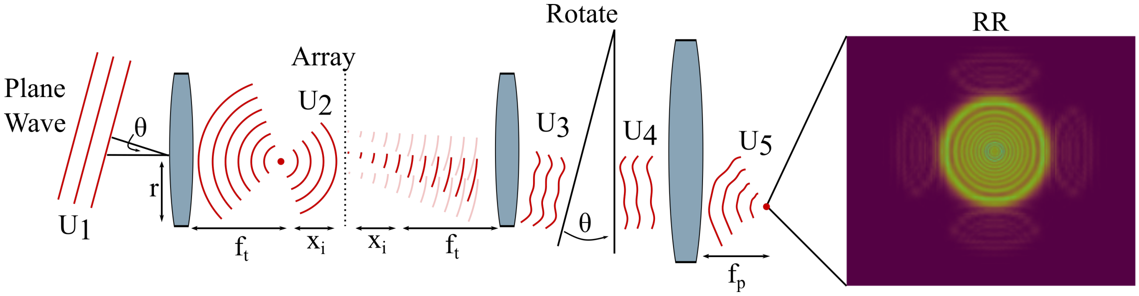


Figure 6.2: **Unrolled Light Path:** This figure shows the unrolled light path in our problem. First, the target lens is illuminated with a rotated plane wave. The light then travels to the sensor array at $f_t + x_i$. The sensor array modulates the intensity of light following its pattern. Then we propagate the result another $f_t + x_i$. The light then leaves the target lens, which we rotate back to be in line with the optical axis of the probe system. The result is then propagated to our probe which images the final retro-reflection (RR).

where a light source illuminates a scene and the resulting RRs from hidden cameras or other small optical systems (Leal-Junior et al. (2020, 2022)) are subsequently detected and localized. In this work, we wish to expand the scope of uses for RR to more than just detection by asking the question "what other information is contained in RRs?" We find that for many target cameras the shape of a RR can give us information on many imaging parameters such as where a camera is looking, the focal length used, and even the depth at which the camera is focused, while also being able to classify cellphone models. Our work has applications in control, privacy, identification, and image validation.

Our RR probe, shown in Fig 6.1, consists of a light source, either a laser or LED (Wu et al. (2021); Xu (2021)), and camera with a shared optical path. First, we send light into the scene, where it will be imaged by a target camera system. Some of the imaged light will reflect off the target camera sensing array and then it will be re-focused by the target camera's lens back along the input light path in a RR. In general, RRs are quite bright compared to standard diffuse reflections because the return light is well-collimated due to the focusing nature of the target camera's lens so very little light is lost on the return path. The intense brightness of RRs have made them useful in detecting small hidden cameras or cameras at long distances. However, we find that RRs also contain rich information about the imaging parameters of a target camera. For example, if the target camera rotates or changes focus, changes appear in the RR allowing us to accurately predict where a target camera is looking and at what depth it is focusing. We also see signals related to the structure and the Bidirectional Reflectance Distribution Function (BRDF) of the sensor and filters in the optical path.

In this paper, we conduct a study of RRs by combining existing diffraction-based models from Gong et al. (Gong et al. (2016); Gong and He (2017)) and Liu et al. (Liu et al. (2019a)) to include rotation effects of the target imaging system. We find the resulting simulation pipeline generates results matching our measurements for thin lens systems. However, for more complex imaging systems, we find the RRs differ significantly from a thin lens model, so we study more complex RRs with a data-driven approach. We capture two datasets with commercial lens systems, one containing sets of RRs and rotation angles and the other sets of RRs and focal distances. We use these datasets to train two models taking as input a RR and predicting the corresponding parameter from each dataset. The first type of model is a linear model based on the popular lens aberration basis, Zernike polynomials, this model shows that changes in the target camera can be understood as lens aberrations in the RR. Our second model is a standard convolutional neural network (CNN) that pushes the performance and performs well on both datasets. The resulting models are able to accurately perform gaze tracking to predict where a target camera is looking such as in Fig 3.1.

We finally study RRs from different cell phones for gaze tracking and classification. We find that the RRs are unique between different models of cell phones allowing for very accurate classification. We also find that cellphone gaze tracking is possible using RRs albeit for some phone models over a small angular range. Additionally, we find that for many cellphones the RR is only present when the cellphone is actively imaging (i.e. camera app is open), allowing us to find when a cellphone is actively recording and what it is looking at. This could be used to validate images by providing outside proof of where and when a photo was taken.

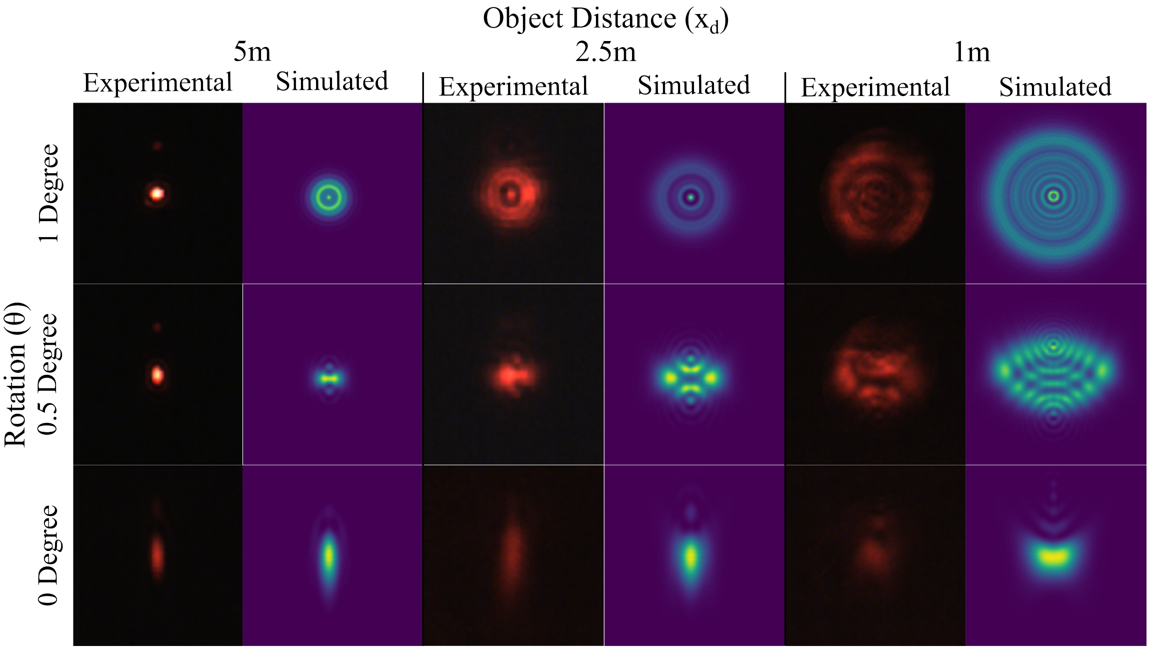


Figure 6.3: **Thin Lens Experiment:** We measure and simulate RRs of a thin 100mm lens focused at different object distances and at different rotations downward. Our simulations and experiments match fairly well.

6.1 Toy Simulation Model

Here we introduce a toy simulation model for the thin lens case in order to gain intuition on the RR problem. Our simulation model is adapted from Gong et al. (Gong et al. (2016); Gong and He (2017)) and Liu et al. (Liu et al. (2019a)) with an additional simulation step to account for the rotation of the lens at the exit of the target aperture. This step aligns the returning light from the target camera with the optical axis, allowing for simulation of larger target rotations without the beam leaving the simulation bounds and without greatly increasing the computation time that would be required to expand the bounds.

The unrolled optical path of our simulated geometry is shown in Fig 6.2. First, we generate the illumination field $U_1(x, y, z = 0)$ at the target aperture which we assume can be well approximated by a plane wave,

$$U_1 = e^{iky \sin\theta} \tag{6.1}$$

where $k = \frac{2\pi}{\lambda}$ is the wave number, θ is the lens rotation off of the optical axis, and for notational simplicity, we have dropped the spatial dependencies of U_1. Because our system is rotationally symmetric we only need to consider the magnitude of the rotation and define the coordinate system such that y is the direction of this rotation. Next, we model U_1 going through the target lens as a circular masking operation followed by multiplication with a quadratic phase exponential,

$$U_1' = U_1 \mathrm{Circ}_r(x, y) e^{-i\frac{k}{2f_t}(x^2+y^2)} \tag{6.2}$$

where r is the aperture radius of the target, $\mathrm{Circ}_r(x, y)$ is a binary circle mask of radius r centered at $(x = 0, y = 0)$ and f_t is the target focal length. We then propagate U_1' to the target's sensor array,

$$U_2 = R[f_t + x_i]\big(U_1'\big) \tag{6.3}$$

where x_i is the distance from the focal plane to the sensor array and $R[d](U)$ is the free space propagation operator of U a distance d. We treat the sensor array as a reflective surface with a mask, $\mathrm{Arr}(x, y)$, for example, $\mathrm{Arr}(x, y)$ could be an arrayed pixel system or microlens array. Therefore U_2 is then modulated by the mask $\mathrm{Arr}(x, y)$. Then in the unrolled version, the result propagates another distance of $f_t + x_i$ before exiting the target lens. These operators give,

$$U_3 = R[f_t + x_i]\big(Arr(x, y)U_2\big) \tag{6.4}$$

$$U_3' = U_3 \mathrm{Circ}_r(x, y) e^{-i\frac{k}{2f_t}(x^2+y^2)} \tag{6.5}$$

where U_3' is the field exiting the target imaging system. If we propagate U_3' it will be moving at an angle of θ with respect to the z-direction and is not perpendicular to our imaging system. In order to deal with this we need to rotate U_3' by θ in order to align it with the new propagation axis. We use the method in (Stock et al. (2017))

to get a rotated field U_4, the method is based on direct integration in the rotation dimension and a frequency domain approach along the non-rotation axis, resulting in significant speed improvements compared to a full integration method.

Finally, U_4 is propagated a distance d and imaged by our infinity focused probe. The resulting field on our imaging sensor, U_5, is given by,

$$U_5 = R[f_p]\Big(e^{-i\frac{k}{2f_p}(x^2+y^2)}R[d]\big(U_4\big)\Big) \tag{6.6}$$

where f_p is the focal length of our probe. The final diffraction pattern is given by $I(x,y) = |U_5(x,y)|^2$. We use the python package Light Pipes implementation (FredvanGoor et al. (2019)) for the $R[d]$ operator and our own implementation of the method in (Stock et al. (2017)) for rotating the field.

Liu. et al (Liu et al. (2019a)) found that RRs can occur from both the sensor array and a filter above the array. This is easy to account for in our model by simply adding the result of multiple forward passes, one that accounts for each optical element.

Distance Invariance: With our simulation model we find that the results are invariant to d, as long as diffraction from the target to d does not create a spot larger than the probe aperture. This allows us to speed up simulations by using smaller values of d. However, experimentally we find that only specular RRs are distance invariant and if there are non-specular components in a reflection they may change size with distance.

Measuring x_i: We avoid the difficulties in directly measuring x_i by instead measuring the object distance, x_d, of the target lens. x_d, and x_i are related by the lens maker's formula,

$$\frac{1}{f} = \frac{1}{x_d} + \frac{1}{f + x_i}. \tag{6.7}$$

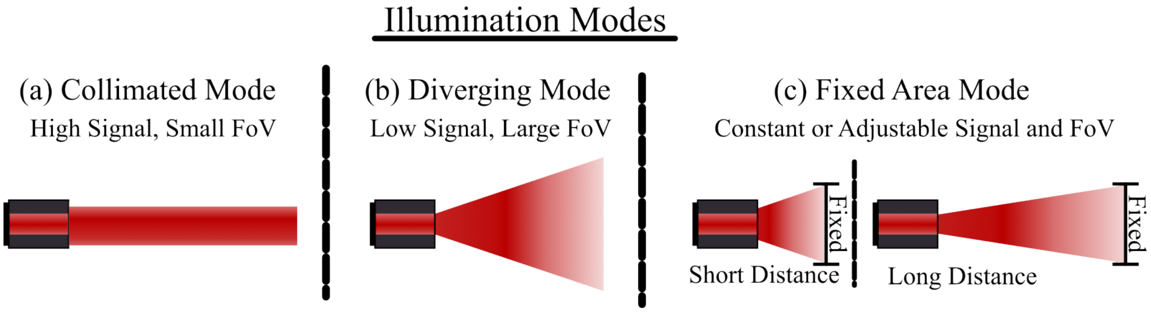


Figure 6.4: **Illumination Modes:** This figure shows the three illumination modes. (a) The collimated mode uses a collimated illumination that maintains a fixed area and signal level with distance. (b) The diverging mode covers a larger area at the cost of signal density as distance increases. (c) The fixed area mode adapts to a predicted target distance by changing the illuminations divergence angle. This mode allows for larger fixed areas than the collimated mode.

6.2 Illumination Modes

We identify 3 illumination modes for probe setups based on the divergence of our illumination source that have different tradeoffs. The illumination divergence angle controls how quickly the illumination beam expands, and mainly affects the illuminated field-of-view (FoV) and the signal strength at the target. The three modes are shown in Fig. 6.4

(a) Collimated Mode: The first mode is the collimated mode, where the illumination light source is collimated. This mode, within the Rayleigh range (12km with a beam radius of 5cm), has distance invariance intensity response due to the collimation of both the illumination light and return light; however, in this mode the illumination light only covers a FoV equal to the aperture of the probe. This is the mode analyzed in our simulations.

(b) Diverging Mode: The second mode seeks to expand the FoV without increasing the complexity or size of the probe. To do this we simply defocus the illumination source, so the illumination beam expands with distance. This mode leads to a larger FoV due to a larger illumination area but it spreads out the illumination power as a function of distance, leading to a distance squared drop off in the RR intensity.

We use this mode in our experiments to have a larger FoV and experimentally find that the divergence angle has very little impact on the shape of the returning RRs (including the Collimated Mode).

(c) Fixed Area Mode: In the final mode, the illumination source divergence is adjusted based on expected or measured target distance to always cover the same area at the target distance. This leads to a system that can observe an area in the scene with signal strength that is independent of distance but allows to observe much larger areas with a limited detector. This mode adds complexity to the probe setup but allows for finer control of the FoV and returning intensity.

Finally, it is possible to combine multiple probes with diverging beams and overlapping FoVs to create signals that overall remain constant with distance in SNR and illumination flux throughout a desired volume. A target at short distance would only be visible to one probe, but generate a stronger signal while a target at larger distances would create weaker signals in an individual probe, but visible to more probes. Using these tools it is possible to create probes with constant SNR and illumination power over any desired scene volume; for example, the audience of a movie theater using a set of probes above the view screen.

6.3 Experiments

We use a 640nm laser, $f_p = 200$mm, and vary the parameters of our target camera. First, we verify our model with a camera with a simple lens (Thorlabs LA1509-B) that fits our thin lens approximation, then move on to commercial photography lenses with more complicated aberration profiles. Due to the complex nature of modern lens designs, our thin lens assumption no longer holds and we use a data-driven approach in order to show the ability of our system to predict different imaging parameters of the target system. Finally, we switch to a 785 nm laser to image 11 different cellphone cameras as the target and use the reflections to classify the phone. Unless otherwise noted the target camera is located 5.5m away from the probe for all measurements and the probe is used in the diverging illumination mode.

Thin Lens Experiments

To verify our diffraction model we start with a target consisting of a machine vision sensor (STC-MBA5MUSB3) and a thin $f_t = 100$mm Thorlabs lens located 5m away from the probe. We simulate and experimentally image the RR at $\theta = [0, 0.5, 1]$ degrees, and $x_d = [1, 2.5, 5]$m. Both results are shown in Fig. 6.3, the shape of the experimental RRs closely matches the simulations albeit with fewer details.

We notice a few patterns in Fig. 6.3 as the target's rotation changes, when the target is looking directly at the probe an airy disk appears. As the target rotates away from the probe at small rotations (i.e. 0.5 degrees) 2 airy disk-like patterns appear that are symmetric across the line perpendicular to the rotation and as the rotation increase the patterns get further apart. Finally, at large rotations, a long ellipse often forms in the direction of rotation. Next we notice patterns as the target changes it's object distance (focus), we notice that the size of the RR is larger when x_d is small, this is due to the target system not focusing on the laser light so the illumination is large on the sensor leading large reflections and RRs shaped like the target aperture. As the target focuses closer to infinity (x_d increases) the RR becomes smaller.

We repeat this experiment for $f_t = 75, 50$mm. At these focal lengths, our simulation begins to break down, likely due to our thin lens assumption; however, the general shapes are the same just not at the exact same parameters predicted by the simulation model. For completeness, the $f_t = 75, 50$mm results are included in the supplement.

Effect of Distance

We conduct an experiment in the diverging illumination mode to see the effect of target distance from our probe. From our simulations, we expect that the shape of the RR will remain mostly constant allowing for our models to be robust to changes in target distance. Also, because the returning light from the target is relatively collimated we expect any decrease in RR intensity to be explained by the increase in the illumination beam area due to distance ($\propto \frac{1}{d^2}$). In order to test this, we capture

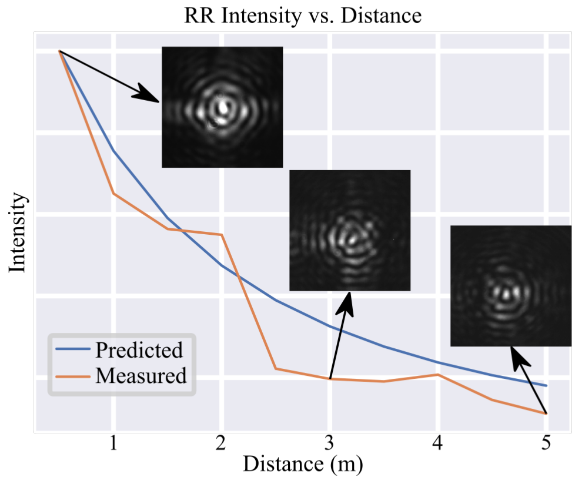


Figure 6.5: **Experimental Distance Behavior:** We measure the effect of distance for a head-on RR in the diverging illumination mode. The RR shape does not change much while the intensity follows the expected decay from an expanding beam. Images have different tone mappings to better show the RR shape.

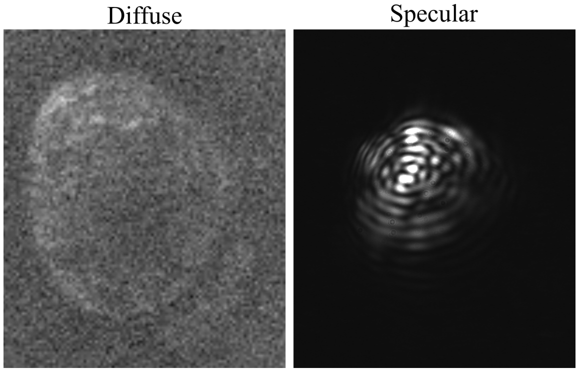


Figure 6.6: **Specular vs. Diffuse RRs:** Specular and diffuse sensor plane surfaces produce different RRs.

a head-on ($\theta = 0$) RR for a camera mounted with a commercial lens (Fujinon DV3.4x3.8SA-1) at distances from 0.5m to 5.0m in 0.5m increments, our results are summarized in Fig 6.5. To find the intensity of a RR, we sum all pixels above the noise floor for each image. We measure our beam expansion to be half a degree and calculate the beam area at each distance. We find the intensity drop-off and RR shape match our expectations.

Specular vs. Diffuse Reflections

We investigate the difference between a specular and diffuse reflection at the target sensor plane. We use a commercial lens (Fujinon DV3.4x3.8SA-1) as the target lens and place a mirror (specular) or white cardboard (diffuse) at the sensor plane. We find that the type of reflection plays a role in the shape and behavior of the RRs. Example images of the diffuse and specular reflection are shown in Fig 6.6. Diffuse reflections are larger and take the shape of our probe aperture and scale in size with d while specular RRs are brighter and constant in size over d. This behavior could potentially be used to help probe the reflectance profiles of sensors at different wavelengths possibly helping with target classification.

Commercial Lens Datasets

Commercial lens designs are often systems of lenses and break the thin lens assumption used in our simulation model, so to deal with these lenses we opt for a data-driven approach. Our goal is to collect a dataset of RRs from a commercial lens along with the corresponding imaging parameters (f_t, x_i, θ, ϕ). For ease of capture and to reduce the total data requirements, we split our imaging parameters into two different datasets. In the first dataset "Focusing," we use a variable focal length Canon lens to investigate the behavior of f_t and x_d. In the second dataset "Rotation," we use a large FoV C-mount lens (Fujinon DV3.4x3.8SA-1) to investigate the lens rotation parameters θ and ϕ.

Rotation Dataset The rotation dataset is made up of 7 100 second videos at 15 FPS using a machine vision camera (STC-MBA5MUSB3) for both the probe and target camera. The two cameras operate simultaneously at the same frame rate and we sync the videos after capture. The target camera is focused on the probe, and the probe laser is visible to the target camera so the location of the focused probe light will be a small spot in the target video. The (x, y) location of the bright laser spot will correspond to a unique (θ, ϕ) pair; for ease, we use (x, y) as a stand-in for (θ, ϕ) in our models as the prediction target.

Fig. 6.7(a) shows an example of the RR as the camera looks upward. Similar to the thin lens, when the target camera looks directly at the probe an airy disk pattern is visible. As the camera is rotated the airy pattern becomes elliptical and a brighter line appears perpendicular to the rotation direction. Finally, at large rotations, a strong central ellipse is present along with a dim "+" pattern.

Focusing Dataset In order to evaluate the effects of f_t and x_d we use a Canon Rebel T5 camera with a Canon EF 18-135mm lens. Instead of explicitly finding x_i we capture a dataset with the target camera focused at different distances x_d away. We vary $f_t = [50, 85, 135]$mm, for each f_t we focus the camera using autofocus at $x_d = [3, 3.5, 4, 4.5, 5, 5.5]$m. Then for each combination of f_t and x_d we capture approximately a 1-minute probe video with the hand-held target camera being rotated and moved. Fig. 6.7(b) shows an example image from the Focusing dataset with $f_t = 135$mm at increasing x_d, at this f_t we see the RR becomes smaller and closely approaches a point as the object distance gets further away.

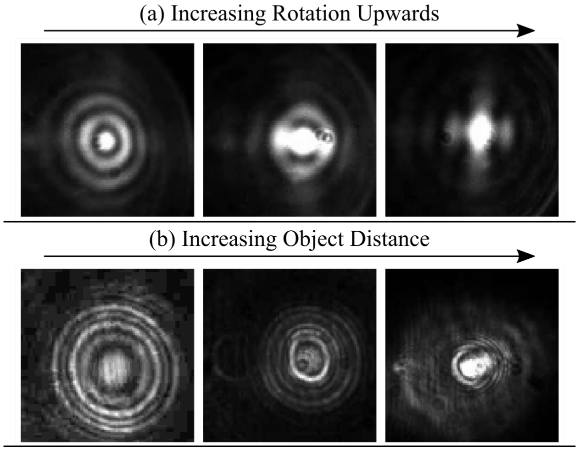

(a) Increasing Rotation Upwards
(b) Increasing Object Distance

Figure 6.7: **Select Dataset Images:** This figure shows a few examples from our two datasets. (a) shows an example of the RR as the target rotates upwards. (b) shows the RR as the target focuses further away (x_d increases) for fixed f_t. (c) shows the effects of decreasing focal length (f_t) on the RR for a fixed x_d.

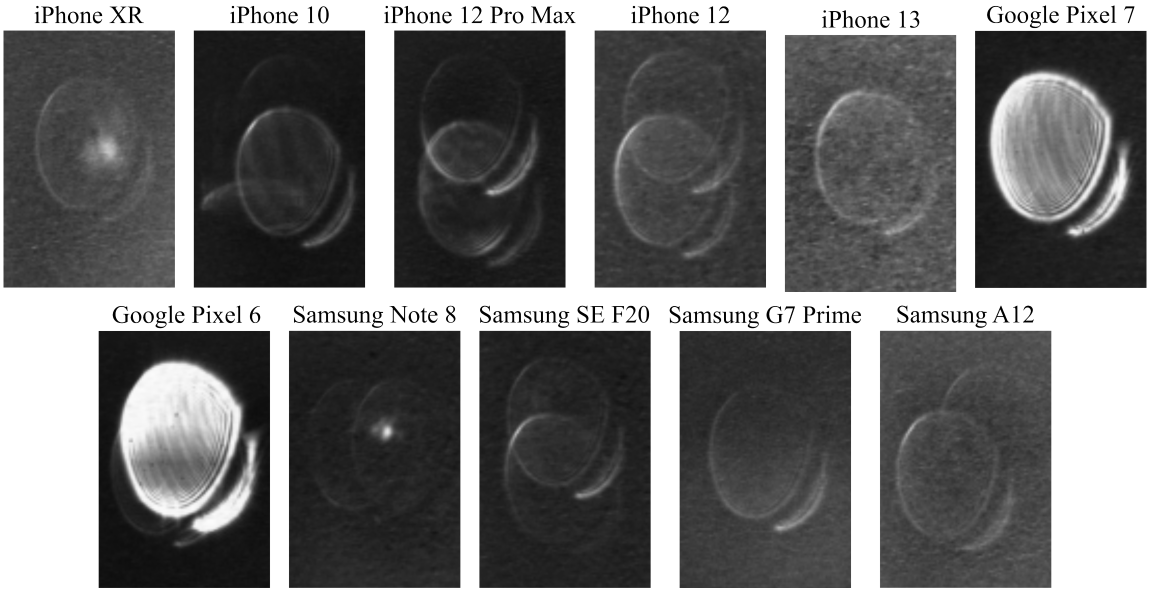


Figure 6.8: **Phone RRs:** This figure shows representative examples of the RRs from 11 phone models. Since most modern phones combine multiple cameras, the RR contains information from each camera's lens in the phone's camera cluster. Some phone models contain strong specular RR spots such as the iPhone XR and the Samsung Note 8.

Cell Phones

Cellphones are particularly challenging due to their small aperture, and complex optics designed to remove many aberrations that allow us to distinguish larger lenses. For many modern cell phones, the RR is just simply a spot at 640 nm with very little variation as the phone rotates. To allow rotation tracking of cell phones we use a 785 nm laser instead which many cellphone camera paths filter out with IR cut filters near the sensor. The IR cut filters may not be placed at the focal plane of the cameras so this will introduce a slight defocus (i.e. non zero x_i) resulting in more unique reflections from cell phone cameras. Because our probe camera now captures IR light we also capture the IR lasers used in the phone LiDARs of newer models giving another useful feature. We notice a strong difference between many phone models in the resulting reflections, an example from each phone model is shown in Fig. 6.8.

We capture a dataset of 11 phones as the target camera using the same procedure as the Rotation Dataset. We then predict the phone model and the phone rotation from the RRs. Many cell phones have a sharp RR signal dropoff with rotation, so the majority of our phone rotation data was confined to the center 50% of the camera FoV

In phones, we notice strong RRs from early optical elements, such as the IR filter, that exhibit distance-dependent sizes from diffuse reflections and a dimmer distance-invariant specular spot that is usually only present when the phone camera is actively capturing. The specular spot is most obvious in the iPhone XR and Samsung Note 8 in Fig. 6.8 and is possible to isolate in other phone models by toggling the camera app causing the specular spot to appear or disappear while the diffuse components are unchanged. Potentially, the presence of this specular spot changes due to focus changes when the camera app is activated or deactivated; possibly a "resting" setting for the focus could explain this behavior, so may be software dependent. These elements may be wavelength dependent so using a system with multiple optimized wavelengths could provide additional information for many camera models. For example, we notice that the specular RR that is visible when the camera app is open is more visible at 640nm than at 785nm for iPhone models while Google Pixel models show a stronger diffuse RR at 785nm.

6.4 Parameter Prediction

We train two models, a linear model based on Zernike Polynomials and a CNN.
Zernike Model: We first test our intuition that lens abberations are useful RR features for parameter prediction. We do this by using the popular Zernike polynomials (Wyant and Creath (1992); Antonello and Verhaegen (2015); Lakshminarayanan and Fleck (2011); Tango (1997)), which are an orthogonal basis on the unit circle often used in optics to distinguish and classify aberrations. If our intuition is correct a simple model based on Zernike polynomial features should provide reasonable performance.

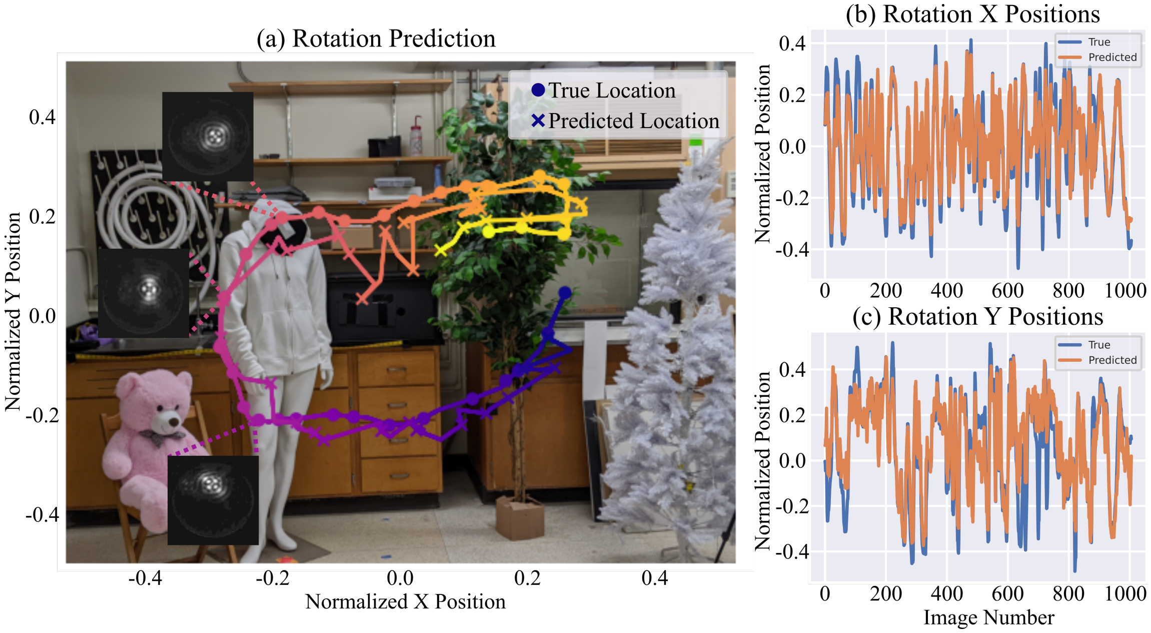


Figure 6.9: **Rotation Prediction Results:** (a) displays the path of the target camera's center of FoV and corresponding Res-Net-18's prediction over time encoded by the color of the line. The background image represents the FoV of the target camera. Three select RRs are also shown on the left side of the image as the target camera looks downward the RRs have large changes. (b-c) show the X and Y Res-Net-18 predictions and ground truth over the entire training set. The predictions are in general close with a few noticeable outliers.

We extract the Zernike features at different scales centered on our RR. We center the RR by finding the brightest point in a smoothed RR image, then crop a region of interest around this point. We then create square crops $(s \times s)$ at 4 different scales, $s = 100, 200, 300, 400$. Let $a^s_{m,n}$ and $b^s_{m,n}$ be the positive and negative Zernike polynomials, respectively, at scale s. We calculate the coefficients on the square root intensity of the crops up to $n = 12$ for all $m \leqslant n$. We use the python implementation from Antonello and Verhaegen (2015) to calculate the Zernike polynomial coefficients. We finally take the square and 3^{rd} power of the coefficients to create our final feature vector $\psi = \{(a^s_{m,n})^p, (b^s_{m,n})^p : s \in [100, 200, 300, 400], 0 \leqslant m \leqslant n \leqslant 12, p \in 1, 2, 3\}$. We then use Scikit-learn's implementation of Ridge regression (Pedregosa et al. (2011); Hoerl and Kennard (2000)) $(\alpha = 10^{-6})$ with features, ψ, to fit our datasets

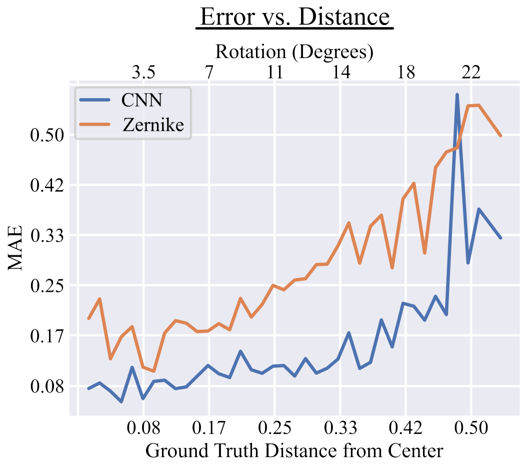


Figure 6.10: **Errors in Rotation dataset:** This figure shows our error rate (MAE) as our target camera looks further away from the probe in normalized units of the fraction of the field of view (to match Fig 6.9).

to get our final Zernike models.

Neural Network: To further improve the performance of our predictions we use Res-Net-18 (He et al. (2016)). We hope that Res-Net-18 is able to pick up on more features than our simple linear model; for example, due to imperfections in our beam splitter, our input light has a dim secondary component at a slightly different angle. This secondary component leads to a dimmer RR at a slightly different illumination angle that the Res-Net-18 model may be able to use.

Table 6.1: This table shows the errors of our models on different datasets, we report errors as MAE/RMSE for each task with both the Zernike and Res-Net-18 models. Rotation errors are normalized to the target camera's field of view.

Method	Rotation	f_t (mm)	x_d (m)
Zernike Model	0.23/0.22	8.89/12.1	0.344/0.448
Res-Net-18	0.12/0.12	0.30/1.45	0.057/0.112

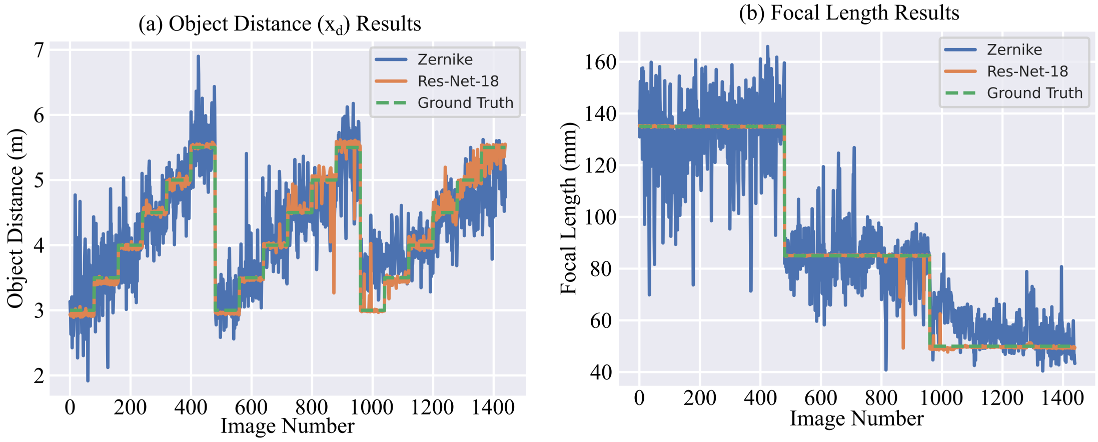


Figure 6.11: **Focusing Dataset Prediction Results:** The results of the prediction models are shown for (a) 6 varying object distances (x_d) from 3 to 5.5 meters, where the camera is focused to a distance x_d, and for each x_d (b) 3 different focal lengths (f_t) are tested. The Res-Net-18 model successfully predicts the different combinations of x_d and f_t.

Model Results

We train both models on the Rotation and Focusing dataset using the same train/test split, for the cellphone dataset we only use the Res-Net-18 model due to multiple cameras in the cellphones. For each video in a dataset we use the first 80% of the video for training and the last 20% for testing.

Commercial Lens Datasets:

For each model, we compute the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) between the model predictions and ground truth.
Rotation: For the Rotation dataset our models predict the pixel location where the probe laser will appear in the target frame, giving us an error in pixels, we normalize this value by our target cameras field of view in pixels. So a RMSE of 0.12 indicates that our prediction was on average within a ball with a radius equal to 12% of the field of view of the camera.

Fig. 6.9 shows the results of the predictions of the Res-Net-18 model on the rotation dataset. Fig. 6.9(a) displays our predictions as well as the ground truth rotation in a section of a test video on top of the FoV of the target camera centered at the probe location along with 3 RR showing the change as the target camera moves. Fig. 6.9(b-c) displays the X and Y normalized coordinate prediction and ground truth over the entire testing set, in general, we notice that the predictions are quite close however there are occasional outliers most likely due to lower signal in certain frames.

To examine how the target gaze affects our error rates, we calculate the distance the laser peak, (x, y), is from the center of a frame, giving us a measure of how far away from head-on our target camera is looking. Fig 6.10 shows how our model error changes as the target rotates away from head-on. We find that as the target looks further away from the probe our model error increases, this could be due to less signal at more extreme angles or because the spot changes less at larger angles. Interestingly, the difference between the two models increases at moderate angles while at small angles or extreme angles the difference is less. This implies that the CNN is able to pick up on more features in the moderate angles than the Zernike model is able to capture. Note, the error for both models near the edge of the FoV (0.5 normalized units) is close to half of the full FoV equivalent to random guessing.

Focusing: The results of both models' predictions for the focal length, f_t, and the object distance x_d are shown by testing set image number in Fig. 6.11. We then calculate the error for both f_t and x_d; the results are summarized in Table 6.1. Unsurprisingly, Res-Net-18 outperforms our Zernike model for both target parameters.

To further understand where focus prediction errors occur, we create a heatmap of the MAE for our Res-Net-18 model versus the true x_d and f_t. The heatmap is shown in Fig. 6.12, we find that the model struggles the most at low f_t. For x_d errors, this could be explained by the observation that lower f_t have larger depth of fields, making it harder to predict a specific x_d.

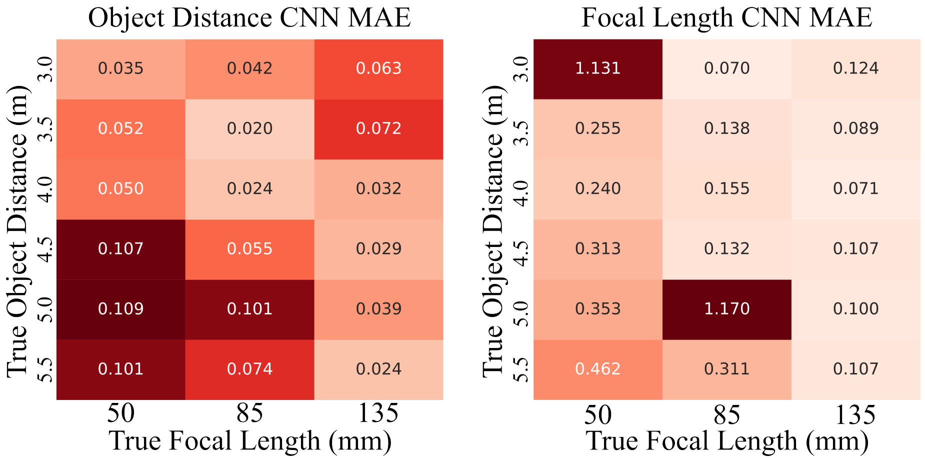


Figure 6.12: **Errors in Focusing Dataset:** This figure shows a heatmap of errors (MAE) for our Res-Net-18 model trained on the Focusing dataset. We show the errors for both the predicted object distance (x_d) (top) and the focal length (f_t) prediction (bottom) versus the true object distance and focal length. We find the model has the highest errors at the shortest focal length and far object distances.

Cellphone Results:

Due to the fact that many phone cameras have multiple lenses the simple Zernike model no longer holds, so we only train the Res-Net-18 model on the cellphone dataset. Similar to the Rotation dataset we predict the rotation of the cellphone. We also predict the cellphone model from one of the 11 cellphones we used. The results for these tasks are summarized in 6.1. We find that the problem of classifying the target phone model is quite easy as Res-Net-18 is able to predict correctly with 99% top-1 accuracy on the testing set. However, rotation prediction is more difficult likely due to lower signal levels and fewer lens aberrations but we are able to achieve error rates similar to the commercial lens dataset. We use the same normalized rotation error value as before, and found that Res-Net-18 achieves a MAE of 0.095 corresponding to half a meter of error for the Google Pixel 6a at 5.5m. We found that the target phone's model plays an important role in achievable accuracy, the rotation errors by phone model are summarized in Table 6.2. The highest error model is the Samsung Note 8 which has low signal levels of the diffuse RRs as

Table 6.2: This table shows the rotation error (MAE/RMSE) of the Res-Net-18 models prediction of cellphone rotation for each of the phone models used in our experiments. We find performance is dependent on the specific phone model used.

Phone Model	Rotation Error
iPhone XR	0.076/0.072
iPhone 10	0.061/0.052
iPhone 12	0.175/0.192
iPhone 12 Pro Max	0.027/0.026
iPhone 13	0.087 / 0.091
Google Pixel 6	0.104/0.122
Google Pixel 7	0.08 / 0.074
Samsung Note 8	0.186/0.21
Samsung SE F20	0.103/0.109
Samsung G7 Prime	0.056/0.05
Samsung A12	0.075/0.072
All Cellphones	0.095/0.11

seen in Fig. 6.8, whereas the lowest error phone is the iPhone 12 Pro Max which has two clear diffuse RRs in Fig. 6.8. The Google Pixel models have the strongest signals but are middle of the pack in terms of performance; however, the Google Pixel models exhibit only one clear diffuse RRs as opposed to many of the iPhone models where two diffuse RRs are visible. Likely, a combination of signal levels and specific RR properties (e.g. one or two diffuse RRs) play a key role in rotation prediction accuracies between phone models.

The main drawback of using this method to predict phone camera gaze is the lack of signal exhibited by certain phone models at large angles restricting the usable field of view to the middle 50%. This problem could possibly be mitigated through a careful selection of laser wavelength. It may also be possible to use many probes nearby each other that work together to cover a wide range of angles and could increase the overall accuracy of the method by combining results from each probe.

6.5 Discussion

In this work, we studied retro-reflections (RR) of target cameras and used RRs to predict imaging parameters. We found that RRs contain aberrations that can be used to predict a target camera's gaze, focal length, and depth of focus, as well as classify phone camera models. We only began to scratch the surface of all that can be done with RRs; for example, spectral content could be exploited with a change in laser wavelength. One idea is to probe the image sensor's spectral reflectance at multiple wavelengths to possibly distinguish specific sensor chips. The choice of laser wavelength(s) will be crucial for tasks involving cellphones, likely many wavelength optimizations can be done to improve performance over a population of cellphones; however, this may be difficult without optical element information from phone manufacturers. Another possible source of new features is to change the divergence of the laser to sweep a focused spot across specific slices of a target's optical path, probing the target along the optical axis.

Applications: This technology has applications in control, privacy, identification, and image validation. A RR sensing system could be used to create a network-free controller only requiring users to have a camera. Collecting user input remotely would usually require installation of software and would consume network resources which is avoided when using RR to determine camera view angle. For example, in a museum users could actively control an exhibit with a cellphone camera without connecting to a network, which provides a smoother user experience. Mass input from many users could be collected at performances where the audience is asked to take part in a vote by pointing the phone camera at a certain location. RR systems could be used for privacy, they could be used to detect cameras in a scene or determine if a camera is pointed at an object or person, whether pictures were taken, and estimate what the taken photograph will look like to help identify it when it appears elsewhere. This could be used to verify the authenticity of a picture, which is becoming a growing concern with the growth of generative AI. RRs contain information on the specific camera model so they could be used to identify which camera was used to take certain images which could be used in criminal or civil investigations. Another application could be protection of sensitive infrastructure of copyright protection; for example, RRs could be used to detect videos being captured in a movie theater.

Data availability. Data underlying the results presented in this paper are available in Seets, Trevor and Epstein, Alec and Velten, Andreas (2024).

Acknowledgements This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1747503, CAREER Grant No. 184688, and McPherson Eye Research Institute "Expanding our Vision" grant. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Support was also provided by the Graduate School and the Office of the Vice Chancellor for Research and Graduate Education at the University of Wisconsin-Madison with funding from the Wisconsin Alumni Research Foundation, and support was provided by the University of Wisconsin-Madison, Discovery to Product (D2P) with funding from the State of Wisconsin.

6.6 Supplementary Section: Notation and Operators

Here we provide the full expansions of the operators and functions used in our simulations from the main paper.

1. $\mathrm{Circ}_r(x, y)$ is a binary filled circle of radius r on the 2d plane:

$$\mathrm{Circ}_r(x, y) = \begin{cases} 1 & x^2 + y^2 \leqslant r^2 \\ 0 & \text{else} \end{cases} \tag{6.8}$$

2. $R[d](U)$ is the free space propagator from Goodman (2005),

$$R[d](U) = \frac{1}{\sqrt{i\lambda d}} \int_{-\infty}^{\infty} U(\mathbf{x}_1) e^{i \frac{k}{2d} ||\mathbf{x}_1 - \mathbf{x}_2||_2^2} d\mathbf{x}_1 \tag{6.9}$$

   where $\mathbf{x}_1$ is the input coordinate, $\mathbf{x}_2$ is the output coordinate, and $||\mathbf{x}||_2^2$ is the L^2 norm.

6.7 Supplementary Section: Simple Lens Result Images

Suppl. Fig 6.13 shows the results of shorter focal lengths in the simple lens experiments from Section 6.3. We found that at shorter focal lengths our simulations don’t match perfectly, likely due to a breakdown of the thin lens assumption. Although the shapes generated by our simulations are similar to shapes found in experiments they may not occur at exactly the same parameters, this reinforces the need for a data-driven approach.

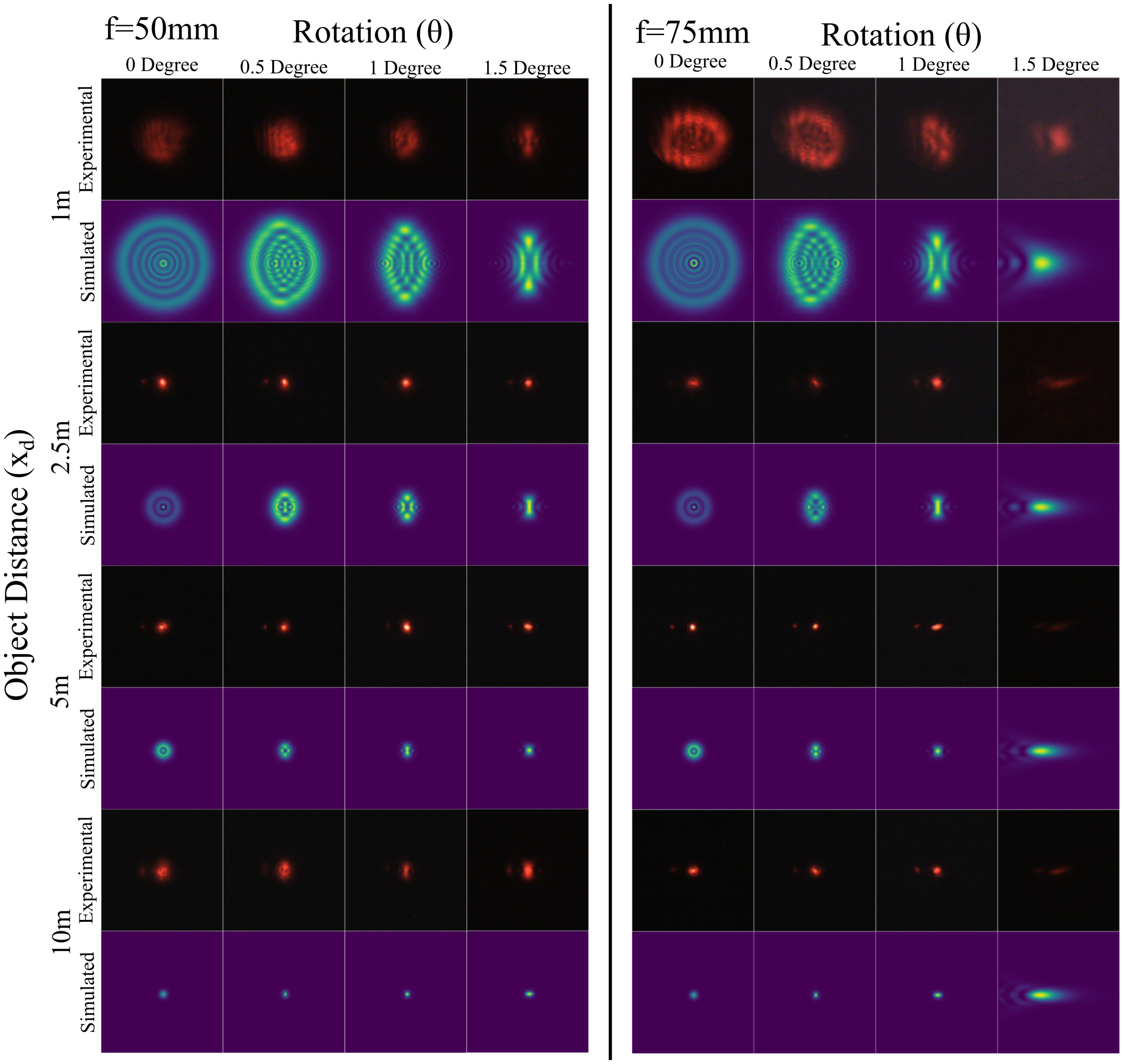


Figure 6.13: **Simple Lens Images:** This figure shows the results of our simple lens experiment for shorter focal lengths. At these focal lengths, the lens becomes quite large which is a likely reason our simulations (which assume thin lenses) no longer hold.

6.8 Supplementary Section: Simulation Distance Invariance

We simulate a variety of target camera parameters at different distances to verify that our simulation gives distance-invariant results. The results are shown in Fig. 6.14, note that the results are identical as distance changes for all camera parameter combinations tested. However due to aliasing, as the distance got large for very defocused beams we saw aliasing in the result. This could be solved by simulating a larger area with denser sampling; however, this would greatly increase the simulation time.

Distance invariance ensures that our returning RR signal only drops off with the illumination drop-off: there is no light lost during the propagation back to the probe. It also allows us to speed up simulation by simulating camera parameters at close distances letting us use fewer samples (i.e. smaller simulated area) to obtain good results.

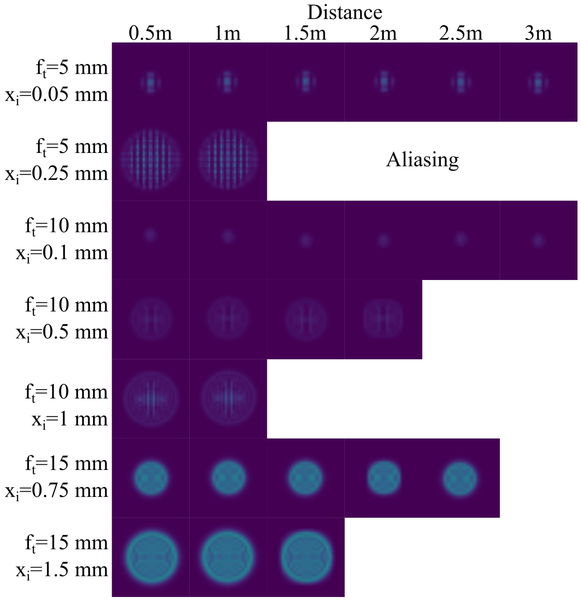


Figure 6.14: **Simulations at different distances:** This figure shows the result of simulations for a variety of different target camera parameters at different distances. The resulting images do not change with distance. Note for some parameter combinations there is aliasing at large distances.

7 CONCLUSION AND FUTURE WORK

In this dissertation, we examined a number of video denoising problems and the algorithms that can be used to remove noise. We began in Chapter 2 by considering a limiting case where each frame of video only contained at most one photon. In this case, a strong temporal prior is needed, we assumed that a piece-wise constant temporal representation was valid. This led us to developing the idea of changepoint detection in time which leads to an event representation of video which we use to deblur or denoise the video.

We then moved to a specific application of fluorescence guided surgery (FGS) where video denoising is crucial for expanding the usable contrast agents while maintaining real time performance. We examined two different noise models in Chapter 3 and 4 where found the importance of an accurate noise model for developing algorithms. In Chapter 4, we developed a realistic noise model for a specific commercial system and found the importance of modeling laser leakage light which results from imperfect hardware. We found a number of prior video denoising algorithms fail in the face of laser leakage light so we developed a set of new baseline algorithms for this problem. We found that the two dimensional noise space of laser leakage light and fluorescence signal create a complex space where different algorithms perform better under different conditions. Then in Chapter 5, we mapped this two dimensional noise space using the performance of our baseline models to define a cost function for FGS hardware. We examined the implications of this cost function for hardware design under a simplified parametric model of hardware components. We hope this work in FGS video denoising sets the stage for future development in this area and provides justification to work towards clinical translation of FGS denoising.

7.1 Future Outlook in FGS Denoising

On of the major areas of this dissertation was examining and setting the stage for video denoising in fluorescence guided surgery (FGS). While there are many minor technical improvements that can be made, the following are the major areas I see for bringing FGS denoising to clinical use.

Data and training: The crux of the translation will be in obtaining clinical data which is generally expensive and time consuming to gather. It likely won't be possible to gather clean data that can be used as ground truth, so a method that works with noisy data such as Noise2Noise Lehtinen et al. (2018) will be necessary to employ. However, most of these methods assume 0-mean noise which will not account for laser leakage light (LLL). For example, in Noise2Noise a pair of noisy images is used of the same scene for the input and target of a network during training, because the noise is uncorrelated in the two images the network will converge the mean value of images, which is noiseless. Dealing with LLL and understanding what data needs to be captured to remove it computationally will be the major challenge in training these models with clinical data.

I see a few viable solutions to getting usable data, first, LLL could be captured directly by a third camera with a different set of filters during clinical trials. While this additional camera may be complex and difficult to add, it will be much easier to add this camera to a single system rather than all systems. Then this LLL image can be used to remove the LLL bias (with a simple scaled subtraction or projection operator) of the noisy target image and a Noise2Noise training scheme can be used.

A second potentially viable option will be to use the full fluorescence video that contains LLL and attempt to correlate the reference video to the LLL portion. Because the fluorescence signal should be relatively uncorrelated to the reference video, it may be possible to train a network that predicts LLL by only using reference and fluorescence with LLL videos. The simplest version of this strategy is to simply train a network to predict the fluorescence video from the reference video, this prediction will only contain things that are correlated to the reference video, so may only contain LLL single. Note if fluorescence signal can be accurately predicted from the reference video then there is no need to use a fluorescence camera for this surgery type because enough information is contained in the reference video alone. An advantage of this training strategy is that is will not need hardware modifications but should validated in small human trials to confirm the LLL and reference videos are strongly correlated.

Efficient data collection is also a very important aspect of building these systems. One way to collect data is in a passive manner targeted for auto-fluorescent signals. Here the idea is to include fluorescence systems in all surgeries happening in a room, the system can capture lots of unstructured auto-fluorescence data without running smaller specific clinical trials targeted for a single surgery type. This will build up large databases to understand the correlation between auto-fluorescent, reference signals, and LLL signals. Once this large database is built it can be used to train a video denoiser with either of the above strategies. The main advantage of this data collection mechanism is large data collection with minimal cost and risk can be realized with low-profile systems designed to not interfere with surgeries. For example, these systems could be built into the lighting of the room.

Non-learning Denoisers: Another interesting area in FGS denoising is non-learning based methods that can span a time gap between now and adaption of learning based denoisers. The idea of a non-learning based denoiser is that it can be directly used on any video without training data allowing faster adoption. This could be used to improve image quality throughout clinical trials, possibly improving patient outcomes while collecting the data necessary to train a better denoiser. Finding a good non-learning based way to remove LLL is an interesting open problem. The main goal of this direction will to be providing an intermediate solution as data is gathered to build network based denoisers and to provide a lower risk solution that may have easier approval paths through the FDA.

High Dimensional Data: As higher dimensional data is used in fluorescence it will also be necessary to design new algorithms. For example, fluorescence lifetime will measure a few hundred time bins per pixel producing a very large dimensional video that will need to be reduced to a RGB video for human use. There are two viable ways forward here. First, as a long term goal, a classification network could be used that uses the higher dimensional input to classify tissue and outputs a colored map of the scene as if it was in a medical textbook. This will require labeled training data which adds another layer of complexity and cost to data gathering, especially in cases where the true state of tissue may be unknown. The second method to dealing high dimensional data relies on finding a suitable compressed space. In this method, first a network could learn to denoising the high dimensional data in the same way as a normal video. Then this denoised data will need to be compressed in a way to convey contrast within the scene. One method that may work is a principal component analysis or another learned basis from prior data. Finding this basis will be an interesting problem that will require working with surgeons and clinical outcome data to ensure a suitable choice is made.

7.2 Commercial Considerations in FGS

Most of this work has discussed the technology and design of FGS denoising methods. It is worth briefly discussing commercial considerations and potential paths for this technology or other technologies relying on machine learning (ML) models. The value of a model is in only what it can derive from the data; models should be considered as a value multiplier or extractor on data. Therefore, data is the key aspect of making these models viable, and ensuring the collected clinical data is useful should be done early on through simulations or other simple experiments. One of the most economically interesting properties of machine learning models is that their value increases with the amount of data used to train them. This implies that a firm with access to more data will be able to create a stronger model then other players, making their model more attractive to use, and thus creating more data for the original firm. This creates a positive feedback loop that will drive the value of a model quickly forward, and in some cases could allow previously unviable procedures to be viable with a strong general model. This implies, that when possible data should be aggregated to increase the value provided by ML models.

Next we consider three options for development of ML models and where in the

FGS technology stack they can be used. We consider three options: development with the contrast agent, integration with the imaging device, and a standalone ML firm. These three methods are not exhaustive nor exclusive, but are meant to illustrate some of the trade-offs in deciding where ML models should be used.

Development with Contrast Agent Research: In this option, ML models are trained during the research stage of a contrast agent. This allows the development of a contrast agent to be tied tightly to the power of a ML model. It garners 2 main advantages; first in contrast agent research, data is the most available as it is already being generated to test the efficacy of the candidate agent, so the ease of model training will be high. However, one potential problem is while data is available it may not be abundant, so the efficacy of an ML model only trained on a small dataset may be limited. The second advantage of integration of the ML model at this stage is it may save contrast agents from failing a clinical trial. The ML model may increase the sensitivity or specificity of the FGS system as a whole so may allow a contrast agent that would fail without it to succeed. This would potentially reduce contrast agent development timelines and cost. This method also carries the problem that if the contrast agent succeeds with the help of the ML model, that model and imaging system may need to be used in clinics. This will increase the bar for other imaging system to use the contrast agent because they will need to collect large enough datasets to train their own models to obtain comparable results.

Integration with the Imaging Device: In this strategy, ML models are developed by the imaging device manufactures. This has the advantage of using a specific system and noise model that will be consistent for the model. Also, device side models will naturally have access to a larger pool of surgery types, which leads to a higher ceiling in the total quantity of data available. There are at least two potential avenues for a device manufacture to gain access to data. First, device manufactures could consider partnering with research institutions where clinical trials are being run. The device manufacturer will supply a system and help conduct a clinical trial in exchange for data.

Second, a device manufacturer could supply pre-clinical and clinical imaging devices to contrast agent developers in exchange for data access. If done at scale with many contrast agent developers at once this strategy may require operating at a loss until the contrast agents are approved wherein the device manufacture will have a large data (and brand) lead on competitors allowing them to capture most of the market. This would also garner the advantages of ML integration at the contrast agent research level. Both of these data strategies can be employed in a way to obtain data from many different contrast agents allowing for a larger model to be developed which can be deployed on all imaging devices built by the manufacturer.

Standalone ML Firms: A standalone ML firm seeks to sell a model to all imaging device manufactures and contrast agent researchers. These firms would need to integrate data from many imaging devices and clinical trials, but they have the potential to provide the best performing models. The primary problem here is aligning the interest of the imaging device firms to convince them to share data with the ML firm. If a device manufacture may develop their own in house models, the gains from the ML firm's model would need to outweigh the disadvantages of giving the ML firm their data; for example, competitors will be able to gain access to a better model provided by the ML firm. If the scale of FGS data grows then it will potentially commercially make sense to have one general model if training of the models is prohibitively expensive to any individual device company.

Public Datasets: All of these strategies seek a balance between creating larger datasets and aligning the incentive structures of different firms in the technology stack. In general, firms do not need to create the best possible technology but rather one that works good enough or better than any competitor; this is fundamentally at odds with data sharing strategies. So while public datasets would improve technology for all firms, the largest players may be uninterested in sharing their data. One place public datasets could be employed is in publicly funded research; grants for research could ask for clinical data to be available in public repositories for use by any firm. This would then increase the overall value FGS adds to society which is in the interest of these funding agencies. Although of course, privacy and regulatory factors will need to be considered when setting up these datasets.

Regulation Risk:

By far the largest risk to a technology based on deep learning methods comes from regulatory bodies. Rightfully so, deep learning methods are far from well understood theoretically and can produce unstable output such as hallucination in some scenarios. Generally, the way to mitigate these risks with careful thought to data collection and validation methods. It will be the job of regulatory bodies to decide how best to test deep model outputs and ensure consistent results across changing hospital environments, and how to value potentially rare unstable model predictions. Abstractly, deep models will add risk and reward to many operations; it will be the job of regulatory officials to decide how to evaluate risk, and what reward is worth any increased risk. In the end, the regulators primary job is to set the rules to ensure safety and quality of care of the patient; .

REFERENCES

Adams, Ryan P., and David J. C. MacKay. 2007. Bayesian online changepoint detection. *arXiv: Machine Learning*.

Afshar, S., T. J. Hamilton, L. Davis, A. Van Schaik, and D. Delic. 2020. Event-based processing of single photon avalanche diode sensors. *IEEE Sensors Journal* 1–1.

Agresti, Alan, and Brent A. Coull. 1998. Approximate is better than "exact" for interval estimation of binomial proportions. *The American Statistician* 52(2):119–126.

Albano, Nicholas J., Weifeng Zeng, Christie Lin, Adam J. Uselmann, Kevin W. Eliceiri, and Samuel O. Poore. 2021. Augmentation of chicken thigh model with fluorescence imaging allows for real-time, high fidelity assessment in supermicrosurgery training. *Journal of reconstructive microsurgery* 37(6).

Alfonso-García, Alba, Xiangnan Zhou, Julien Bec, Silvia N Anbunesan, Farzad Fereidouni, Lee-Way Jin, Han S Lee, Orin Bloch, and Laura Marcu. 2022. First in patient assessment of brain tumor infiltrative margins using simultaneous time-resolved measurements of 5-ala-induced ppix fluorescence and tissue autofluorescence. *Journal of Biomedical Optics* 27(2):020501–020501.

Antonello, Jacopo, and Michel Verhaegen. 2015. Modal-based phase retrieval for adaptive optics. *J. Opt. Soc. Am. A* 32(6):1160–1170.

Arias, Pablo, and Jean-Michel Morel. 2017. Video denoising via empirical bayesian estimation of space-time patches. *Journal of Mathematical Imaging and Vision* 60:70–93.

Auclair, Michel, Yunlong Sheng, and Jean Fortin. 2013. Identification of targeting optical systems by multiwavelength retroreflection. *Optical Engineering* 52(5):054301.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Baer, Richard L. 2006. A model for dark current characterization and simulation. In *Sensors, cameras, and systems for scientific/industrial applications vii*, ed. Morley M. Blouke, vol. 6068, 606805. International Society for Optics and Photonics, SPIE.

Bardow, Patrick, Andrew J Davison, and Stefan Leutenegger. 2016. Simultaneous optical flow and intensity estimation from an event camera. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 884–892.

Barnes, J.A., and D.W. Allan. 1966. A statistical model of flicker noise. *Proceedings of the IEEE* 54(2):176–178.

Basseville, Michèle, Igor V Nikiforov, et al. 1993. *Detection of abrupt changes: theory and application*, vol. 104. Prentice Hall Englewood Cliffs.

Batson, Joshua, and Loic Royer. 2019. Noise2Self: Blind denoising by self-supervision. In *Proceedings of the 36th international conference on machine learning*, ed. Kamalika Chaudhuri and Ruslan Salakhutdinov, vol. 97 of *Proceedings of Machine Learning Research*, 524–533. PMLR.

Belykh, Evgenii, Eric J Miller, Arpan A Patel, Baran Bozkurt, Kaan Yağmurlu, Timothy R Robinson, Peter Nakaji, Robert F Spetzler, Michael T Lawton, Leonard Y Nelson, et al. 2018. Optical characterization of neurosurgical operating microscopes: quantitative fluorescence and assessment of ppix photobleaching. *Scientific reports* 8(1):12543.

Betz, CS, M Mehlmann, K Rick, H Stepp, G Grevers, R Baumgartner, and A Leunig. 1999. Autofluorescence imaging and spectroscopy of normal and malignant mucosa in patients with head and neck cancer. *Lasers in Surgery and Medicine: The Official Journal of the American Society for Laser Medicine and Surgery* 25(4):323–334.

Bradski, G. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Cao, Jiezhang, Qin Wang, Jingyun Liang, Yulun Zhang, Kai Zhang, Radu Timofte, and Luc Van Gool. 2023. Learning task-oriented flows to mutually guide feature alignment in synthesized and real video denoising. `2208.11803`.

Carr, Jessica A, Daniel Franke, Justin R Caram, Collin F Perkinson, Mari Saif, Vasileios Askoxylakis, Meenal Datta, Dai Fukumura, Rakesh K Jain, Moungi G Bawendi, et al. 2018. Shortwave infrared fluorescence imaging with the clinically approved near-infrared dye indocyanine green. *Proceedings of the National Academy of Sciences* 115(17):4465–4470.

Chan, Kelvin CK, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. 2021a. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 4947–4956.

———. 2021b. Understanding deformable alignment in video super-resolution. In *Proceedings of the aaai conference on artificial intelligence*, vol. 35, 973–981.

Chan, Kelvin CK, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. 2022a. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 5972–5981.

———. 2022b. On the generalization of basicvsr++ to video deblurring and denoising. *arXiv preprint arXiv:2204.05308*.

Charbonnier, P., L. Blanc-Feraud, G. Aubert, and M. Barlaud. 1994. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proceedings of 1st international conference on image processing*, vol. 2, 168–172 vol.2.

Chen, Liangyu, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. 2022. Simple baselines for image restoration. In *European conference on computer vision*, 17–33. Springer.

Chen, Liangyu, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. 2021. Hinet: Half instance normalization network for image restoration. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 182–192.

Chen, Xinyuan, Li Song, and Xiaokang Yang. 2016. Deep rnns for video denoising. In *Applications of digital image processing xxxix*, vol. 9971, 573–582. SPIE.

Cheng, Shen, Yuzhi Wang, Haibin Huang, Donghao Liu, Haoqiang Fan, and Shuaicheng Liu. 2021. Nbnet: Noise basis learning for image denoising with subspace projection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 4896–4906.

Dai, Jifeng, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable convolutional networks. In *Proceedings of the ieee international conference on computer vision*, 764–773.

Davy, Axel, Thibaud Ehret, Jean-Michel Morel, Pablo Arias, and Gabriele Facciolo. 2019. A non-local cnn for video denoising. In *2019 ieee international conference on image processing (icip)*, 2409–2413.

Demos, Stavros G, Regina Gandour-Edwards, Rajen Ramsamooj, and Ralph deVere White. 2004. Near-infrared autofluorescence imaging for detection of cancer. *Journal of biomedical optics* 9(3):587–592.

Dhulla, Vinit, Sapna S Mukherjee, Adam O Lee, Nanditha Dissanayake, Booshik Ryu, and Charles Myers. 2019. 256 x 256 dual-mode cmos spad image sensor. In *Advanced photon counting techniques xiii*, vol. 10978, 109780Q. International Society for Optics and Photonics.

Dobson, Geoffrey P. 2020. Trauma of major surgery: a global problem that is not going away.

DSouza, Alisha V, Huiyun Lin, Eric R Henderson, Kimberley S Samkoe, and Brian W Pogue. 2016. Review of fluorescence guided surgery systems: identification of key performance capabilities beyond indocyanine green imaging. *Journal of biomedical optics* 21(8):080901–080901.

Ester, Martin, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second international conference on knowledge discovery and data mining*, 226–231. KDD'96, AAAI Press.

Evangelidis, Georgios D, and Emmanouil Z Psarakis. 2008. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30(10):1858–1865.

Farnebäck, Gunnar. 2003. Two-frame motion estimation based on polynomial expansion. In *Scia*.

Fossum, Eric R, Jiaju Ma, Saleh Masoodian, Leo Anzagira, and Rachel Zizza. 2016. The quanta image sensor: Every photon counts. *Sensors* 16(8):1260.

FredvanGoor, guyskk, and jmmelko. 2019. Lightpipes. `https://github.com/opticspy/lightpipes/`.

Frumovitz, Michael, Marie Plante, Paula S Lee, Samith Sandadi, J F Lilja, Pedro F. Escobar, Lilian T. Gien, Diana L. Urbauer, and Nadeem R Abu-Rustum. 2018. Near-infrared fluorescence for detection of sentinel lymph nodes in women with cervical and uterine cancers (film): a randomised, phase 3, multicentre, non-inferiority trial. *The Lancet. Oncology* 19 10:1394–1403.

Gallego, G., T. Delbruck, G. M. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza. 2020. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1–1.

Gallego, Guillermo, Henri Rebecq, and Davide Scaramuzza. 2018. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Gastal, Eduardo S. L., and Manuel M. Oliveira. 2011. Domain transform for edge-aware image and video processing. *ACM TOG* 30(4):69:1–69:12. Proceedings of SIGGRAPH 2011.

Gehrig, Daniel, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. 2019. EKLT: Asynchronous, photometric feature tracking using events and frames. *Int. J. Comput. Vis.*

Gibbs-Strauss, Summer L., Khaled A. Nasr, Kenneth M. Fish, Onkar Khullar, Yoshitomo Ashitate, Tiberiu M. Siclovan, Bruce F. Johnson, Nicole E. Barnhardt, Cristina A. Tan Hehir, and John V. Frangioni. 2011. Nerve-highlighting fluorescent contrast agents for image-guided surgery. *Molecular imaging* 10(2):91–101.

Gnanasambandam, Abhiram, Omar Elgendy, Jiaju Ma, and Stanley H Chan. 2019. Megapixel photon-counting color imaging using quanta image sensor. *Optics express* 27(12):17298–17310.

Gong, Mali, and Sifeng He. 2017. Periodicity analysis on cat-eye reflected beam profiles of optical detectors. *Optical Engineering* 56(5):1 – 7.

Gong, Mali, Sifeng He, Rui Guo, and Wei Wang. 2016. Cat-eye effect reflected beam profiles of an optical system with sensor array. *Appl. Opt.* 55(16):4461–4466.

Goodman, Joseph W. 2005. Introduction to fourier optics. *Introduction to Fourier optics, 3rd ed., by JW Goodman. Englewood, CO: Roberts & Co. Publishers, 2005* 1.

Gustafson, Tiffany, Sergey Dergunov, Walter Akers, Qian Cao, Selena Magalotti, Samuel Achilefu, Eugene Pinkhassik, and Mikhail Berezin. 2013. Blood triggered rapid release porous nanocapsules. *RSC advances* 3:5547–5555.

Gyöngy, I., T. A. Abbas, N. Dutton, and R. Henderson. 2017. Object tracking and reconstruction with a quanta image sensor. In *Proceedings of the international image sensor workshop*.

Gyongy, Istvan, Neale AW Dutton, and Robert K Henderson. 2018. Single-photon tracking for high-speed vision. *Sensors* 18(2):323.

Han, Qi, Zejia Fan, Qi Dai, Lei Sun, Ming-Ming Cheng, Jiaying Liu, and Jingdong Wang. 2021. Demystifying local vision transformer: Sparse connectivity, weight sharing, and dynamic weight. *arXiv preprint arXiv:2106.04263* 2(3).

Hasinoff, Samuel W. 2014. *Photon, poisson noise*, 608–610. Boston, MA: Springer US.

Hasinoff, Samuel W, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. 2016. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (TOG)* 35(6):1–12.

He, Kaiming, Jian Sun, and Xiaoou Tang. 2012. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence* 35(6):1397–1409.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 ieee conference on computer vision and pattern recognition (cvpr)*, 770–778.

He, Sifeng, Yuan Meng, and Mali Gong. 2018. Active laser detection system for recognizing surveillance devices. *Optics Communications* 426:313–324.

Hoerl, Arthur E., and Robert W. Kennard. 2000. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 42(1):80–86.

Hu, Jie, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 7132–7141.

Huang, Jiahao, Haiyang Zhang, Lin Wang, Zilong Zhang, and Changming Zhao. 2021. Improved yolov3 model for miniature camera detection. *Optics and Laser Technology* 142:107133.

Huang, Yan, Wei Wang, and Liang Wang. 2015. Bidirectional recurrent convolutional networks for multi-frame super-resolution. *Advances in neural information processing systems* 28.

Ingle, Atul, Andreas Velten, and Mohit Gupta. 2019. High flux passive imaging with single-photon sensors. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 6760–6769.

Instruments, Lambert. 2023.

Ishizawa, Takeaki, Noriyoshi Fukushima, Junji Shibahara, Koichi Masuda, Sumihito Tamura, Taku Aoki, Kiyoshi Hasegawa, Yoshifumi Beck, Masashi Fukayama, and Norihiro Kokudo. 2009. Real-time identification of liver cancers by using indocyanine green fluorescent imaging. *Cancer* 115(11):2491–2504. `https://acsjournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cncr.24291`.

Istvan, Gyongy, Dutton Neale, Luca Parmesan, Davies Amy, Saleeb Rebecca, Duncan Rory, Rickman Colin, Dalgarno Paul, and Robert K Henderson. 2015. Bit-plane processing techniques for low-light, high speed imaging with a spad-based qis. In *International image sensor workshop*, 1–4.

Killick, R., P. Fearnhead, and I. A. Eckley. 2012. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association* 107(500):1590–1598. `https://doi.org/10.1080/01621459.2012.737745`.

Kim, Sung Won, Seo Hyun Song, Hyoung Shin Lee, Woong Jae Noh, Chulho Oak, Yeh-Chan Ahn, and Kang Dae Lee. 2016. Intraoperative Real-Time Localization of Normal Parathyroid Glands With Autofluorescence Imaging. *The Journal of Clinical Endocrinology and Metabolism* 101(12):4646–4652. `https://academic.oup.com/jcem/article-pdf/101/12/4646/10528203/jcem4646.pdf`.

Kingma, Diederik P., and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd international conference on learning representations, ICLR 2015, san diego, ca, usa, may 7-9, 2015, conference track proceedings*, ed. Yoshua Bengio and Yann LeCun.

Kitai, Toshiyuki, Takuya Inomoto, Mitsuharu Miwa, and Takahiro Shikayama. 2005. Fluorescence navigation with indocyanine green for detecting sentinel lymph nodes in breast cancer. *Breast Cancer* 12(3):211–215.

Komorowska-Timek, Ewa, and Geoffrey C Gurtner. 2010. Intraoperative perfusion mapping with laser-assisted indocyanine green imaging can predict and prevent complications in immediate breast reconstruction. *Plastic and reconstructive surgery* 125(4):1065–1073.

Konnik, Mikhail, and James Welsh. 2014. High-level numerical simulations of noise in ccd and cmos photosensors: review and tutorial. *arXiv preprint arXiv:1412.4031*.

Krull, Alexander, Tim-Oliver Buchholz, and Florian Jug. 2019. Noise2void-learning denoising from single noisy images. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2129–2137.

Kulick, Johannes, nariox, Dan Marthaler, Minesh A. Jethva, and Sean Kruzel. 2020. bayesian changepoint detection. `https://github.com/hildensia/bayesian_changepoint_detection`.

Lai, Wei-Sheng, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. 2017. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 624–632.

Lakowicz, Joseph R. 2006. *Principles of fluorescence spectroscopy*. Springer.

Lakshminarayanan, Vasudevan, and Andre Fleck. 2011. Zernike polynomials: A guide. *Journal of Modern Optics - J MOD OPTIC* 58:1678–1678.

Laurenzis, Martin. 2019. Single photon range, intensity and photon flux imaging with kilohertz frame rate and high dynamic range. *Optics Express* 27(26):38391–38403.

Leal-Junior, Arnaldo, Leticia Avellar, Vitorino Biazi, M Simone Soares, Anselmo Frizera, and Carlos Marques. 2022. Multifunctional flexible optical waveguide sensor: On the bioinspiration for ultrasensitive sensors development. *Opto-Electronic Advances* 5(10):210098–1.

Leal-Junior, Arnaldo, Leticia Avellar, Anselmo Frizera, and Carlos Marques. 2020. Smart textiles for multimodal wearable sensing using highly stretchable multiplexed optical fiber system. *Scientific Reports* 10(1):13867.

Lecocq, Christophe, Gilles Deshors, Olga Lado-Bordowsky, and Jean-Louis Meyzonnette. 2003. Sight laser detection modeling. In *Laser radar technology and applications viii*, ed. Gary W. Kamerman, vol. 5086, 280 – 286. International Society for Optics and Photonics, SPIE.

Lehtinen, Jaakko, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. 2018. Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*.

Levin, Anat, Peter Sand, Taeg Sang Cho, Frédo Durand, and William T Freeman. 2008. Motion-invariant photography. *ACM Transactions on Graphics (TOG)* 27(3): 1–9.

Li, Li, Jianlin Ren, and Xingbin Wang. 2015. Fast cat-eye effect target recognition based on saliency extraction. *Optics Communications* 350:33–39.

Li, Yiwen, Ying Song, Lian Zhao, Gabriel Gaidosh, Alan M Laties, and Rong Wen. 2010. Direct labeling and visualization of blood vessels with lipophilic carbocyanine dye dii. *Nature protocols*.

Liu, Chun, Chang Zhao, Hai Zhang, Zilong Zhang, Shuyuan Gao, and Yunshi Wang. 2019a. Analysis of mini-camera's cat-eye retro-reflection for characterization of diffraction rings and arrayed spots. *IEEE Photonics Journal* PP:1–1.

Liu, Chun, Changming Zhao, Haiyang Zhang, Zilong Zhang, Zitao Cai, and Zhipeng Li. 2019b. Spectrum classification using convolutional neural networks for a mini-camera detection system. *Appl. Opt.* 58(33):9230–9239.

Liu, Chun, Changming Zhao, Haiyang Zhang, Zilong Zhang, Yanwang Zhai, and Yali Zhang. 2019c. Design of an active laser mini-camera detection system using cnn. *IEEE Photonics Journal* 11(6):1–12.

Liu, Lu, Shangfeng Wang, Baozhou Zhao, Peng Pei, Yong Fan, Xiaomin Li, and Fan Zhang. 2018. Er3+ sensitized 1530 nm to 1180 nm second near-infrared window upconversion nanocrystals for in vivo biosensing. *Angewandte Chemie* 130(25): 7640–7644.

Liu, Zhuang, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A convnet for the 2020s. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 11976–11986.

Ma, Sizhuo, Shantanu Gupta, Arin C. Ulku, Claudio Bruschini, Edoardo Charbon, and Mohit Gupta. 2020. Quanta burst photography. *ACM Trans. Graph.* 39(4).

Maggioni, Matteo, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. 2012. Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society* 21:3952–66.

Maggioni, Matteo, Yibin Huang, Cheng Li, Shuai Xiao, Zhongqian Fu, and Fenglong Song. 2021. Efficient multi-stage video denoising with recurrent spatio-temporal fusion. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 3466–3475.

Marcu, Laura. 2012. Fluorescence lifetime techniques in medical applications. *Annals of biomedical engineering* 40:304–331.

Mieremet, Arjan L., Ric M. A. Schleijpen, and P. N. Pouchelle. 2008. Modeling the detection of optical sights using retro-reflection. In *Laser radar technology and applications xiii*, ed. Monte D. Turner and Gary W. Kamerman, vol. 6950, 69500E. International Society for Optics and Photonics, SPIE.

Morimoto, Kazuhiro, Andrei Ardelean, Ming-Lo Wu, Arin Can Ulku, Ivan Michel Antolovic, Claudio Bruschini, and Edoardo Charbon. 2020. Megapixel time-gated spad image sensor for 2d and 3d imaging applications. *Optica* 7(4):346–354.

Noble Anbunesan, Silvia, Alba Alfonso-Garcia, Xiangnan Zhou, Julien Bec, Han Sung Lee, Lee-Way Jin, Orin Bloch, and Laura Marcu. 2023. Intraoperative detection of idh-mutant glioma using fluorescence lifetime imaging. *Journal of Biophotonics* 16(4):e202200291.

Olson, Madeline T, Quan P Ly, and Aaron M Mohs. 2019. Fluorescence guidance in surgical oncology: challenges, opportunities, and translation. *Molecular imaging and biology* 21:200–218.

Orosco, Ryan K, Viridiana J Tapia, Joseph A Califano, Bryan Clary, Ezra EW Cohen, Christopher Kane, Scott M Lippman, Karen Messer, Alfredo Molinolo, James D Murphy, et al. 2018. Positive surgical margins in the 10 most common solid cancers. *Scientific reports* 8(1):5686.

Ostrowski, Piotr Kopa, Efklidis Katsaros, Daniel Węsierski, and Anna Jezierska. 2022. Bp-evd: Forward block-output propagation for efficient video denoising. *IEEE Transactions on Image Processing* 31:3809–3824.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Pegg, Timothy J, Daniel K Gladish, and Robert L Baker. 2021. Algae to angiosperms: Autofluorescence for rapid visualization of plant anatomy among diverse taxa. *Applications in Plant Sciences* 9(6):e11437.

Pogue, Brian, Timothy Zhu, Vasilis Ntziachristos, Brian Wilson, Keith Paulsen, Sylvain Gioux, Robert Nordstrom, Joshua Pfefer, Bruce Tromberg, Heidrun Wabnitz, et al. 2023. Guidance for performance evaluation for fluorescence guided surgery systems: evaluation for fluorescence guided surgery systems.

Pont-Tuset, Jordi, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. 2017. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*.

Qi, Chenyang, Junming Chen, Xin Yang, and Qifeng Chen. 2022. Real-time streaming video denoising with bidirectional buffers. In *Proceedings of the 30th acm international conference on multimedia*, 2758–2766.

Qian, Feng, Bao Zhang, Chuanli Yin, Mingyu Yang, and Xiantao Li. 2015. Recognition of interior photoelectric devices by using dual criteria of shape and local texture. *Optical Engineering* 54(12):1 – 10.

Raskar, Ramesh, Amit Agrawal, and Jack Tumblin. 2006. Coded exposure photography: motion deblurring using fluttered shutter. In *Acm siggraph 2006 papers*, 795–804. ACM.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–miccai 2015: 18th international conference, munich, germany, october 5-9, 2015, proceedings, part iii 18*, 234–241. Springer.

Ruiz, Alberto J, Mindy Wu, Ethan PM LaRochelle, Dimitris Gorpas, Vasilis Ntziachristos, T Joshua Pfefer, and Brian W Pogue. 2020. Indocyanine green matching phantom for fluorescence-guided surgery imaging system characterization and performance assessment. *Journal of Biomedical Optics* 25(5):056003–056003.

Sami, Sriram, Sean Rui Xiang Tan, Bangjie Sun, and Jun Han. 2021. Lapd: Hidden spy camera detection using smartphone time-of-flight sensors. In *Proceedings of the 19th acm conference on embedded networked sensor systems*, 288–301. SenSys '21, New York, NY, USA: Association for Computing Machinery.

Seets, Trevor, Alec Epstein, and Andreas Velten. 2024a. Watching the watchers: camera identification and characterization using retro-reflections. *Optics Express* 32(8):13836–13850.

Seets, Trevor, Atul Ingle, Martin Laurenzis, and Andreas Velten. 2021. Motion adaptive deblurring with single-photon cameras. In *Proceedings of the ieee/cvf winter conference on applications of computer vision (wacv)*, 1945–1954.

Seets, Trevor, Wei Lin, Yizhou Lu, Christie Lin, Adam Uselmann, and Andreas Velten. 2024b. Ofdvdnet: A sensor fusion approach for video denoising in fluorescence-guided surgery. In *Medical imaging with deep learning*, ed. Ipek Oguz, Jack Noble, Xiaoxiao Li, Martin Styner, Christian Baumgartner, Mirabela Rusu, Tobias Heinmann, Despina Kontos, Bennett Landman, and Benoit Dawant, vol. 227 of *Proceedings of Machine Learning Research*, 1564–1580. PMLR.

Seets, Trevor and Epstein, Alec and Velten, Andreas. 2024. Watching the Watchers: Camera Identification and Characterization using Retro-reflections - Dataset [Dataset]. `https://doi.org/10.5061/dryad.6t1g1jx64`.

Snoeij, Martijn F., Albert J. P. Theuwissen, Kofi A. A. Makinwa, and Johan H. Huijsing. 2006. A cmos imager with column-level adc using dynamic column fixed-pattern noise reduction. *IEEE Journal of Solid-State Circuits* 41(12):3007–3015.

Stock, Johannes, Norman Girma Worku, and Herbert Gross. 2017. Coherent field propagation between tilted planes. *J. Opt. Soc. Am. A* 34(10):1849–1855.

Stoffregen, Timo, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. 2019. Event-based motion segmentation by motion compensation. In *Proceedings of the ieee international conference on computer vision*, 7244–7253.

Sutton, Paul A, Martijn A van Dam, Ronan A Cahill, Sven Mieog, Karol Polom, Alexander L Vahrmeijer, and Joost van der Vorst. 2023. Fluorescence-guided surgery: comprehensive review. *BJS open* 7(3):zrad049.

Svedbrand, Daniel, Lars Allard, Magnus Pettersson, Pontus Köhler, Markus Henriksson, and Lars Sjökvist. 2019. Optics detection using an avalanche photo diode array and the scanning-slit-method. In *Technologies for optical countermeasures xvi*, ed. David H. Titterton, Robert J. Grasso, and Mark A. Richardson, vol. 11161, 167 – 177. International Society for Optics and Photonics, SPIE.

Tang, Yue, I Leah Gitajn, Xu Cao, Xinyue Han, Jonathan T Elliott, Xiaohan Yu, Logan M Bateman, Bethany S Malskis, Lillian A Fisher, Jessica M Sin, et al. 2023. Automated motion artifact correction for dynamic contrast-enhanced fluorescence imaging during open orthopedic surgery. In *Molecular-guided surgery: Molecules, devices, and applications ix*, vol. 12361, 17–20. SPIE.

Tango, William J. 1997. The circle polynomials of zernike and their application in optics. *Applied physics*.

Tartakovsky, Alexander, Igor Nikiforov, and Michele Basseville. 2014. *Sequential analysis: Hypothesis testing and changepoint detection*. CRC Press.

Tassano, Matias, Julie Delon, and Thomas Veit. 2019. Dvdnet: A fast network for deep video denoising. In *2019 ieee international conference on image processing (icip)*, 1805–1809.

———. 2020. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (cvpr)*.

Technologies, Princeton Lightwave/AMS. 2012 (accessed June 20, 2020). *32 x 32 geiger-mode avalanche photodiode (gmapd) camera*. `http://www.amstechnologies.com/fileadmin/amsmedia/downloads/4796_gmapdcameradatasheet.pdf`.

Truong, C. 2017 (accessed June 20, 2020). *ruptures python package*. `https://ctruong.perso.math.cnrs.fr/ruptures-docs/build/html/index.html`.

Truong, Charles, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Processing* 167:107299.

Tu, Zhengzhong, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. 2022. Maxim: Multi-axis mlp for image processing. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 5769–5780.

Vahrmeijer, Alexander L, Merlijn Hutteman, Joost R Van Der Vorst, Cornelis JH Van De Velde, and John V Frangioni. 2013. Image-guided cancer surgery using near-infrared fluorescence. *Nature reviews Clinical oncology* 10(9):507–518.

Vaksman, Gregory, Michael Elad, and Peyman Milanfar. 2021. Patch craft: Video denoising by deep modeling and patch matching. In *Proceedings of the ieee/cvf international conference on computer vision*, 2157–2166.

Van Veen, Dave, Ben A Duffy, Long Wang, Keshav Datta, Tao Zhang, Greg Zaharchuk, and Enhao Gong. 2021. Real-time video denoising to reduce ionizing radiation exposure in fluoroscopic imaging. In *International workshop on machine learning for medical image reconstruction*, 109–119. Springer.

Velten, A., A.J. Uselmann, S. Prajapati, J.S. Bredfeldt, T.R. Mackie, and K.W. Eliceiri. 2020. Transient room lighting for ambient light multiphoton microscopy.

Wang, Wei, Xin Chen, Cheng Yang, Xiang Li, Xuemei Hu, and Tao Yue. 2019a. Enhancing low light videos by exploring high sensitivity camera noise. In *2019 ieee/cvf international conference on computer vision (iccv)*, 4110–4118.

Wang, Xintao, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. 2019b. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition workshops*, 0–0.

Wang, Zhendong, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. 2022. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 17683–17693.

Wang, Zhou, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004a. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4):600–612.

Wang, Zhou, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004b. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13(4):600–612.

Wei, Kaixuan, Ying Fu, Jiaolong Yang, and Hua Huang. 2020. A physics-based noise formation model for extreme low-light raw denoising. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2758–2767.

Weyers, Brent W, Andrew C Birkeland, Mark A Marsden, Athena Tam, Julien Bec, Roberto P Frusciante, Dorina Gui, Arnaud F Bewley, Marianne Abouyared, Laura Marcu, et al. 2022. Intraoperative delineation of p16+ oropharyngeal carcinoma of unknown primary origin with fluorescence lifetime imaging: Preliminary report. *Head & neck* 44(8):1765–1776.

Wu, Kejun, Hongqiao Zhang, Yanxu Chen, Qian Luo, and Kaikai Xu. 2021. All-silicon microdisplay using efficient hot-carrier electroluminescence in standard 0.18um cmos technology. *IEEE Electron Device Letters* 42(4):541–544.

Wyant, James, and Katherine Creath. 1992. Basic wavefront aberration theory for optical metrology. *Appl Optics Optical Eng* 11.

Xiang, Liuyu, Jundong Zhou, Jirui Liu, Zerun Wang, Haidong Huang, Jie Hu, Jungong Han, Yuchen Guo, and Guiguang Ding. 2022. Remonet: Recurrent multi-output network for efficient video denoising. In *Proceedings of the aaai conference on artificial intelligence*, vol. 36, 2786–2794.

Xie, Bangwen, Marieke A. Stammes, Pieter B. A. A. van Driel, Luis J. Cruz, Vicky T. Knol-Blankevoort, Martijn A. M. Löwik, Laura Mezzanotte, Ivo Que, Alan Chan, Jeroen P. H. M. van den Wijngaard, Maria Siebes, Sven Gottschalk, Daniel Razansky, Vasilis Ntziachristos, Stijn Keereweer, Richard W. Horobin, Mathias Hoehn, Eric L. Kaijzel, Ermond R. van Beek, Thomas J. A. Snoeks, and Clemens W. G. M. Löwik. 2015. Necrosis avid near infrared fluorescent cyanines for imaging cell death and their use to monitor therapeutic efficacy in mouse tumor models. *Oncotarget* 6(36): 39036–39049.

Xu, Kaikai. 2021. Silicon electro-optic micro-modulator fabricated in standard cmos technology as components for all silicon monolithic integrated optoelectronic systems. *Journal of Micromechanics and Microengineering* 31(5):054001.

Xu, Xiangyu, Muchen Li, Wenxiu Sun, and Ming-Hsuan Yang. 2020. Learning spatial and spatio-temporal pixel aggregations for image and video denoising. *IEEE Transactions on Image Processing* 29:7153–7165.

Yoshimatsu, Hidehiko, Ryo Karakawa, Mario F Scaglioni, Yuma Fuse, Kenta Tanakura, and Tomoyuki Yano. 2021. Application of intraoperative indocyanine green angiography for detecting flap congestion in the use of free deep inferior epigastric perforator flaps for breast reconstruction. *Microsurgery* 41(6):522–526.

Yue, Huanjing, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang. 2020. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2301–2310.

Zajac, Jocelyn C, Aiping Liu, Adam J Uselmann, Christie Lin, Sameeha E Hassan, Lee D Faucher, and Angela Lf Gibson. 2022. Lighting the way for necrosis excision through indocyanine green fluorescence-guided surgery. *Journal of the American College of Surgeons* 235(5):743—755.

Zamir, Syed Waqas, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2022. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 5728–5739.

Zamir, Syed Waqas, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. 2020. Learning enriched features for real image restoration and enhancement. In *Computer vision–eccv 2020: 16th european conference, glasgow, uk, august 23–28, 2020, proceedings, part xxv 16*, 492–511. Springer.

———. 2021. Multi-stage progressive image restoration. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 14821–14831.

Zhang, Lin, Lei Zhang, Xuanqin Mou, and David Zhang. 2011. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing* 20(8):2378–2386.

Zhang, Ray R., Alexandra B. Schroeder, Joseph J. Grudzinski, Eben L. Rosenthal, Jason M. Warram, Anatoly N. Pinchuk, Kevin W. Eliceiri, John S. Kuo, and Jamey P. Weichert. 2017. Beyond the margins: real-time detection of cancer using targeted fluorophore. *Nature reviews. Clinical oncology*.

Zhang, Richard, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 586–595.

Zhang, Yi, Dasong Li, Xiaoyu Shi, Dailan He, Kangning Song, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. 2023. Kbnet: Kernel basis network for image restoration. *arXiv preprint arXiv:2303.02881*.

Zhang, Yi, Hongwei Qin, Xiaogang Wang, and Hongsheng Li. 2021. Rethinking noise synthesis and modeling in raw denoising. In *Proceedings of the ieee/cvf international conference on computer vision*, 4593–4601.

Zhao, Shengyu, Yilun Sheng, Yue Dong, Eric I-Chao Chang, and Yan Xu. 2020. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Ieee/cvf conference on computer vision and pattern recognition (cvpr)*.

Zhu, Lin, Siwei Dong, Jianing Li, Tiejun Huang, and Yonghong Tian. 2020. Retina-like visual image reconstruction via spiking neural model. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 1438–1446.

Zhu, Xizhou, Han Hu, Stephen Lin, and Jifeng Dai. 2019. Deformable convnets v2: More deformable, better results. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 9308–9316.