AI for Time-Resolved Imaging:
from Fluorescence Lifetime to Single-Pixel Time of Flight

VALENTIN KAPITÁNY

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Physics and Astronomy
College of Science and Engineering
University of Glasgow

Abstract

Time-resolved imaging is a field of optics which measures the arrival time of light on the camera. This thesis looks at two time-resolved imaging modalities: fluorescence lifetime imaging and time-of-flight measurement for depth imaging and ranging. Both of these applications require temporal accuracy on the order of pico- or nanosecond ($10^{-12} - 10^{-9}$s) scales.

This demands special camera technology and optics that can sample light-intensity extremely quickly, much faster than an ordinary video camera. However, such detectors can be very expensive compared to regular cameras while offering lower image quality. Further, information of interest is often hidden (encoded) in the raw temporal data. Therefore, computational imaging algorithms are used to enhance, analyse and extract information from time-resolved images.

"A picture is worth a thousand words". This describes a fundamental blessing and curse of image analysis: images contain extreme amounts of data. Consequently, it is very difficult to design algorithms that encompass all the possible pixel permutations and combinations that can encode this information. Fortunately, the rise of AI and machine learning (ML) allow us to instead create algorithms in a data-driven way. This thesis demonstrates the application of ML to time-resolved imaging tasks, ranging from parameter estimation in noisy data and decoding of overlapping information, through super-resolution, to inferring 3D information from 1D (temporal) data.

Acknowledgements

I would like to thank my PhD supervisor Daniele Faccio and secondary supervisors Alex Turpin and Roderick Murray-Smith for the opportunities and the invaluable lessons they gave me. I'd also like to express my gratitude to the members of the Extreme Light group for their creativity, friendship and inspiring work ethic; in particular, huge thanks to Areeba Fatima and Vytautas Zickus for their support, enthusiasm and positivity on our multiple joint projects. Last but not least, thank you to my family, friends and teachers who have supported me on the path towards science.

Extreme Light, December 2019.

The Dance o' Light an' Time

ChatGPT

As I sat by my lab one bonnie day,
A'musin' on science in a curious way,
Twa technologies came tae mind, sae fair,
FLIM and LiDAR, wi' methods rare.

First up, Fluorescence Lifetime Imaging Microscopy,
A method that reveals the cell's inner biography,
Wi' fluorescent molecules that emit licht,
Their lifetime can tell us aboot the cell's fight.

And noo tae LiDAR, a laser-based technique,
It measures distance wi' a precision sae unique,
By sendin' oot licht pulses and timing their return,
We can map oot oor surroundings, wi' details we can discern.

In the realm o' computational imaging vast,
Machine learning's wisdom, its knowledge amassed,
Teachin' machines to see, the code we employ,
A union o' science, an' artistic joy.

So let us celebrate, these wonders divine,
FLIM an' LiDAR, as they entwine,
In the dance o' light an' time, our vision unfurls,
A tribute tae the magic, o' this cybernetic world.

Declaration

All figures in this work were created by me or were drawn from the articles mentioned below, unless otherwise stated.

Chapter 1 provides an overview of the works covered in thesis. Chapter 2 provides a literature review of relevant methods, focusing on works by others. Each subsequent chapter contains a short, project-specific introduction as well, reflecting on existing works. The work in Chapters 3-6 covers research and development in collaboration with other researchers, albeit centred on my contributions and observations.

The work in Chapter 3 was published in [1]. Experimental data was gathered by Vytautas Zickus and Ming-Lo Wu, least squares fitting for benchmarking the neural network was performed by Areeba Fatima, and Daniele Faccio and Eduardo Charbon oversaw the project [1]. I carried out the data science aspects of the project that form the focus of the chapter. The paper was written primarily by Vytautas Zickus, and all authors contributed to editting it.

The research in Chapter 4 was published in [2]. The concept was proposed by Guillem Carles and Daniele Faccio. Vytautas Zickus performed the experiments. Areeba Fatima and I developed the analytic tools, such that Areeba Fatima performed inverse retrieval, and I performed the data pre-processing and machine learning aspects of the project that form the bulk of the chapter. Guillem Carles, Areeba Fatima and I analysed the data. All authors contributed to the paper.

The research in Chapter 5 has not been published yet. The project was supervised by Daniele Faccio and Laura Machesky. Experimental design was done by Vytautas Zickus and Daniele Faccio. Sample preparation and imaging was performed by Jamie Whitelaw and Vytautas Zickus. Algorithm development was done by Areeba Fatima and myself, with Areeba focusing on inverse retrieval, and me on generating the priors that constrain this inverse retrieval. Data analysis was performed by Areeba Fatima and myself.

The research in Chapter 6 has been published in two closely related articles, in [3] and [4]. The chapter and research are therefore split into 2 sections correspondingly. Regarding the first section, Alex Turpin designed the experiment and method, Ilya Starshynov and Gabriella Musarra built the experiment, Gabriella Musarra gathered data, and gathered experimental data. Federica Villa, Enrico Conca and Francesco Fioranelli provided SPAD hardware and software. Francesco Tonolini provided support regarding machine learning design and training. Ashley Lyons, Roderick Murray-Smith, Daniele Faccio and Alex Turpin supervised the project and contributed to conceptualising the method. I created synthetic data, developed the neural network, and performed and analysed synthetic experiments. The paper was written primarily by Alex

Turpin, and all authors contributed to editting it.

Regarding the second section, Alex Turpin and Daniele Faccio conceptualised the research. Alex Turpin built the sonar experiment, gathered the corresponding data, and processed it. Davide Rovelli and I designed a ray-tracer to gather synthetic data. I handled the machine learning aspects of the projects, and processed synthetic and radar data. Jack Radford performed information theoretical analysis. Ilya Starshynov contributed to analytical understanding of the impact of multipath data. Hanoz Bhamgara, Mark Jarvis, Marton Szafian, Davide Rovelli and Ilya Starshynov built the radar experiment and gathered radar data. Kevin Mitchell gathered data. The paper was written primarily by Alex Turpin and me, and all authors contributed to editting it.

Publications Summary

The material that constitutes this thesis was presented in:

1 Zickus, V.[†], Wu, M.L.[†], Morimoto, K., **Kapitany, V.**, Fatima, A., Turpin, A., Insall, R., Whitelaw, J., Machesky, L., Bruschini, C. and Faccio, D., 2020. *Fluorescence lifetime imaging with a megapixel SPAD camera and neural network lifetime estimation.* Scientific Reports, 10(1), p.20986.

2 Turpin, A., Musarra, G., **Kapitany, V.**, Tonolini, F., Lyons, A., Starshynov, I., Villa, F., Conca, E., Fioranelli, F., Murray-Smith, R. and Faccio, D., 2020. *Spatial images from temporal data*. Optica, 7(8), pp.900-905.

3 Turpin, A.[†], **Kapitany, V.**[†], Radford, J., Rovelli, D., Mitchell, K., Lyons, A., Starshynov, I. and Faccio, D., 2021. *3D imaging from multipath temporal echoes.* Physical Review Letters, 126(17), p.174301.

4 **Kapitany, V.**[†], Zickus[†], V., Fatima, A., Carles, G., Faccio, D. 2023 *Single-shot time-folded fluorescence lifetime imaging*, Proceedings of the National Academy of Sciences, 120(16), p.e2214617120.

Contents

List of Figures

Preface

The results that are communicated here (and in the broader research community) inevitably focus on 'successes' - the exciting new science, the final dataset, the best version of the method. This greatly compresses published information, allowing months or years of work to be summarised in a few pages, but it also glosses over the hardships and dead-ends of real-life research and development, which are all useful lessons. Therefore, I will spend the next few pages recounting the timeline of my PhD from my own R&D perspective.

Oct 2019

MegaFLIM. The original target set out for my PhD was to use machine learning (ML) to super-resolve fluorescence lifetime images using a fusion of low-resolution (LR) FLIM and high-resolution (HR) fluorescence intensity. This was part of a brand new initiative between my research group (the Extreme Light group) and Laura Machesky's group in the Beatson Insitute for cancer research, which sought to build the first real-time megapixel FLIM detector system. The initiative was hence called megaFLIM. It aimed to use physics to enable new biological discoveries. My position within the project plan was straightforward: I would have to use machine learning to transform pairs of LR FLIM (taken with a SPAD array) and HR fluorescence intensity images (taken with a CMOS) into HR FLIM images.

This was an ambitious project, as my research group had no experience with FLIM or FLIM-FRET at the time and limited experience with microscopy. Further, we also had little machine learning knowledge: only 1 postdoc and 1 PhD student had notable ML experience, and my super-resolution project was one of the first ML-focused project strands in the group. Lastly, we had no FLIM data just then; we needed to wait for our collaborators in the Beatson group to produce a dataset for ML. Furthermore, we had to learn how to process time-resolved data (extract fluorescence lifetime), and to understand FLIM to make sense of the results.

Oct 2019 - Dec 2019

Flash single-pixel LiDAR. Upon joining the group, my supervisors added me to an ongoing project to learn the ropes, which researched flash illumination single-pixel 3D imaging. My job was to create synthetic data with various properties and to design, train and evaluate neural networks to analyse the response of our imaging paradigm to these parameters. This taught me about ray tracing, ML, and developed my research skills. My main physics-related takeaway from this project was that a static background enables the inference of spatial images from purely temporal data. My key statistics/ML observation was that fully connected neural networks readily overfit on complex data distributions (like images), therefore one should be sceptical and rigorously test

the extent of the input-label distribution over which an ML model generalises. We first submitted our work to Physical Review Letters in December 2020. Since our work was more of a novel imaging scheme than new physics, we resubmitted it to Optica in March 2020, where it was published in July 2020 [3].

MegaFLIM. While I was working on this project, our collaborators at the Beatson Institute for Cancer Research gathered a set of 900 fluorescence decay samples using a confocal PMT system. As data began trickling in, I started learning about FLIM and lifetime extraction (at first, using tail-fitting via least-squares estimation).

Jan 2020 - June 2020

MegaFLIM ANN/CNN. In my initial FLIM SR attempts, I found the lifetime of a set of $\sim$ 900 $256 \times 256 \times 75$ (x,y,t) fluorescence datacubes obtained from point-scanning cancer cells (dyed with green and red fluorescent proteins mClover and mCherry, and their composite FRET molecule, Rac1-Raichu). I sparsely downsampled/decimated these FLIM estimates to get 64×64 LR FLIM data, and summed the original datacubes a long time to get 256×256 intensity data. My first attempts were to train a neural network to map the 64×64 downsampled FLIM images onto the 256×256 parents, using a fully-connected neural network. It quickly became apparent that fully-connected neural networks are not suited for super-resolution, as they utterly failed to generalise and the image dimensionality was far too large for dense layers. I rapidly switched to a convolutional architecture, inspired by SRCNN [5]. However, the CNN's output quality was still far worse than that of the input LR FLIM images. To improve upon this and get more photorealistic results, I tried implementing SRGAN from [6].

ML lifetime estimation. In parallel, we sought to improve our FLIM data, by investigating whether our ground truth lifetime images were correct. At the time, we were collaborating with Eduardo Charbon and Ming-Lo Wu from EPFL, who granted us early access to their megaX SPAD camera. We decided to use this as an opportunity to learn about lifetime fitting and the potential of ML as a lifetime estimator for noisy, experimental data. The key takeaway from this study was that neural networks can estimate parameters of interest (in particular, fluorescence lifetime) from noisy measurements in the presence of multiple covariates, with similar fidelity as curve fitting methods. In order to achieve this for 1D temporal decay measurements, I found that dataset diversity was more important than model optimisation. Though our method was simple, it was thoroughly tested; the most informative tests were included in our paper, whereas my thesis also includes some further details. Our research was interrupted by the outbreak of COVID-19 in Europe in March 2020 and the subsequent lockdown. Publication was delayed significantly as a result; we ended up publishing in Scientific Reports in December 2020 [1].

Spiking flash single-pixel LiDAR. Simultaneously, we collaborated with Gaetano Di Caterina

and Paul Kirkland from the University of Strathclyde, to examine the potential of neuromorphic processing for improving our flash-illuminated single-pixel LiDAR scheme. The idea was to test whether a spiking convolutional neural network could infer 3D images from temporal histograms. The bulk of this work was done by Paul Kirkland, I only supplied data and conferred with him about his progress. Hence I did not include this project in my thesis. It was published in [7].

July 2020 - Oct 2020

MegaFLIM GAN. I continued my work with super-resolution generative adversarial networks (GANs). Neither I nor others in the group had experience with generative modelling; nonetheless, I decided to work out how to build a conditional GAN by myself. However, at the time there were only a few conditional GAN implementations online, and their training mechanism differed significantly from regular neural networks. The simple Keras ML API which I had previously relied on had to be replaced with custom TensorFlow functions for the conditional GAN losses (including perceptual losses for the generator in the main online implementation I found) and the alternating generator and discriminator forward-backward passes. Even when coded correctly, my SRGAN was frustrating to train, since its large size meant hyperparameter search was slow, and it was prone to mode collapse and mode hopping. Simultaneously, we kept figuring out problems with the training data itself, causing us to have to restart our research. Pretraining on public datasets of natural images helped, but did not fully solve these issues. Over the course of 2020, I iteratively debugged and improved my super-resolution GAN code.[1]

Multipath flash single-pixel LiDAR. Concurrently with my generative modelling work for the megaFLIM project, we began a follow-up study to our flash-illuminated single-pixel 3D imaging work. This study focused on exploiting information from the multipath scattering of light. My task was to create a ray-tracer that captures both 1D multipath temporal histograms and direct 3D (ToF) images of a scene; to design a neural network to transform the 1D data to 3D; to investigate the impact of the number of scattering paths on this task; and finally to process experimental radar data gathered by my teammates. This project prompted me to study how ray tracers like Blender function, and to emulate this for our specific purposes. The key takeaway from this project was that multipath echoes are central for the 3D image reconstruction task; this fact seems to be underappreciated in many depth imaging and ranging applications. Our work was submitted to Physical Review Letters in November 2020, and after a series of revisions, was published in April 2021 [4].

[1] Overall, this was a valuable experience, where through trial and error I learned a lot about machine learning and generative modelling, as well as about how to conduct secondary and primary research for new, unknown tasks more effectively. My key takeaway from my GAN work regarding primary research was that advice from experts is invaluable for overcoming the initial hurdle of tasks that have steep learning curves and many rabbit holes. By expert guidance, I include personal mentoring, viewing of forums from people attempting similar work, as well as copying and understanding existing online implementations for similar tasks in detail before attempting to solve one's own application.)

Nov 2020 - March 2021

MegaFLIM VAE. My SRGAN implementation was deterministic. Although its final iterations estimated realistic-looking SR FLIM patterns, these just reflected the noisiness/'graininess' of the FLIM images caused by lifetime estimation uncertainty. In early 2021 I began experimenting with variational alternatives, letting us sample multiple possible high-resolution FLIM images from a given LR-FLIM HR-intensity input pair. This would let us quantify the pixel-wise confidence of the network in the predictions. Inspired by Francesco Tonolini's work, I implemented a conditional variational autoencoder. However, in early 2021 I realised that our initial dataset of 900 samples had approximately uniform lifetime distributions beyond estimation uncertainty. The SNR of our $\sim$ 200 FRET samples was so low, that we were uncertain whether they were even FRETing; the estimation uncertainty was higher than any discernible lifetime variation. Moreover, the training set was too small to properly train a generative model. Thus our partners at Beatson would have had to acquire a new, larger FLIM dataset. This would have been very costly and time-consuming, especially if we wanted to ensure the samples had biological relevance and good SNR. The COVID lockdown also meant labs were closed. We began exploring alternatives that were not data-driven, using compressed sensing (CS) instead.[2]

Cavity FLIM, synthetic data. In parallel, I was invited to join a relatively new project, which aimed to use an optical cavity to enable FLIM. My role was to predict fluorescence lifetime from experimental data. For this, I would need to figure out the forward model and the experimental parameters in this model from raw measurements, code a simulation to make synthetic data and train a neural network to predict the lifetime of these synthetic datasets. Initial results with synthetic data were promising.

March 2021 - July 2021

Google radar imaging. My PhD research was paused (officially between March and May 2021, in actuality until July) for a work placement with Google, whose objective was to test whether Google's SOLI radar, designed for gesture detection in smartphones, could be used to infer depth images with machine learning. SOLI is a low-power FMCW radar using one transmitter and three receiver microstrip (patch) antennas. In this project, I modified Google's C++ code to allow readout of the device and to process the raw radar data. I also wrote a script to read an Intel RealSense D435 depth camera and synchronised the two sensors with UDP communication. I gathered datasets of myself walking around in front of the detectors. The SOLI data gave surprisingly poor depth inference, so I tried a series of physics-inspired and time-series-based ML approaches to improve the SOLI depth inference. I also synchronised another radar

[2]Theoretical work for this was done by Areeba Fatima, so I did not include it in the main body of my thesis. Experimental CS-based super-resolution approaches would involve making hardware changes, which limits their practical usefulness, and which would have been difficult to implement during lockdown, so we abandoned this idea after a few months.

(Position2Go) with both devices to act as a reference. While the algorithmic improvements (particularly joint analysis of time-series of multiple frames instead of processing individual frames independently), the SOLI's low emission power was a fundamental bottleneck: the higher power, more directive P2Go radar consistently gave significantly better results. This work was not part of my PhD, so it is not included in the main body of my thesis.

July 2021 - Feb 2022

Metal detection with radar. Following my work with Google, Kevin Mitchell and Khaled Kassem began working on using data fusion of FMCW radar signals with ToF data for concealed weapon detection. My role in this project was advisory; I gave them my previous radar data processing code and counselled them on how to set up the experiment and capture the necessary data. Our work was published in [8].

Cavity FLIM, experimental data. During this period, my focus was mainly on cavity FLIM. On synthetic data, lifetime prediction was accurate, but for experimental data, they were both inaccurate and imprecise. Improving my algorithms to account for this ranged from nailing down forward model parameters, through computationally aligning the data, to investigating the impact of degradations like blur, spatially and temporally varying PSF, measurement noise and background signal. During this time, we also moved to a new building, for which the experiment was dismantled.

From megaFLIM to SiSIFUS. Around August 2021, our megaFLIM experimentalists started collecting new fluorescence data. In the hopes that we acquire a better fluorescence dataset than the previous one, I returned to working on my CNN/cGAN/cVAE models between $\sim$ September and December. However, by December I realised that it was unrealistic to expect a sufficiently large and diverse dataset for conventional super-resolution. Therefore, I proposed a new approach for megaFLIM. The proposal was fueled by the observation that, despite there being no analytical formula linking fluorescence intensity and lifetime in the general case, our new samples often showed a correlation between fluorescence intensity and lifetime. My proposal was thus to add a prior to Areeba Fatima's super-resolution inverse retrieval scheme, to explicitly encode correlation statistics between the LR FLIM and HR intensity image. This method formed single sample image fusion upsampling (SiSIFUS).

Feb 2022 - Dec 2022

Cavity FLIM. Having gotten cavity FLIM to work on experimental data as well as synthetic data, I set out to examine what novelty and advantages the optical cavity offers to the signal processing pipeline. I performed these tests both analytically and in simulation. The first aspect I considered was whether the SNR of the measurement is improved by the optical cavity. It is not, since

(assuming a near-lossless cavity) the same amount of photons are captured either way, they are merely redistributed differently on the iCCD. I showed this by designing a Bayesian uncertainty estimation framework which unknowingly mimicked Fisher information evaluation. Secondly, I tested how reasonable is it to use the cavity FLIM method to estimate the lifetimes and component fractions of multi-exponential decays. I found that this task was possible in theory, but the prediction uncertainty was high unless one knew *a priori* the lifetime of the various components. Since we do not have experimental data to back up this finding, this was not included in my thesis. Thirdly, I examined whether the CMOS was necessary for lifetime prediction, or if we had found a method of widefield lifetime estimation from a single iCCD image. As with the multi-exponentials, we found that this is doable, but it increases prediction error increase drastically. Further, without the CMOS our inverse retrieval scheme failed to converge.

SiSIFUS. In parallel, I worked on capturing statistical correlations within individual pairs of LR FLIM and HR intensity images for SiSIFUS. I developed approaches for capturing both global and local correlations. This work was focused heavily on image compression via feature space representations, and led to secondary research on perceptual similarity.[3]

Jan 2023 - Feb 2023

Cavity FLIM. During this period, we were answering reviews for Cavity FLIM, which sought to examine certain image artefacts in our images. The main takeaway from this round of review was that a small angular tilt in the replicas, as well as pixel misalignments between the iCCD and CMOS, can create vertical replicas. Our work was published in the Proceedings of the National Academy of Sciences (PNAS).

Concurrently, I wrote this thesis.

[3] We are considering patenting this work, hence it has not been published.

Chapter 1.

Introduction

In recent years, artificial intelligence and its subfields, machine learning and deep learning, have emerged as the technologies of the future. With an ability to learn directly from data, these algorithms have great promise for automating complex tasks that are not easy to program analytically. From medical diagnostics and chatbots to factory production lines and self-driving vehicles, AI is changing the world.

Imaging is no exception. Whether it comes to generating digital art, identifying faces in smartphone camera feeds, or reading handwritten text, deep learning algorithms are shattering the previous limits of computer performance. With the availability of enormous image datasets on the internet and the importance of photos and videos in modern life, computational imaging has emerged as one of the leading applications for machine learning.

In this thesis, I examine the use of AI for processing and analysing images, particularly time-resolved ones. Estimation of sample and scene parameters is demonstrated from noisy, superposed, low-resolution, or low-dimensional data. For each application, I examine the data and the model mapping it to parameters of interest, and report on any physics insights. In particular, this thesis demonstrates fluorescence lifetime estimation from noisy temporal data (chapter 3); finding fluorescence lifetime from overlayed spatial images (chapter 4); a novel singe-sample super-resolution scheme using multi-modal data-fusion, dubbed SiSIFUS (chapter 5); and estimating spatial scenes from their temporal traces (chapter 6).

1.1 Thesis overview

Chapter 2 provides an overview of fluorescence lifetime imaging microscopy (FLIM) and LiDAR, and compares these two time-resolved imaging modalities. Computational imaging is introduced alongside the inverse problems arising in these fields. The chapter concludes with a short review of machine learning tools for addressing these inverse problems.

Chapter 3 covers lifetime estimation (parameter estimation) from noisy fluorescence data. In this chapter, a gated SPAD is used to capture the fluorescent emission of a sample, and a feed-forward, fully connected neural network predicts the lifetime of this decay. We demonstrate that neural networks are a class of fit-free lifetime estimators which are robust to noise and variation in experimental parameters, reaching similar performance as curve-fitting at a fraction of the

computational time.

In chapter 4, a novel fluorescence lifetime setup is introduced, called cavity FLIM. Here, an optical cavity scans multiple time-gated measurements of our decay in parallel, on an overlapping field of view. The forward model and noise model are detailed; using these, Bayesian uncertainty analysis provides a lower bound on the estimation accuracy of an ideal lifetime predictor for our setup. Two lifetime predictors are provided: iterative inverse retrieval via gradient descent, and a physics-inspired dilated convolutional neural network. The strengths and weaknesses of this FLIM scheme are then discussed.

Chapter 5 covers computational super-resolution for FLIM. First, we review classical single-image super-resolution (SISR) techniques, including interpolation, inverse retrieval, and example-based SISR. Deep learning-based methods, in particular generative SISR, are then introduced, drawing analogies between them and the classical algorithms. We look at the difference between FLIM and natural images, explaining the findings which guided us towards the algorithm presented in this chapter. This algorithm is dubbed Single Sample Image Fusion UpSampling (SiSIFUS); it uses machine learning to combine data fusion with self-similarity concepts, upsampling FLIM images from the information contained in a single sample.

Chapter 6 covers the most ill-posed task in this thesis: depth (3D) imaging from purely temporal (1D) measurements. We describe the experimental difficulties associated with this task, then consider unconventional sources of information that can help constrain our inverse problem. A set of experiments are performed to analyse the impact of several experimental parameters on the reconstruction quality of this scheme. In particular, we demonstrate that a fixed background allows one to reconstruct 3D images from temporal data, and in a follow-up study, we examine the temporal information content of multipath reflections of light within the scene. Combining these concepts, we show that neural networks can exploit a fixed background scene and the multipath echoes within it, to transform simple hardware such as a radar transceiver or microphone&speaker into a 3D camera. This work was done in collaboration with various members of the Extreme Light group; Alex Turpin spearheaded the research.

Chapter 2.

Background and related work

2.1 Fluorescence lifetime imaging

Fluorescence lifetime imaging (FLIM) is a time-resolved imaging modality. Unlike conventional time-integrated (intensity) imaging, where we simply accumulate photons over an acquisition period, FLIM demands temporal information about our sample. In return, it allows us to probe specific properties of a fluorophore or its environment, such as pH, oxygen levels, and protein interactions.

In particular, protein interactions, visible through FLIM, allow us to study a wide variety of biological properties. These include tension in the cell membrane [9], tension at focal adhesions [10], dynamics of the actin cytoskeleton [11], absolute intracellular calcium concentration [12], neural signaling [13], and much more.

Fluorescence lifetime

Fluorescence refers to the delayed emission of light after a molecule is excited. First, a fluorescent molecule (fluorophore) is illuminated with light of a specific wavelength, called the excitation wavelength. A photon of this wavelength λ has a set amount of energy, $E = hc/\lambda = hf$, where h is Planck's constant, c is the speed of light and f is the frequency of the light. This photon carries just enough energy to excite an electron within the fluorophore from its ground state into an excited state in a higher energy band. Next, this electron rapidly and non-radiatively decays to the lowest energy state of the high energy band ($\sim 10^{-15}$s), where it is more stable. Finally, the electron spontaneously decays back to the ground state, simultaneously emitting a photon whose energy matches the band gap traversed by the electron. The emission of such photons is called fluorescence, and their wavelength is called the emission wavelength.

Electron decay, and thereby fluorescence, occurs at a specific rate constant k. The decay rate constant describes the probability distribution of the decay of an individual molecule over time. It can be measured by exciting the same molecule over and over again and observing the distribution of the emitted light, or by exciting a larger sample of identical molecules and collecting light from many de-excitation events in parallel. Analogously to radioactive decay, the inverse of the fluorescence decay rate is the fluorescence lifetime τ,

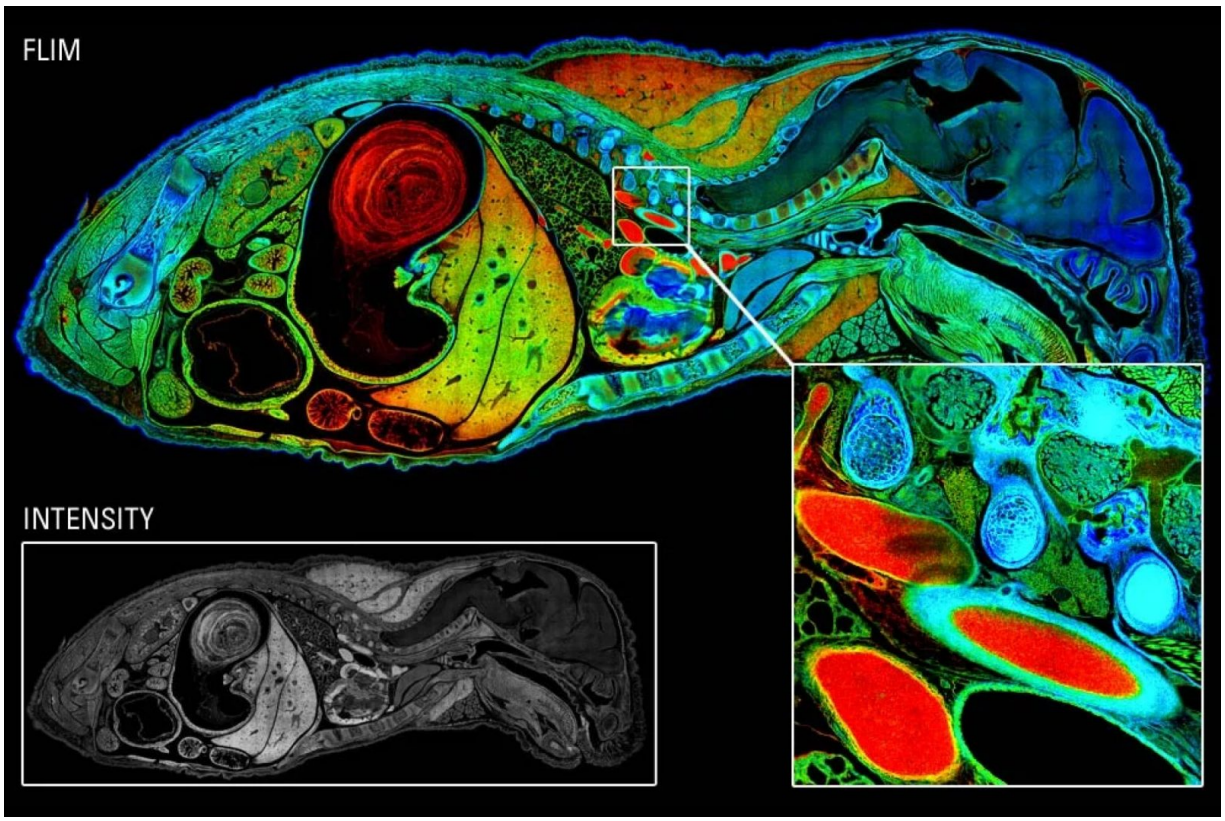

Figure 2.1: Mosaic image of a mouse embryo, comparing fluorescence intensity and lifetime. The FLIM image contains information that the intensity image fails to convey. Image from Martin Stöckl.

$1-1/e$ of a population of initially excited molecules will decay within one lifetime. With this formalism, we express the fluorescent decay rate constant k(t) as a mono-exponential function of time t and lifetime τ:

$$k(t) = \begin{cases} k_0 \exp\left(\frac{t_0 - t}{\tau}\right), & t \geq t_0 \\ 0, & t < t_0 \end{cases} \tag{2.1}$$

where k_0 is the initial decay rate and t_0 is the arrival time of the laser pulse on the sample. The expected fluorescent signal h(t) in a time bin δt around time t is then:

$$h(t) = \int_{t-\delta t/2}^{t+\delta t/2} n \cdot k(t) dt \overset{\text{small } \delta t}{\approx} n \cdot k(t) \cdot \delta t \tag{2.2}$$

where n is the number of excited fluorophores. See Fig. 2.2(a) for an overview of fluorescence and Fig. 2.2(b) for a schematic of fluorescence and decay lifetime.

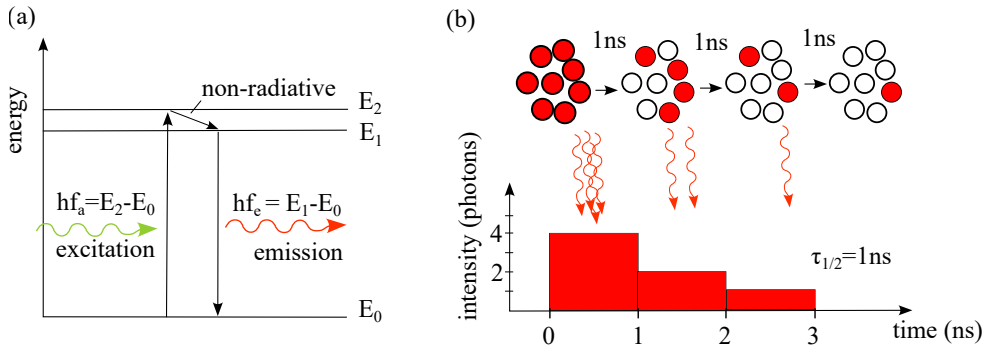

Figure 2.2: **(a)** Schematic of a simple 3-level fluorescent decay system. A photon, whose frequency matches the absorption bandgap of the molecule in question, excites an electron in a molecule from its ground state into an excited state in a higher energy band. The electron rapidly decays to the lowest state in its band, and then decays back to the ground state by emitting a photon. **(b)** Illustration of the lifetime of fluorescence decay. A pulsed laser excites a population of 8 fluorescent molecules. The molecules decay with a half-life of 1ns; so, the most likely scenario is that in the first 1ns after excitation, half of the excited molecules (4) decay and emit photons. In the following 1ns, half of the remaining 4 de-excite, emitting 2 photons; in the next 1ns, we expect 1 decay. Our camera measures a decay signal similar to this one but with added noise.

Optics are used to image this signal onto a detector. The signal is broadened by the laser pulse width (which may be on the order of 50-100ps for a high-end pulsed laser diode [14, 15], or on the order of femtoseconds for ultrafast sources such as Ti:Sapphire lasers) as well as scattering in the emission path (which is generally negligible compared to other pulse broadening effects). The detector also has some variance in transit time (the time taken for propagating charge in the detector), which further broadens the pulse. Lastly, there is bound to be some timing jitter: uncertainty in the measurement of the charge spike created by photons.

Overall, sources of timing uncertainty are combined into the instrument response function (IRF): $IRF = \sqrt{\sigma^2_{pulse} + \sigma^2_{transit} + \sigma^2_{jitter}}$. IRF is the net mapping of an impulse detection event, namely photon arrival time, to the signal measured by the instrument. The IRF of detectors varies significantly.

The measured signal s(t) is a convolution of the emitted signal h(t) - Eq. 2.2 - and this net IRF, along with any noise source n our system:[1]

$$A(t) = (h * IRF)(t) + n(t) \tag{2.3}$$

[1]This is generally taken for granted, but when considering the definition of convolution, it can be counter-intuitive. Appendix Sec. 8.0.1 explains and graphically shows why the convolution definition is correct.

FLIM-FRET

Fluorescence intensity is usually an indicator of fluorophore concentration in a sample, allowing us to see otherwise invisible structures through fluorophores bound to these structures. Fluorescence lifetime measurements are instead typically used to indicate changes in the environment of the molecule. This is because the environment of the fluorophore can impact its energy band system. For example, if a second molecule gets near the fluorophore, the interactions between their respective energy bands can result in Förster Resonance Energy Transfer (FRET).

In this process, the excited fluorophore (donor) transfers energy to the second molecule (acceptor) via a set of mechanisms. For relatively small molecular separations ($\lessapprox 100nm$), non-radiative dipole-dipole interactions mediated by virtual photons in the dominant mechanism; over larger distances, classical radiative interactions take over, with the donor emitting a real photon which is then absorbed by the acceptor [16]. The effect is strongest in the non-radiative domain. Here, the rate of energy transfer k_{ET} is strongly dependent on the distance r between the 2 molecules:

$$k_{ET} = \frac{1}{\tau_d}\left(\frac{R_0}{r}\right)^6. \tag{2.4}$$

where τ_d is the donor's lifetime in the absence of the acceptor. R_0 is the Forster distance of the 2 particles, namely the distance at which FRET efficiency, E, is 50%.[2] Energy transfer efficiency refers to the probability that an excited donor decays via energy transfer to the accepter as opposed to some other relaxation mechanism:

$$E = \frac{k_{ET}}{k_f + k_{ET} + \Sigma k_i} \tag{2.5}$$

where k_f represents the donor's fluorescence and k_i refers to all other de-excitation pathways. The FRET mechanism is outlined in the inset of Fig. 2.3. The donor molecule is excited with light of wavelength $h\nu_A$ from the fundamental electronic state S_0 to some vibrational state in the first electronic state S_1. Absorption is very fast, happening on the order of 10^{-15}s; then it rapidly relaxes to the lowest vibrational level of this state via non-radiative internal conversion, taking $10^{-13} - 10^{-11}$s [17].

From the lowest state of S_1, the electron can de-excite through several pathways, including internal conversion, fluorescence, FRET, and in some cases phosphorescence. During the fluo-

[2] The $1/r^6$ energy transfer rate originates from the square of the potential energy between 2 dipoles, which scales as $1/r^3$ where r is the dipole-dipole separation.

rescence decay path, the electron de-excites by emitting light at a wavelength $h\nu_F$. However, the rate of light emission is greater than normal as some electrons couple to the acceptor molecule via FRET. The acceptor molecule is now excited, and decays by emitting fluorescence at $h\nu_{ACC}$; however, in the works studied in this thesis, we are not interested in the acceptor's fluorescence; we filter this light out. Instead, we are interested in the increase in the *rate* of donor light emission compared to the normal rate k_f, caused by the FRET rate k_τ, as well as other interval conversion processes Σk_i that may be present, all of which quench the donor's fluorescence lifetime. In short, the donor's fluorescence lifetime decreases as the rate of FRET increases, which marks that the donor molecule is near the acceptor.

We show an example of a FLIM-FRET probe in Fig. 2.3. This probe is Rac1-Raichu, a FRET sensor composed of a green fluorescent protein donor, mClover (or clover, for short) donor, and a red fluorescent protein acceptor, mCherry. They are bound to a PAK molecule and a Rac1 GTPase, forming a complex molecule known as Rac1-Raichu. The sample is illuminated at mClover's excitation wavelength of 488nm. In Rac1-Raichu's "open" state, shown in the left-hand side of Fig. 2.3(b), our 488nm excitation light excites mClover, and we detect its 510nm emission with some high lifetime around 2.7ns.

On the other hand, when Rac1 is activated by GEF, the molecule curls up into the "closed" state, bringing mClover and mCherry into close proximity and allowing mClover's excited electrons to de-excite via FRET to the mCherry molecule. This means that the rate of electron de-excitation increases, causing an equivalent decrease in the donor's fluorescence lifetime. Thus, FRET can be considered a lifetime quenching mechanism, whose effect increases with k_{ET}. FLIM-FRET acts as a sort of ruler, as the donor's lifetime reflects k_{ET}, and via Eq. 2.4 also the distance r between the mClover and mCherry molecules with great precision, informing us whether the Rac1-Raichu molecule is open or closed.

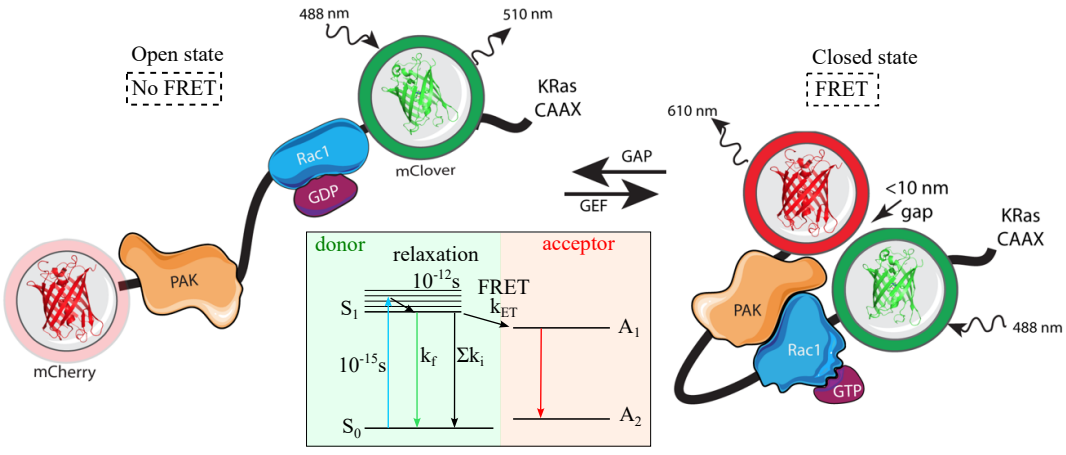

Figure 2.3: Schematic of a Rac1-Raichu molecule (image from Dr. Jamie Whitelaw), with an inset showing the FRET mechanism.

Time domain (pulsed).

There are several different imaging techniques used to make a FLIM measurement. Most techniques can be classified based on the excitation laser, illumination, imaging, and temporal sampling mechanism.

The most common form of FLIM, time-domain FLIM, uses pulsed lasers, as illustrated in Fig. 2.2(b). A pulsed laser excites the sample, creating fluorescent emission with a well-defined start point in time. Fig. 2.4(a) shows a pulsed measurement. The decay probability follows a mono-exponential probability density function, shown in red. If the molecule is excited by the light pulse, it decays according to this probability density, thus photon emission is likeliest to occur just after the excitation pulse, and less and less likely as time passes. The lifetime refers to the amount of time it takes for the molecule to decay with a probability of $1/e$.

TCSPC. Once we get the sample to fluoresce, the emitted light must be measured by some time-resolved detector so we can estimate the sample's lifetime. There are two main branches of detection. In time-correlated single photon counting (TCSPC), the arrival time of individual photons is tagged, as shown in Fig 2.4(b). A single-photon detector such as a photomultiplier tube (PMT) [18] or SPAD [19, 20] detects the signal from a single photon, and an electronic clock logs its arrival time. The arrivals times of many photons are binned into a histogram, which is distributed according to Eq. 2.3.

Time-gated sampling. Time-gated measurements use some time-dependent mechanism to collect only a portion of the signal in a given scan position. The size of this portion depends on the lifetime. Gating can be done in hardware, for example with an intensified Charge-Coupled Device (iCCD) whose gain is varied over time. Time-gated sampling applies to both pulsed-laser [24] and AMCW-laser illumination [25]. Time-gating is also possible in read-out, as in a time-gated single-photon avalanche diode (SPAD) [26, 27, 1]. Fig. 2.4(c) shows a form of time-gated FLIM that relies on 2 fixed time gates (typically used with RLD 2.1.7), but one can also scans a time-gate over the temporal decay signal in discrete steps, sampling the decay at each gate position - as in Fig. 2.4(d).

Frequency domain (AMCW).

Alternatively, an amplitude-modulated continuous-wave (AMCW) laser can also be used for so-called frequency domain (FD) FLIM. The optical-frequency laser is intensity-modulated with a much lower frequency envelope such as a sine wave, shown in Fig. 2.4(d). The decay signal is the convolution of this modulation signal and the mono-exponential decay function of the

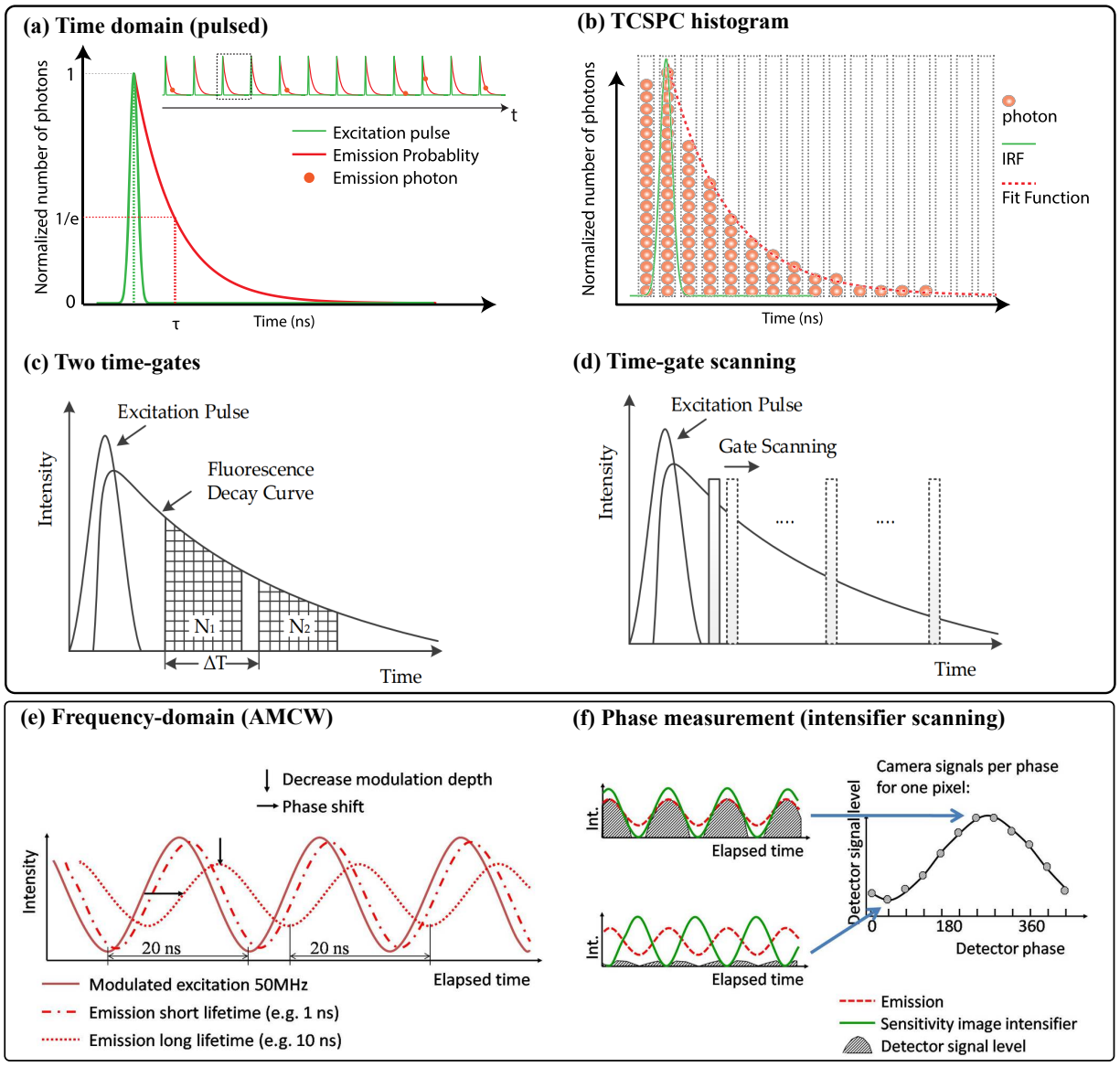

Figure 2.4: **(a)** Shows a pulsed FLIM measurement of a single molecule. We zoom in on a single excitation pulse and the subsequent decay. **(b)** shows a TCSPC measurement. A set of emission photons are observed, and their arrival time is logged into time bins. From this, a histogram of arrival times is generated and is fitted to find the lifetime. (a-b) Are adapted from [21]. **(c)** Schematic of two time-gate FLIM. The time-gates integrate photons that arrive within the given intervals. The ratio of these two integrals is used to find fluorescence lifetime. **(d)** The time-gate scanning approach integrates light over many time-gate positions. (c-d) Are adapted [22]. **(e)** Frequency-domain FLIM. An AMCW excitation envelope (a sine function with frequency 50×10^6Hz) is plotted as a continuous line. The carrier frequency is the frequency of optical light, which is much higher$\sim\ 10^{14} - 10^{15} Hz$. Emission light is shown for a short and long lifetime. **(f)** Phase shift measurement used in FD-FLIM. (e-f) adapted from [23].

fluorophore. This convolution does two things to the modulation signal. Its modulation depth (the amplitude of the oscillation from the mean) is decreased and the signal is phase-shifted. The longer the lifetime, the greater the phase shift and also the greater the decrease in modulation depth. Due to its uniform power distribution, CW illumination can deliver higher illumination doses without damaging the sample, thus increasing the total volume of detected light; however, the sampling scheme of frequency domain FLIM is less self-evident than that of pulsed-laser FLIM.

Phase-measurement A common way to find the phase and modulation depth of the signal uses an intensified charge-coupled device ((iCCD), whose intensity gain (sensitivity) is modulated to match the frequency matching of the excitation beam. For more details on iCCDs, refer to Chapter 4 Sec 4.2.2. In short, the iCCD uses a photocathode to convert incoming photons into electrons, and a multichannel plate creates an electron cloud from this photoelectron. A phosphor screen converts the electron cloud into light, which is imaged by a CCD. Fig. 2.4(e) depicts the phase measurement process, where we scan the iCCD gain signal by changing its phase. The iCCD signal is the dot product between the emission intensity and the intensifier.

When the iCCD sensitivity and the emission are in phase, we get a high signal. When the iCCD gain is out-of-phase with the emission, the signal is low. Therefore, the detector signal is a waveform whose phase matches the phase of the emission. Its modulation depth also follows the modulation depth of the emission.[3] The relationship between phase/modulation depth and lifetime are non-trivial, and are derived in the section on phasor estimators, Sec. 2.1.7.

Microscopy techniques.

The excitation and detection mechanisms are not unique to FLIM or even microscopy, and largely overlap with other time-resolved imaging fields, as we will explore later for the case of LiDAR. However, there are some microscopy-specific considerations as well.

Multiphoton excitation. In order to increase the contrast of our FLIM measurements, we often want to localise excitation to some point or plane of interest. Multiphoton excitation schemes rely on illuminating the sample with photons of a lower frequency/energy than what is required to excite electrons in the sample. The most common schemes, two- and three-photon illumination use light whose frequency is half and a third that of the sample's absorption frequency, respectively. This means a single photon does not carry enough energy to excite fluorophore and

[3]For instance, if the emission has 0 modulation depth (i.e. it is a flat, DC signal), then the iCCD gives a flat signal too, as the dot product of a sine wave with a flat signal is constant, no matter their phase relationship. Vice versa, if the modulation depth of the emission is large, the iCCD gives a large modulation depth too. It is worth noting, that we do not need to scan a very large number of points. Minimally, one must measure 3 unique iCCD phases, but 12 points are typically used for improved accuracy [28].

initiate fluorescence; instead, two or three photons are required to arrive at the same molecule in rapid succession to allow an electron to traverse the band gap. The probability density function of two or three photons arriving quasi-simultaneously scales as the square or cube of the 4D (spatial and temporal) intensity distribution of the beam. Hence, the effective range of excitation becomes more and more localised to the region of peak beam power.

Widefield illumination. The simplest imaging modality uses widefield illumination. We shine light through the sample transversely to our object plane, illuminating the entire field of view. However, the issue in widefield illumination is out-of-focus excitation. We can try to limit out-of-focus light by focusing the beam such that the object plane of our detector coincides with the focal plane of our illumination; epifluorescence setups, where excitation and emission light travel through the same arm, aid this. Further, a high numerical aperture objective can be used to minimise the depth of field and depth of focus, allowing us to focus on a narrower plane.

Light sheet illumination. To further reduce out-of-focus excitation, various light sheet setups exist [29]. A 2D light sheet aims to create a planar beam entering the object plane from the side, such that the planes above and below the object plane are not excited and thus out-of-focus fluorescence is not detected. A 1D light sheet creates a pencil beam to illuminate a single line in the object plane, likewise from the side, which can then be scanned through the sample.

Surface illumination. There are also specialist applications of decreasing out-of-focus excitation, particularly near surfaces. Total internal reflection fluorescence (TIRF) microscopy relies on imaging a sample near the refractive index boundary between the sample and the microscope-slide [30]. Light is sent in at a high angle through the glass slide (high refractive index) towards the sample (lower refractive index). The beam undergoes total internal reflection, never penetrating the sample: only a thin ($\sim$ 100nm) layer of light enters the sample, called the evanescent wave. This allows us to image surface effects like focal adhesions in high contrast. A related branch of FLIM exploits plasmonics, enhancing the detected fluorescent signal in the vicinity of specific surface via plasmonic coupling between the surface and the sample [31].

Confocal imaging. Point scanning illumination can be achieved by focusing the illumination beam down to a single point. This allows for elegant optical setups, including the gold-standard confocal microscopy setup, where either the detection, or both the illumination and detection beam are passed through a pinhole to improve the beam's focus. The emission light beam is focused down using optics, but out-of-focus object planes are imaged onto to slightly different positions around the beam waist, causing an enlarged beam waist. By placing a pinhole at the beam waist, we can mechanically filter out light from out-of-focus planes.

STED. Point scanning also permits an illumination super-resolution scheme known as stimulated emission depletion (STED) microscopy. In STED, the diffraction limit of a single spot can

be overcome, by exciting our spot with a diffraction-limited spot, causing pumping and spontaneous emission, meanwhile exciting a doughnut-shaped region around our spot with a red-shifted beam, causing population inversion and stimulated emission of lower energy photons, depleting fluorophores in the doughnut and leaving spontaneous emission only in the centre. STED can be combined with confocal imaging to achieve very high FLIM image resolutions, however, it should be noted that STED is not typically used with FLIM. This is because STED uses high light dosages to achieve population inversion around the central spot, which tends to cause photobleaching or phototoxicity in the sensitive fluorophores used for FLIM [32].

PSF engineering. Lastly, there are several illumination schemes based on point-spread-function (PSF) engineering, including double helix PSFs for improved 3D localisation [33], 3D PSFs for compressive sensing [34]. These schemes can extract additional information about the scene compared to normal PSFs by encoding some spatial property about the scene in the PSF; e.g. a double helix rotates along the optical axis, and it produces an image of two points, which are at some angle to one another. This angle matches the state of the PSF when it excites fluorescence, thus the angle between the points encodes the location of a fluorescent source along the optical axis, i.e. depth.

Lifetime fitting

There are several methods of estimating fluorescence lifetime from a decay measurement. They can be broadly classified as either fitting approaches, which minimise some cost function between the measurement and a decay model with some variable parameters, or fit-free approaches. The cost function f is the measurement of the disagreement between and the fit. The most common curve-fitting approaches include least squares fitting (LSF) and maximum likelihood estimation (MLE). There exist other methods too, covered in more detail in [35].

LSF. The objective of least squares fitting, as the name implies, is to minimise the sum of squared errors between our time-series measurement y_t and the fitted curve $\hat{y}_t$. We must therefore minimise f:

$$f = \sum_t (y_t - \hat{y}_t) \tag{2.6}$$

LSF assumes we have 0 mean noise on the data; if this is true, it is an unbiased estimator [36]. It also assumes the noise distribution to be identically distributed along the time series, since it weights all elements of the measurement equally.

Weighted LSF. A more sensible approach is to weight errors along the time series with the expected variance from the various elements of our measurement. If, for example, we expect

low variance at points near the end of the decay, then we can be confident that the fitted curve should follow these points strictly. If we expect high variance in the measurement at the peak of the decay, then we should allow the fit to deviate somewhat from these peak points. In other words, we weigh the squared error with the inverse of the expected variance of the measurement at each time point.

The cost function is thus the chi-squared distribution:

$$f = \sum_t \frac{(y_t - \hat{y}_t)}{\sigma_t^2} = \sum_t \frac{(y_t - \hat{y}_t)}{y_t} \tag{2.7}$$

The second equality in Eq. 2.7 arises because we do not know *a priori* the variance of the signal. It is therefore estimated as following a Poissonian distribution, as this is the distribution that quantised instances of a continuous variable follow in nature. A Poissonian's variance equals its expectation, and our best estimate of this expectation is the measured value in the given time bin. We note that weighted LSF is the most common form of lifetime fitting [35].

Laguerre expansion. The Laguerre expansion is a trick for turning the non-linear lifetime deconvolution problem into a linear fitting problem, speeding up calculations with respect to lifetime fitting [35]. We do this by estimating the sample's fluorescence decay h(t) with a set of discrete-time Laguerre basis functions. We then perform least squares estimation between our measurement and the Laguerre-basis measurement; however, this is a linear LSE, and can be approximated with matrix multiplication in a single step [37].

MLE. Maximum likelihood methods aim to maximise the probability of observing the measurement given some fitted curve, ensuring an unbiased estimate for a given noise model. Therefore, we aim to maximise the likelihood L:

$$L = \prod_t p(y|\hat{y})$$

Assuming that noise arises from the Poissonian distribution of photon counts, we can evaluate this probability distribution, and then take its logarithm to find the log-likelihood. This trick is useful since it turns products into sums, which are easier to deal with computationally. Log-likelihood and likelihood are maximised for the same set of parameters as the logarithm of a probability (a number between 0 and 1) strictly monotonically increases with the probability.

$$p(y_t|\hat{y}_t) = \frac{\hat{y}_t^{y_t} \exp(-\hat{y}_t)}{y_t!}$$

$$\Rightarrow L = \prod_t \frac{\hat{y}_t^{y_t} \exp(-\hat{y})}{y!}$$
$$\Rightarrow L_{\log} = \sum_t \log\left(\frac{\hat{y}_t^{y_t} \exp(-\hat{y}_t)}{y_t!}\right)$$
$$\Rightarrow L_{log} = \sum_t \left[y_t \log(\hat{y}_t) - \log(y_t!) - \hat{y}_t\right]$$

We do not need to worry about $y_t!$ since it does not depend on the fitted curve; it is a constant no matter what fit we choose. We define our merit function, m, (which tells how well our data agrees with a given curve) as the fit-dependent part of the log-likelihood, and the cost function, f, (which we aim to minimise) as the negative of the merit:

$$m = \sum_t \left[y_t \log(\hat{y}_t) - \hat{y}_t\right]$$
$$\therefore f = -\sum_t \left[y_t \log(\hat{y}_t) - \hat{y}_t\right]$$

Fit-free lifetime estimation

Fitting is computationally expensive; if real-time, high frame-rate lifetime fitting is required for large images, fitting can be too slow. Instead, we can use fit-free estimation tools. Typically, we measure fixed linear combinations of our observed time series, and perform a simple calculation on this linear combination to find the sample's lifetime.

Rapid Lifetime Determination. Rapid lifetime determination is a fast, minimalist estimator of fluorescence lifetime, using as few samples as possible to deduce lifetime. To find a mono-exponential lifetime, we require at least two distinct temporal samples, since Eq. 2.1 has two unknowns, A_0 and τ. We can prove that this is the case.

Consider the simplest case of two long, rectangular gates from t_0 to t_1 for the first measurement a_1, and another rectangular gate from t_1 to ∞. Using a change of variables, $x = t_0 - t$, we can express the measurements as:[4]

$$a_1 = A_0 \int_0^{x_1} \exp\left(\frac{-x}{\tau}\right) dx$$
$$a_2 = A_0 \int_{x_1}^{\infty} \exp\left(\frac{-x}{\tau}\right) dx$$
$$\ldots$$

[4] Full derivation in Appendix Sec. 8.0.2

$$\therefore \tau = -\frac{x_1}{\log(\frac{a_1}{a_2} + 1)}$$

Note that the time gates are simply integrals of the signal, hence each gate outputs a given linear combination of the signal, weighing each term with 0 if its outside the gate and with 1 if inside, and then summing. These two linear combinations, a_1 and a_2, are fitted with the non-linear mapping derived above to find fluorescence lifetime.[5]

Rapid lifetime determination (RLD) is a minimalist form of lifetime determination, where two samples are used to estimate the lifetime of a mono-exponential decay signal. Equivalently, a bi-exponential decay has four unknowns and thus requires at least four samples, a tri-exponential at least 6, and so on [38].

Phasor. Phasor estimation is used natively for frequency domain FLIM, but it can be applied to time-domain FLIM algorithmically. Here we Fourier transform the decay measurement and compare the real and imaginary components. The shorter the lifetime, the larger the real component will be relative to the imaginary component.

Consider a fluorescent signal A(t) from time 0 to ∞. The phasor approach analyses this data by taking its Fourier transform from the peak onwards at a specific frequency, then dividing by the total intensity. Any frequency will do as long as one period covers most of the signal. Let us first take the Fourier transform:[6]

$$\begin{aligned} A(t) &= A_0 \exp(\frac{-t}{\tau}) \text{ for } 0 <\leq \infty \\ \Rightarrow \; F(A)(\omega) &= A_0 \int_0^\infty \exp(\frac{-t}{\tau}) \exp(-j\omega t) dt \\ &\dots \\ &= \underbrace{A_0 \frac{\frac{1}{\tau}}{(\frac{1}{\tau^2} + \omega^2)}}_{\text{Re}} \underbrace{-j\, A_0 \frac{\omega}{(\frac{1}{\tau^2} + \omega^2)}}_{\text{Im}} \end{aligned}$$

Therefore, after divide by the intensity, $\int_0^\infty A_0 \exp(-t/\tau) = A_0\tau$, we get the real and imaginary

[5] TCSPC is not used in RLD since it would require purposely discarding signal, which increases the uncertainty of our measurement; however, a TCSPC signal could theoretically be used for RLD as well - see Appendix Sec. 8.0.2.

[6] Full derivation in Appendix Sec. 8.0.3

components g, and s (in fact, we consider the negative of the imaginary component):

$$g = \frac{Re(F(A)(\omega))}{\int_0^\infty A(t)dt} = \frac{\frac{A_0}{\tau}}{\frac{1}{\tau^2} + \omega^2} \cdot \frac{1}{A_0\tau}$$

$$s = -\frac{Im(F(A)(\omega))}{\int_0^\infty A(t)dt} = \frac{A_0\omega}{\frac{1}{\tau^2} + \omega^2} \cdot \frac{1}{A_0\tau}$$

Let us consider the phase θ of this complex number, $g + sj$, and see that it gives fluorescence lifetime.

$$\begin{aligned} tan(\theta) &= \frac{s}{g} \\ \Rightarrow\ \theta &= \arctan\left(\frac{s}{g}\right) = \arctan(\omega\tau) \\ \therefore \tau &= \frac{\tan(\theta)}{\omega} \end{aligned} \tag{2.8}$$

We can also consider the magnitude m of this waited Fourier transform, $g + sj$, and see that it, too, gives fluorescence lifetime.[7]

$$\begin{aligned} m &= \sqrt{g^2 + s^2} \\ &\dots \\ &= \sqrt{\frac{1}{1 + \omega^2\tau^2}} \\ \therefore \tau &= \frac{\sqrt{\frac{1}{m^2} - 1}}{\omega} \end{aligned} \tag{2.9}$$

Similarly to RLD, phasor uses a linear combination of the inputs to find fluorescence lifetime. The real and imaginary components of the phasor measurement are simply intensity weighted dot products of the signal and a fixed cosine/sine signal respectively. The frequency domain FLIM technique natively outputs the phase shift θ of the signal; the modulation m is also directly measurable, as the amplitude divided by the average signal. These quantities are then related to lifetime in the same manner as their Fourier transformed time-series counterparts [39].

Centre-of-mass. The centre-of-mass method (CMM) is another fit free method. Treating fluorescent intensity as mass, we aim to find the exponential decay's centre of mass along the time

[7] For full derivation, see Appendix Sec. 8.0.3.

axis. The intuition is that longer lifetimes, which have longer tails, have a centre-of-mass farther from the excitation pulse. Loosely, CMM is a time-domain equivalent of phase-shift phasor for frequency-domain, both of them measuring a mean lag between the excitation and emission.

For a fluorescent signal A(t), for time bins from 0 to ∞, CMM is the expected time of arrival:[8]

$$CM \equiv \frac{\int_0^\infty tA(t)dt}{\int_0^\infty A(t)dt} \tag{2.10}$$

$$(\text{where } A(t) = A\exp(-t/\tau))$$

$$\dots$$

$$CM = \frac{A\tau^2}{A\tau} = \tau \tag{2.11}$$

Yet again, as in RLD and phasor estimation, CMM uses a fixed linear combination the signal (each signal element weighted with time) to estimate fluorescence lifetime. We can also correct for IRF by subtracting its centre-of-mass from the measured CM [35]:

$$\tau = CM - \frac{\int_0^\infty t \cdot IRF(t)dt}{\int_0^\infty IRF(t)dt}$$

The logic here is that the IRF increases the average arrival time of photons by some fixed amount; thus the underlying fluorescence signal's average arrival time (its lifetime) is the CM (average arrival time) of the measured signal, minus the CM of the IRF.

2.2 3D imaging with LiDAR and RADAR

A regular camera shows a 2D projection of the world. Each of our eyes, too, creates a 2D projection of the world. This 2D projection is formed, typically, by mapping the light intensity observed from the scene onto some camera pixels using a lens. More precisely, the lens images a given object plane onto a given image plane. It is appropriate to speak of imaging in angular coordinates: light from the scene which is incident onto the detector at some given azimuth and elevation will be mapped onto a given pixel in the x-y plane of the detector, such that a given range will be in focus on the camera.

However, many applications require imaging in 3D, not just 2D. Such applications include the automotive industry [40]: a driving system, akin to a human driver, needs to know the distance

[8] Full derivation shown in the Appendix Sec. 8.0.4

between the car and the scene to plan a route that navigates effectively around objects on the road, to park the car in a tight space without bumping into the surroundings, or to break when people are crossing the road in front of the car. Other applications include surveying the sea floor (e.g. for building offshore wind farms) [41], imaging human tissues in sonography [42], or scanning airspaces for aircraft [43] and missiles [44] with radio waves.

There is a wide range of 3D imaging schemes. Humans and animals primarily rely on two methods: stereo vision and contextual depth estimation. Stereo vision arises from our binocular vision: since our eyes view the world from slightly different angles, they form images of are slightly offset from one another. These two images are stitched together by our brain, which infers a rough depth estimate from this information. Stereo vision is also widely used in computer vision [45].

We also make use of contextual cues: for example, we can estimate depth in a scene on the TV, even though the TV screen is a flat plane. People who lose one eye can still effectively navigate the world.[9] Even in a depth-of-field corrected image (e.g. where objects at various focal depths are imaged separately, and the image stacked together), we can estimate the depth of various objects based on our experience (e.g. an object that occludes another is closer than the occluded object). Many computer vision schemes, particularly deep learning-based ones, can reconstruct depth from object queues [46, 47].

In this thesis, we look at a family of ranging schemes that measure the time of flight of light from the transceiver to the objects in the scene. In particular, we consider light detection and ranging (LiDAR) and radio detection and ranging (RADAR/radar). The three most popular schemes are pulsed, amplitude-modulated continuous wave (AMCW), and frequency-modulated continuous wave (FMCW) LiDAR/radar, which can all be implemented with a single-pixel arrangement.

In LiDAR, a transmitter emits a modulation pattern at some known time $t = 0$, the light reflects off an object, and the time of flight t of the modulation pattern is measured with a receiver. The object's range from the transceiver R is related to t by the speed of light $c_0 = c/n$ in the medium (of refractive index n) between the transmitter and the imaged scene:

$$R = \frac{c \cdot t}{2} \tag{2.12}$$

where the factor $1/2$ accounts for light traveling a round-trip, twice the object-transceiver dis-

[9] This analogy is commonly used to explain contextual depth estimation, however, it should be noted that even a single eye can estimate depth by the amount of blur experienced when the eye focuses on a given distance. While this can be interpreted as contextual information since we know from experience how objects should look in focus, if we scan through the focal planes that a monocular camera sees, there are also context-free mathematical descriptors showing where blur is minimised. Therefore, monocular vision is not purely contextual.

tance.

In pulsed schemes, a transmitter illuminates the scene with a pulse (a short intensity spike) and the time of flight of the returning intensity spike is measured directly. Lasers are typically used as the transmitter for LiDAR, as they have high light intensity in a narrow wavelength band, and can emit a short pulse. The narrow wavelength band means that detected light can be filtered to reduce background noise from sources such as reflected sunlight. Near-infrared wavelength bands at 905, 1300, or 1550nm are commonly used in LiDAR [48].

In CW-LiDAR/radar, we instead modulate either the amplitude or frequency of a continuous carrier wave. Analogously to radio communication, the modulation signal or envelope contains the information content of the measurement, while the underlying wave is the carrier wave used simply to enable the propagation, transmission, and reception of the encoded signal.

Any ranging system has a limit imposed on its range resolution by the transmitted wave. The **range resolution** is the system's ability to resolve the ranges of two planar scattering objects, one of which is slightly closer to the transceiver than the other. If the system's range resolution is low, there is large uncertainty in the exact distance between the two objects and the transceiver, and their range signals overlap. Therefore there is little contrast to distinguish the two objects - the marginal range signal appears to show only one object. Vice versa, if the system's range resolution is high, the marginal range signal shows two distinct peaks corresponding to the two objects.

The range resolution δR is dependent on the bandwidth of the transmitted waves B and the speed of light c_0 in the medium:

$$\delta R = \frac{c_0}{2B} \tag{2.13}$$

There are several differences between LiDAR and RADAR, based on the scattering properties of materials at optical and mmWave wavelengths, the bandwidths used by the transceiver, and optomechanical considerations for generating and directing optical and radio waves. For instance, radio waves penetrate several optically opaque materials like fog, clothes, or plastic. Conversely, LiDAR carriers are in the optical regime and thus have much higher frequencies and available bandwidths than radar, therefore the maximum resolution achievable by a LiDAR system is superior, as dictated by Eq. 2.13.

The **lateral (angular) resolution** of a system is equivalent to the resolution of a regular plane imaging system, measuring the ability to distinguish two nearby objects at the same range. Lateral resolution is set by the diffraction limit of the transceiver and is thus directly proportional to wavelength whilst being inversely proportional to the aperture size.

Emission power is another important parameter of ranging systems. Higher emission power improves the signal-to-noise ratio of our measurement, as the intensity of reflected light is increased. The noise level typically increases less than the signal. Thermal noise in the detector is more or less independent of the measured intensity. On the other hand, shot noise, arising from the quantised nature of photon measurements, has a standard deviation that scales as the square root of the intensity. Thus, the signal-to-noise ratio (the limiting factor of any measurement) is higher when the illumination is brighter. However, higher emission power is costly and can be dangerous to the objects or people in the scene, thus it is limited in practical application.

The **maximum operating range** is limited by the transmitted and received power. For this reason, long-range ranging systems such as aircraft detection systems rely on high gain (highly directive) transmission, to reduce the power loss caused by the divergence of the illumination beam. Beam steering is used for long-range applications to retain the necessary **field of view**. A flash ranging system (which flood-illuminates the full scene and images it onto a detector array) instead sacrifices maximum operating range for a higher instantaneous field of view.

The **operating frame rate** and **resolution** are inversely correlated parameters, for fixed emission power. Scanning systems rely on scanning $N \times N$ points to form an $N \times N$ image, which scales poorly with image size, hence the scan time scales accordingly when we wish to acquire a given SNR on each scanned point. A flash LiDAR system's resolution and frame rate are also inversely correlated for fixed aperture size since the resolution is limited by the active area (the area collecting signal) of the detector array pixels. If we aim to increase the number of pixels using a detector with a given aperture size and fixed emission power, we are spreading the same amount of light on more pixels, decreasing the amount of radiation incident on an individual given pixel.

Pulsed LiDAR.

The pulsed scheme is the simplest form of LiDAR as well as radio detection and ranging (RADAR), and is analogous to time-domain FLIM. A brief intensity spike, known as a pulse, is generated by a laser, hence the range distribution of objects in the scene is given directly by the distribution of photon arrival times with respect to a clock synchronised to these pulses. Pulsed LiDAR is therefore also referred to as direct time of flight (ToF), or just ToF.[10]

Single photon detectors such as single-photon avalanche diodes (SPADs)[11] or photomultiplier

[10] The same principle exists for sound waves. For example, when we clap our hands in front of a large building, we can hear the echo of the clap. If we move farther from the building, the echo returns later. Hence, the time of flight of our pulse-like clap informs us of the distance to the building.

[11] see Sec. 3.1.1 for details

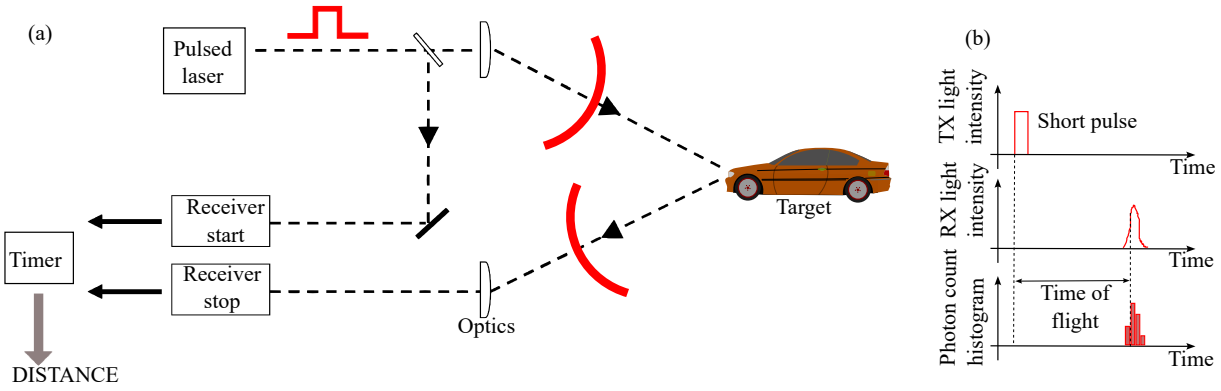

Figure 2.5: (a) Schematic of pulsed LiDAR schemes, adapted from [49]. (b) Timing diagram of a time-correlated single-photon counting (TCSPC) pulsed LiDAR measurement, adapted from [48]. The ratio of the signal measured by the gates is used to deduce the time of flight of the signal.

tubes (PMTs) are commonly used in this scheme. Pulse separation must be high enough that virtually all light reflected from a given pulse scatters or is absorbed before the next pulse.

A pulsed ranging system's resolution is physically limited by the width of the emitted pulse, and the IRF of the detector. Assuming an arbitrarily good detector, two objects are considered resolvable when their signals are separated by at least half of the pulse width time τ. This gives the following potential radar resolution R_{pot}:

$$\delta R_{pot} \geq \frac{c_0 \cdot \tau}{2} \sim \frac{c_0}{2B} \tag{2.14}$$

The laser pulse width τ is defined as the full width at half maximum of the pulse, and it depends on the pulse shape. For high range resolution, short pulses are required, where $\tau \sim B^{-1}$. A physical intuition for why a pulse's bandwidth affects the pulse length is that the mode-locked interference of sinusoidal waves within the bandwidth is what causes intensity pulsing in the first place. In reality, the detector's IRF and noise, as well as the background signal, all limit a pulsed LiDAR system's effective resolution further.

Fig. 2.5(a) shows the mechanism of pulsed LiDAR, and equivalently a pulsed radar. A pulsed laser emits a pulse and simultaneously triggers a timer to start. The pulse is relayed with optics to illuminate the scene. The target reflects the pulse, and further optics relay this signal to a receiver, triggering the timer to stop. The time between the start and stop triggers is the time of flight. Fig. 2.5(b) describes TCSPC pulsed LiDAR timing measurements. A short pulse is emitted by the transmitter (TX) and the reflected light is detected by a receiver (RX) after some time. The shape of the reflective object modulates the received signal. The detector, e.g. a TCSPC SPAD,

samples this distribution and generates a histogram of time of flight vs photon count. A pulsed radar may be operated on the same principle, sampling the signal using antennas instead.

Amplitude-modulated continuous wave (AMCW).

In AMCW LiDAR, a continuous wave's amplitude is modulated and this wave illuminates the scene. The returning signal has the same modulation, but the envelope is phase-shifted with respect to the transmitted signal due to the wave's time of flight. As such, AMCW LiDARs are a type of indirect time of flight (iToF) device.

Sine-modulated AMCW LiDAR is analogous to frequency domain FLIM. Fig. 2.6(a) shows the basic elements of an AMCW ranging setup. A source such as a continuous wave laser diode is intensity modulated by a driver, which adjusts the current through the diode according to the desired modulation amplitude (forward current is directly proportional to output power beyond the threshold current).

The modulated wave is emitted, hits the target, and is collected by a detector. It is phase shifted with respect to the transmitted wave depending on the time of flight to the object. This phase offset is found by cross-correlating (gain mixing) the signal with several local copies of the illumination waves modulation signal, each phase shifted. This is equivalent to frequency

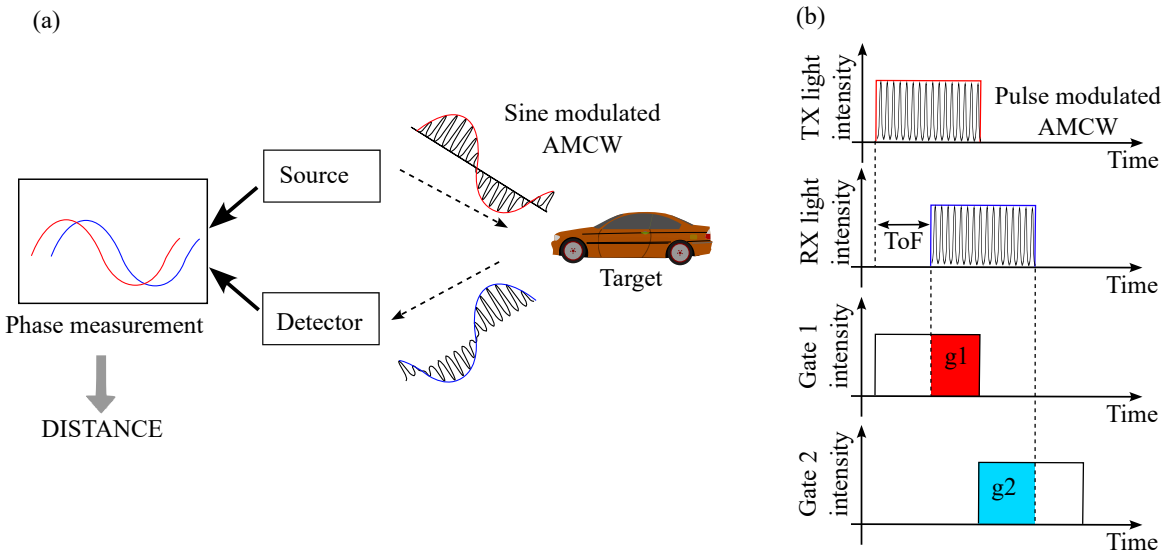

Figure 2.6: (a) Schematic of an AMCW LiDAR scheme using sine modulation. The target is illuminated with a CW carrier laser, whose amplitude modulated with a lower frequency envelope, and this signal is returned. The envelope is extracted, e.g. using a low-pass filter, and the phase shift is found, giving time of flight, and hence distance. (b) Schematic of a square pulse (or tophat) modulated AMCW measurement. Two gates, separated in time, measured the integrated return signal. The ratio between these integrals tells the time of flight.

domain FLIM. Optionally, the received signal can be mixed with a copy of the modulation signal of a slightly different frequency; this produces a lower-frequency beat signal, whose frequency is given by the difference in the frequency of these two waves. The beat signal also has a phase offset proportional to the offset of the returning signal, hence this phase offset is measured to deduce time of flight [50, 51].

Fig. 2.6(b) shows a pulse-modulated AMCW scheme, using 2 charge transfer gates that split the charge generated at the receiver. The gates are synchronised to the input signal (i.e. the modulation envelope), but one gate starts later than the other. We measure what fraction of the signal arrives within each gate, which depends on the time of flight of the signal.

Frequency-modulated continuous wave (FMCW).

FMCW ranging relies on modulating the frequency of the transmitted signal with some function and observing the difference in frequency between this and the received signal. Fig. 2.8(a) illustrates how a typical FMCW radar system generates an intermediate frequency (IF) signal

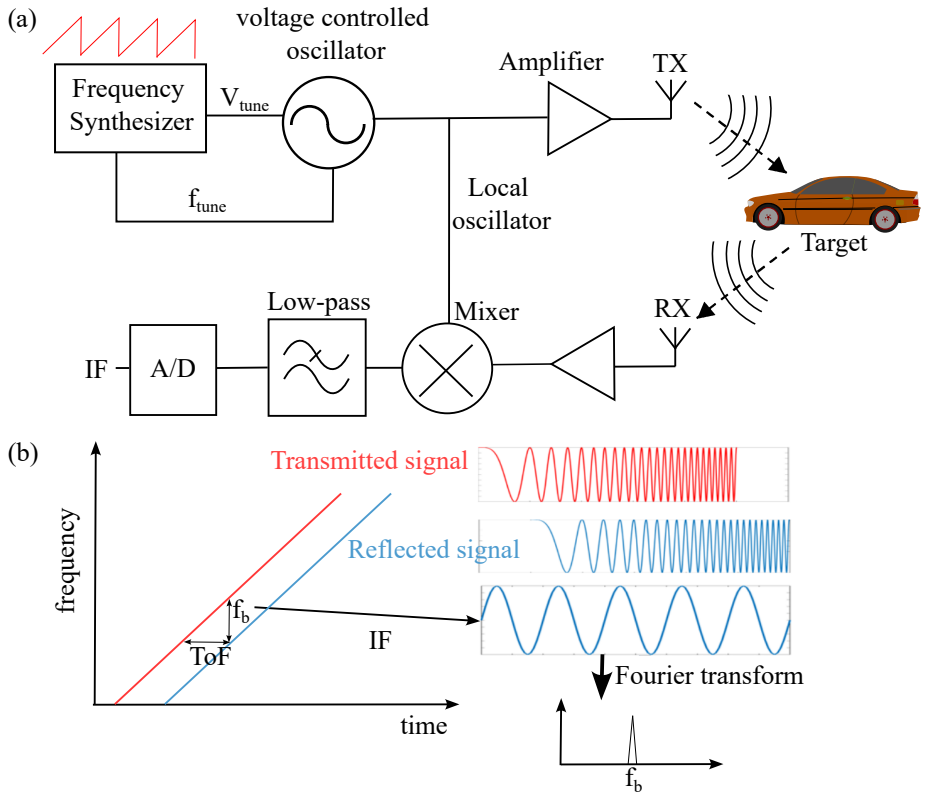

Figure 2.7: (a) Schematic of an FMCW radar system. (b) Signal processing used to extract range information. The reflected signal is shifted in time with respect to the transmitted signal, hence at any given point, there is a frequency difference f_b between them. The longer the time of flight, the larger f_b.

whose frequency matches the beat frequency (frequency difference) between the transmitted and received signals. A frequency synthesizer and voltage-controlled oscillator generate a signal whose frequency follows the desired modulation pattern, such as the shown sawtooth wave. This signal is divided between an amplifier and a local oscillator. The amplifier is wired to the transmitter antenna (TX) which illuminates the target. The receiver (RX) picks up the reflected signal, which is amplified and then mixed with the local copy of the transmitted signal. The mixer outputs a signal with frequencies matching the sum and difference between the received signal and the local copy of the transmitted signal. A low-pass filter cleans the output of the mixer, leaving only the IF signal oscillating at the beat frequency f_b, which is the difference between the received and transmitted signal frequencies.

Fig. 2.8(b) graphs the transmitted and received signals as a function of time, showing how the time of flight of the reflected wave depends on the measured beat frequency (f_b), and on the gradient of the frequency ramp. Therefore, the signal measured by an FMCW system can be Fourier transformed to find the time of flight of light between the transceiver and a reflecting object, and equivalently the object's range (Eq. 2.12).

Performance comparison

AMCW lidars can use relatively simple electronics both on the emitter and detector side (including CMOS-like detectors [52]), so they are typically cheaper than pulsed LiDAR systems [53]. However, they have poor maximum range and resolution. FMCW systems can have rather complicated readout and processing compared to pulsed and AMCW detectors, but they offer extreme range resolution. Fig. 2.8 shows a performance comparison of pulsed, AMCW, and FMCW LiDAR.

We can see that both FLIM, and time of flight ranging, rely on measuring the time of flight of photons illuminating a scene. In FLIM, the sample is at a fixed distance from the transceiver, hence ToF tells us the distribution of fluorescence emission over time. Conversely in ranging, we assume the objects reflects light the moment it is illuminated, hence ToF tells us the distance between the transceiver and the object. The measurement, in both cases, is time of flight.

Pulsed FLIM and pulsed LiDAR/RADAR work analogously, and they can both be operated via either time-gated or TCSPC detection. Frequency domain FLIM and AMCW LiDAR are also similar; in both cases, a carrier wave is intensity modulated, often with a sine wave, and the phase shift between the illumination and received waves can be used to deduce lifetime or range.[12]

[12] Up to my knowledge, FMCW FLIM is not used. This might be for a variety of reasons, for example, fluorophores have unique and non-linear absorption and emission spectra, hence the fluorescence process changes the excitation

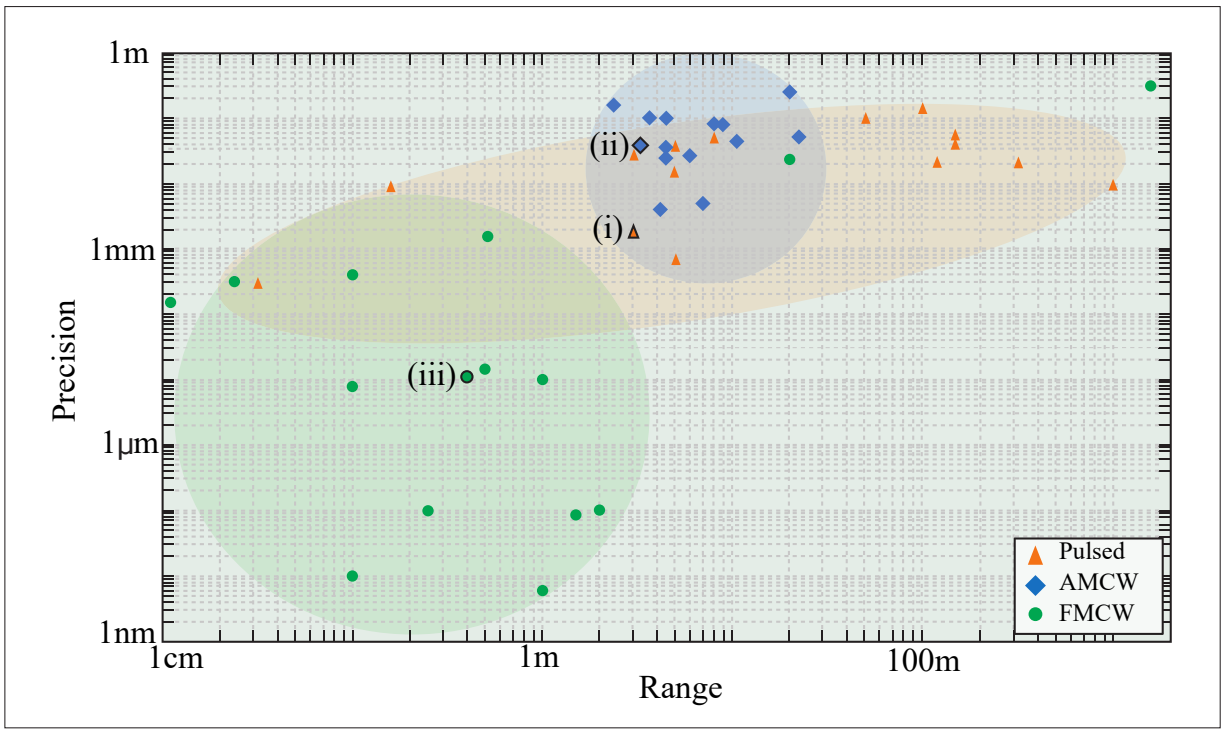

Figure 2.8: Comparison of pulsed, AMCW and FMCW lidar ranges and precisions industrial and academically published lidars since 1990, adapted from [48]. We highlight (i) a pulsed LiDAR from [54], (ii) an AMCW LiDAR from [52], and (iii) an FMCW LiDAR system from [55].

2.3 Computational imaging

Computational imaging (CI) is a field that combines the strengths of computer algorithms with traditional imaging techniques. Its main goal is to capture and process images computationally for improved resolution, lower power requirements, and better unscrambling of measured information. CI has many real-world applications, particularly in medical imaging, where algorithms such as compressive sensing are used to generate MRI and CT scan images. Other examples include image formation tools for fluorescence lifetime imaging (FLIM) and LiDAR/RADAR.

Computer vision (CV) is a closely related field that uses algorithms to process, interpret, understand, and improve the information content of images. In medical imaging, the analysis of MRI and CT scans using CV techniques [56, 57] allows doctors to more accurately detect a wide range of conditions, from cancer to heart disease. This can lead to earlier diagnosis and more effective treatment, potentially saving lives.

spectrum with respect to the emission spectrum. Therefore the beat frequency between the excitation and emission signal is not only dependent on time of flight (lifetime), but also on the fluorophore spectra. FMCW LiDAR/radar assume that the reflected light has the same spectrum as the illumination light, hence this issue is resolved. FMCW FLIM might be an interesting area of research.

CV is also widely used in other areas such as manufacturing, where it is employed to inspect products for defects [58, 59]. Facial recognition technology, which is commonly used in smartphones, is another example of CV in action. In LiDAR, CI forms the pipeline for creating a 3D image, while CV tools are used to segment objects in a scene and classify various entities in the images. [13]

In the field of microscopy, researchers are using computational imaging to create images of biological samples with unprecedented detail, using techniques such as STORM/PALM [60, 61]. To make sense of and enhance these microscopy images, there are many simple and robust applications, such as contrast enhancement toolkits, denoising methods, and edge detection. More complex but still robust CV applications include image segmentation, popularised in recent years via the open-source deep-learning-based package, Cellpose [62].

Despite the many successes of CI and CV, there are still many challenges to overcome, particularly when the processing is complex and data-driven, like in deep learning algorithms. The biggest challenges include generalizing algorithm behaviour from the training set to a wide range of unseen, real-world imaging conditions. This often involves dealing with large amounts of data and creating clever processing algorithms. The algorithms used in computational imaging can also be computationally intensive, resulting in hardware and energy consumption bottlenecks.

Inverse problems

Among computational image processing tasks, many of the most challenging yet most useful ones are so-called inverse problems.

Inverse problems. In terms of causality, an inverse problem is when we are trying to find out what caused an observable effect, as opposed to a forward model which simulates the effect that a cause will have. A classic example is trying to figure out the shape of an object based on its shadow. If we know the shape of the object, it's easy to figure out what the shadow will look like. But if we only see the shadow, trying to figure out what object caused it can be a real headache.

The best way to tackle an inverse problem is to have a good understanding of the underlying physics or process that's causing the effect. With that knowledge, we can use mathematical techniques like iterative optimization or Bayesian inference to search for the most likely cause. And, even if we can't get a perfect answer, we can still come up with approximate solutions that are close enough for most practical purposes.

[13] It's worth noting that the distinction between CI and CV can be blurry. For instance, both may be involved in generating a fluorescence lifetime map from TCSPC data. However, CI generally refers to image capture and processing tools, while CV refers to the algorithms used to extract information from formed images.

In the real world, inverse problems are common. A CT scanner trying to image the inside of a patient's body is a great example. The scanner sends out X-rays and other forms of radiation, but it only receives the shadows that are cast by the internal organs. It is an inverse problem to try and figure out what the internal organs look like based on the shadows. There are inverse problems in signal processing, weather forecasting, seismology, and many more areas. Some of the most prominent inverse problems in imaging include denoising, super-resolution, and compressed sensing.

We can express inverse problems as the restoration of an input (cause) x from some observation (effect) y. There is some function, the measurement operator M, that forms this observation. We try to invert this forward process to form an estimate of the input $\hat{x}$:

$$\begin{aligned} y &= Mx \\ \Rightarrow\ x &= M^{-1}y \end{aligned}$$

M is the forward model, which maps the cause to the effect. In the best-case scenario, the inverse problem is well-defined, that is, the information content of the image is not lost in the measurement process. Equivalently, this means the measurement operator is bijective - each element of the input set maps to one element of the output set, and each member of the output set has an input mapping onto it. In this case, the problem is invertible, with a unique, well-defined solution. For example, the cubic function $y = x^3$ is invertible, using the cube root.

Often, however, the forward model causes some information to be lost; such problems require us to use prior assumptions on the input x to invert the problem. For example, consider a forward model which simply squares the input $y = x^2$, and acts on the set of all real numbers. If we want to get back x from y, the sign (+/ -) is lost. To address this, we must use prior knowledge or assumptions to constrain the inverse problem. In the $y = x^2$, scenario for instance, if we know that our input set only contains positive numbers (for example, if x is light intensity measured on a pixel), the problem becomes invertible. More generally, we simplify the inverse problem using **regularisation**. A regulariser is an assumption that helps us decide which of the various possible solutions of an inverse problem is most likely.

Inverse problems addressed in this thesis. In Chapter 3, we consider the task of fitting a decay model to noisy data; this task is a combination of an ill-posed denoising task and curve fitting. In Chapter 4, we examine a task that involves reconstructing fluorescence lifetime from time-resolved images that have been overlapped onto one another. The observed image is a linear combination of the target images. In Chapter 5, we describe a computational pixel super-

resolution task, in which we aim to increase the number of pixels in an image. Our input is a large image and our operator is a sparse measurement, which keeps certain pixels and removes others; the output is therefore a linear combination of the input with binary weights.

In Chapter 6, we endeavour to reconstruct a 3D image from its time-resolved trace. As an inverse problem, the input is a depth map (x, y, z), and the measurement operator is the summation of the input along the x and y axes, which is a linear combination that decreases the dimension of the input. A common thread that runs through the tasks in Chapters 4-6 is that they are all inverse problems in which the measurement is some dimension-reducing linear combination of the input.

Compressed sensing.

There is a theoretical framework that allows us to solve such tasks analytically, known as compressed sensing (CS). CS is a field of signal processing that allows us to recover high-dimensional signals from a small number of observations. The basic idea behind compressed sensing is that many real-world signals, such as images or audio, can be represented with a lot fewer measurements than the number of pixels or samples in the signal. The key insight that enables this is that many signals are "sparse" in some basis.

The mathematical foundation of compressed sensing is based on the concept of **sparsity**. A signal is considered sparse if most of its coefficients in a certain basis are zero or close to zero. In other words, a sparse signal can be represented with only a small number of non-zero coefficients in a certain basis or frame.

To understand the working of compressed sensing, let's take an example of an image. An image can be represented as a matrix of pixels, where each pixel has a colour value. However, many images have lots of similar or redundant pixels. In other words, the image can be represented by a small number of significant measurements, which capture the essential information of the image.

In compressive sensing, we take a small number of measurements of the image, which is significantly less than the number of pixels. These measurements are often random and are taken in the form of a dot product with a certain measurement matrix. This measurement matrix and the sparsity basis should be incoherent. This helps fulfil the **uniqueness requirement**, that distinct sparse signal should be mapped to distinct measurements [63].

The recovered image is obtained by solving an optimization problem, which helps us to recover the original image from a small number of measurements. The sparsity of the signal acts

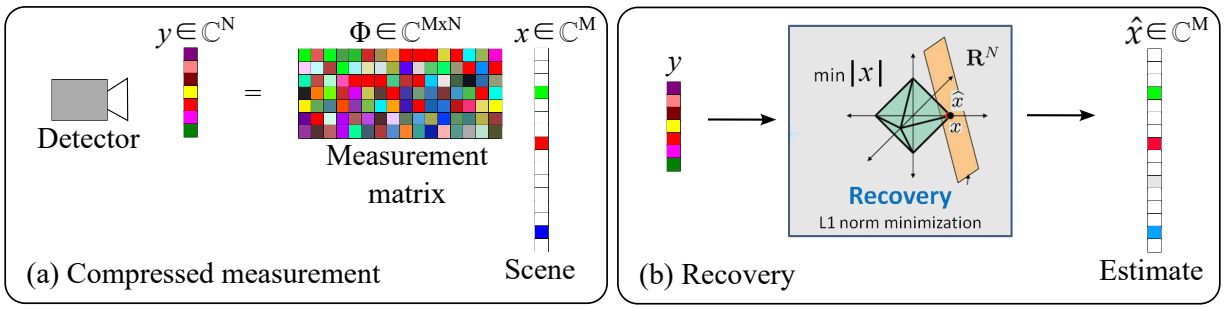

Figure 2.9: Schematic of a compressed sensing pipeline. (a) A sparse signal x is compressed by some physical process, a measurement matrix Φ. The compressed mixture is detected and sampled giving measurement y. (b) Using some algorithm, such Lagrangian/L1 norm minimisation (shown in Eq.2.15), we find an estimate of the signal, $\hat{x}$.

as a regulariser, which can be encoded by minimising the number of non-zero coefficients of the signal in the basis of interest, e.g. via L1 minimization, while still recovering the original signal. Our estimate $\hat{x}$ is then found by joint minimisation [34]:

$$\hat{x} = \underset{x}{\operatorname{argmin}} ||\Phi x - y||_2 + \lambda ||x||_1 \tag{2.15}$$

Fig. 2.9 illustrates the compressive sensing workflow. The key takeaway is that recovering a high dimensional signal from few measurements is possible with analytical tools, but two things are required. Firstly, the signal must be sparse in some basis, making it compressible and lowering its information content such that it can be encoded by a low dimensional measurement. Secondly, the measurement matrix should preserve the uniqueness of distinct (sparse) input signals. There are a series of matrices that fulfil this criterion for a given sparsity basis; see [64].

As we will see in the descriptions of our various imaging tasks in the following chapters, many computational imaging tasks do not meet this second criterion. This necessitates the use of more generic methods for solving such inverse problems. Hence we turn to machine learning.

Machine learning and deep learning

Machine learning is a field of computer science that involves the development of algorithms and statistical models that enable computers to learn from and make predictions or decisions without being explicitly programmed to do so. The ultimate goal of ML is to develop models that can generalize well to new, unseen data.

When we solve inverse problems with ML, our model ideally learns an approximate inverse

for the forward model. In reality, the model learns some mapping from a set of inputs to a set of labels; if the model manages to generalise to test data, we can be confident that it learned to approximately invert the forward model. If not, then the model has found a different mapping that happens to coincide with the inverse model on the training data; this is known as overfitting.

Machine learning encompasses many fields. These range from supervised learning [65], where a mapping is learnt on input-label pairs. This includes methods such as linear and polynomial regression (curve fitting) for mapping inputs to continuous variables. It also encompasses classification, for example via support vector machines (SVMs), decision trees, or K-nearest neighbours (KNN). Another family of ML is unsupervised learning [66] where a mapping is learnt either from the input to itself, typically used for dimensionality reduction[14] via linear algorithms such as principal component analysis (PCA) or non-linear algorithms such as uniform manifold approximation and projection (UMAP). Unsupervised learning is also used for clustering[15], via algorithms such as K-means (centroid-based clustering) or DBSCAN (density-based clustering). Reinforcement learning is a more disjoint branch of machine learning, which seeks to learn policies for agents to follow, in a way that enables them to achieve the highest reward in their environment. Machine learning has been reviewed extensively - see [67, 68].[16]

Deep learning is a subfield of machine learning and the fastest-growing one in recent years. Deep learning relies on neural networks, which are algorithms that perform computations sequentially. Common computations include linear combinations of the inputs (dense layers), convolving the inputs with some kernel (convolutional layers), and weighted multiplication of input elements with one another (attention layers). Elementwise functions, called activation functions (sigmoid, tanh, rectified linear unit, leaky rectified linear unit, etc.), add non-linearities to the system. Memory-based computations (e.g. LSTM) allow neural networks to encode serial dependence of inputs, whilst skip connections (e.g. recurrent neural networks, UNETs) allow cross-referencing of various layers, providing computational benefits. The basics of deep learning are covered in [69]; but the field is evolving rapidly, so works are usually considered outdated within a few years.

In this thesis, a set of neural networks are explored. Chapter 3 uses a small fully connected neural network for estimating the lifetime of a 1D fluorescence decay curve. Chapter 4 uses a physics-inspired, dilated convolutional neural network, essentially for unmixing superposed fluorescence and thus finding the lifetime of the sample. Chapter 5 introduces generative modelling

[14] Dimensionality reduction maps the inputs to a lower dimensional continuous variable that retains the useful information about the input

[15] Clustering maps the inputs to discrete classes based on common characteristics

[16] Whilst these tools are not covered in my thesis, as the final workflow of my projects relies on deep learning, these basic, robust tools are invaluable for analysing data; in particular, before I engage in any deep learning, dimensionality reduction techniques (PCA and UMAP) are my go-to for making sense of high dimensional data by eye.

for super-resolution, particularly generative adversarial networks (GANs) and variational autoencoders (VAEs); these approaches were tried for our task, but eventually discarded due to training set limitations; instead, we ended up steering towards explicit data fusion of two different image modalities. This was implemented via a lightweight convolutional-dense network, designed to be retrained from scratch for each test image, on the image itself. Lastly, Chapter 6 considers both a fully connected neural network and a convolutional autoencoder for reconstructing 3D scenes from purely temporal data.

Feature extraction and convolutional neural networks

Machine learning algorithms are designed to automatically look for and exploit patterns in data. In ordered data, such as a matrix of information about people (where columns represent features like height, weight and age, and rows represent the different people) dimensionality reduction tools like PCA identify covariance between the data's features.

Images work differently, showing raw intensity values across a field of view. For example, consider an algorithm for identifying emotions from photos of peoples' faces. We first identify low-level features such as edges, textures, contrast, and colour. Typically, various low-level features are spread throughout the image, so we aim to decompose an image into small patches, which we process separately. The patches combine into a feature map that shows the 'strength' or prevalence of various patterns throughout the image. Low-level feature maps join to form higher-level feature maps (i.e. abstractions) like an eye or lip, ultimately enabling the algorithm to differentiate between emotions.

Convolution is excellent for extracting image features. In convolution, a small kernel slides across the image, and we calculate the dot product of this kernel and the input at each position, giving a so-called feature map. Networks that performs convolution, aptly named convolutional neural networks, are the workhorses of image processing in ML, and are known for building hierarchical representations of image features.[17] This offers several advantages. Firstly, the kernel is small, allowing patches of the image are processed independently, capturing the local correspondence of pixels and letting us build hierarchical abstractions based on these, instead of processing the entire image in one go. Secondly, convolution slides the same kernels across the field of view, so feature extraction is cost-efficient and translation-invariant. Finally, convolution can be implemented in a computationally efficient manner using Fast Fourier Transforms.[18]

[17] To map from the feature maps of a given layer to a single feature map of the next, a convolutional layer convolves (or cross-correlates) a unique kernel across each input map, and adds the convolution outputs. A bias is also added, and an activation function is performed, as with dense layers. Different kernels then perform the same operation to get another feature map in the output layer.

[18] The convolution theorem states that the Fourier transform of the convolution of a and b equals the Fourier transform of a multiplied by the Fourier transform of b, $FT(a * b) = FT(a) \times FT(b)$.

Chapter 3.

Real-time megapixel lifetime estimation

Summary. Real-time fluorescence lifetime estimation using deconvolution via fitting, with either least-squares fitting or maximum likelihood estimation is time-consuming. It is challenging to achieve real-time processing on a similar time scale as fluorescence exposure time. Fit-free methods can be employed in real-time, however, their prediction uncertainty is higher than fitting-based methods. To address this, we developed an artificial neural network to perform rapid, accurate, and robust fluorescence lifetime fitting. The network fits SPAD-array data pixel-by-pixel, estimating the lifetime of a 0.5MP image in 2.7s, outperforming gold-standard least-squares fitting by several orders of magnitude while maintaining similar performance. Our results were published in Scientific Reports (2020) [1].

3.1 Introduction

Wide-field imaging for fast acquisition. Fluorescence lifetime imaging setups are typically either scanning or wide-field based. Point-scanning (raster scanning) systems work well with confocal and multiphoton microscopy and can use simple detector architectures as the signal is bucket collected. Hence they do not require a full camera, just a time-resolved pixel. However, traditional point-scanning systems used in FLIM can suffer from photo-bleaching, limiting the amount of optical energy we can shine on the sample. Additionally, these systems cannot provide instantaneous full field of view (FOV) information, which is important for dynamic samples or in vivo applications.

Wide-field acquisition can be up to N^2 times faster than raster scanning, where N is the number of pixel rows in a square array detector. This assumes we sample each of the N^2 points in a raster scanned image for time t, requiring a total time of N^2t, ignoring any source of delay like the time needed to move the optical path in the scanner. With wide-field, one can image all N^2 points in parallel, requiring only t time.[1] However, this also assumes that every pixel in the wide-field system has the same active area and sensitivity as the bucket detector used in point scanning - if the scanner's active area is larger (as it generally is) or its sensitivity higher, it will gather more emission light, so it will be able to reach our SNR requirements faster. Nonetheless, it is generally true that large detectors in wide-field setups, such as the SPAD array used in this

[1] This assumes that we illuminate the field of view simultaneously with the same optical power density as we would use if imaging a spot, thus imaging at the same SNR.

work, provide the fastest imaging capability for the dim samples that are often encountered in biologically relevant experiments.

SPAD array for spatiotemporal resolution. Traditionally FLIM systems have usually used microchannel plate-based optical intensifiers to record temporal resolution in wide-field systems, combined with a camera such as a Charge-Coupled Device (CCD). Alternatively, Single Photon Avalanche Diode (SPAD) arrays, made with complementary-metal-oxide semiconductor (CMOS) technology, can be used. SPAD cameras operate in either TCSPC or time-gated acquisition mode. The main advantages of SPADs over CCD/CMOS cameras are their picosecond temporal resolution and single-photon sensitivity, which make them useful for a wide range of ultrafast time-resolved imaging applications.

Recent technological advances have led to the development of SPAD arrays with a large number of "active areas" or "pixels". Such high-resolution SPAD arrays are enabled by time-resolved electronic gating, improving photon measurement probability and simplifying the electronics compared to TCSPC.

Accelerating lifetime prediction with ANN. A FLIM datacube (of dimensions $X \times Y \times T$, where T is the number of gate scans) must be processed to find the underlying fluorescence lifetime [2]. Fitting-based lifetime estimation methods, in particular LSF and MLE, rely on iterative deconvolution of a mono-exponential decay curve (or a mixture of exponential decay curves with different lifetimes) with the IRF. This process is relatively slow and scales poorly with the number of free parameters within the optimisation scheme. As such, fitting-based methods are typically reserved for offline processing [35] [3].

Fit-free methods such as RLD, phasor analysis, and central moment estimation are rapid, hardware friendly and easy to interpret since they rely on simple linear combinations of the measurements. Consequently, they are preferable over curve-fitting for real-time applications. However, this same simplicity makes them less flexible than fitting methods. They cannot account for uncertainty in parameters such as the shape of the IRF or the arrival time of the pulse on the sample; RLD cannot even directly account for the background signal.

To address this, we created a new fit-free lifetime estimator, an artificial neural network (ANN) that combines the speed of a fit-free method with the flexibility of fitting-based methods, enabling robust real-time lifetime estimation. Smith et. al. used a convolutional neural network to estimate fluorescence lifetime [70]; we use a simple ANN architecture instead, which performs linear combinations on the input, followed by element-wise non-linearities (activation

[2] See Section 2.1.6 for a review of lifetime fitting.

[3] Laguerre basis expansion is a powerful tool to fix the time consumption of fitting-based estimators, but fit-free methods are still faster. Laguerre expansion might seem complicated for most users of FLIM (even if it is mathematically robust), hindering its popularity for now.

functions), inspired by established fit-free techniques. We designed and tested the model on estimating lifetimes from experimental measurements made by our SPAD array.

SPADs

A single-photon avalanche diode (SPAD) is a photodiode (a semiconductor with a P-N junction) reverse biased above the breakdown voltage, illustrated in Fig. 3.2. A photon causes impact ionisation, separating an electron hole pair. The reverse bias attracts the electron into the n-type region, where it causes avalanche multiplication. Hence, a single photon can create a detectable current.

The avalanche builds up rapidly, so a readout circuit with low timing jitter can tag the time of flight of the photon that caused impact ionisation based on the avalanche spike. Then, the avalanche current is quenched to reset that SPAD. There are both passive and active quenching schemes. While quenching, the voltage across the SPAD drops to the breakdown level, from which it resets to the normal reverse bias level. If a photon arrives during the quenching and reset phases, it will not trigger the SPAD reliably. Hence, if two photons arive one after the other, the first one is detected, whereas the second is not.

If the detected light intensity is high enough, more than one photon can arrive on the SPAD per laser pulse cycle, causing a phenomenon known as pile-up. In TCSPC-SPADs, the TCSPC only tags the first-arriving photon after a laser pulse, hence photons that arrive later go undetected. This causes an intensity-dependent bias towards detecting early photons, such that at high light intensities, photons appear to 'pile up' near the time-axis origin (the excitation pulse) [71]. In time-gated SPADs, the TCSPC tagging problem appears to be mitigated, as the time-gate only reads a specific portion of the temporal signal at each gate position. However, time-gated SPADs typically have a binary (0 or 1) output per gate position, per laser pulse. Consequently, even though time-gated SPADs are less preferrential to piling up photons near the time-axis origin, they still have a non-linear intensity response, such that the percentage of undetected photons increases with detected light intensity. To put it another way: if the time-gate detects a 1 on every laser pulse, filling its dynamic range over the course of the measurement, and despite the Poissonian nature of photon emission, none of the laser pulses went by without triggering a detection event, chances are, more than 1 photon arrived on many of these pulses.

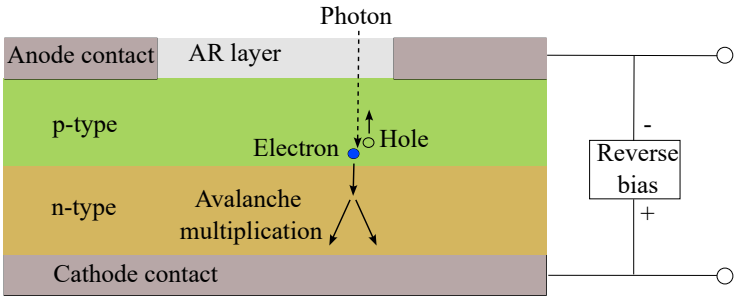

Figure 3.1: Schematic of a single-photon avalanche diode (SPAD).

3.2 Method

Experimental parameters

Setup. The experiment consisted of a pulsed 470nm HORIBA DeltaDiode laser with a pulse width of 47ps, illuminating a roughly $330 \times 180 \mu m$ field of view. Our setup used an epifluorescence design, meaning that illumination and collection of fluorescent emission were done through the same microscope objective, as shown in Fig. 3.2(a). Fluorescence was gathered by a time-gated[4] megapixel SPAD array developed by Eduardo Charbon's group [26]. However, the SPAD array was divided into two sections, as shown in Fig. 3.2(b), of which only one had

[4]See Sec. 2.1.3 for an overview on time-gated FLIM.

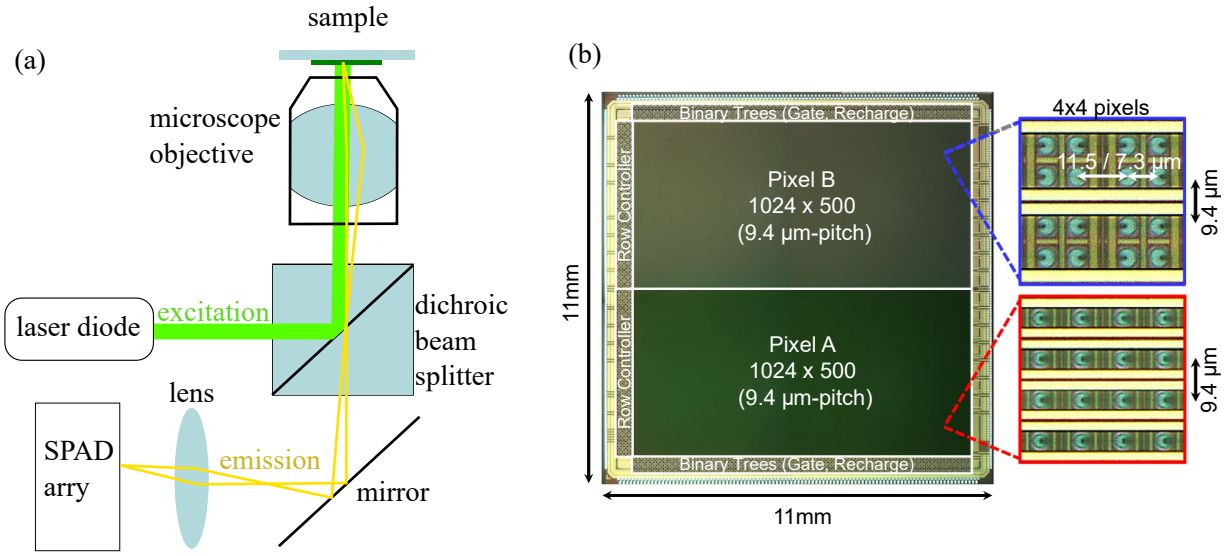

Figure 3.2: **(a)** Schematic of our epifluorescence FLIM setup. A pulsed laser diode illuminates the sample through an objective. The sample's fluorescence emission is imaged onto a SPAD array. A dichroic beam splitter ensures that the excitation light reaches the sample, and that the emission light reaches the SPAD. **(b)** Schematic of the SPAD array, adapted from [26].

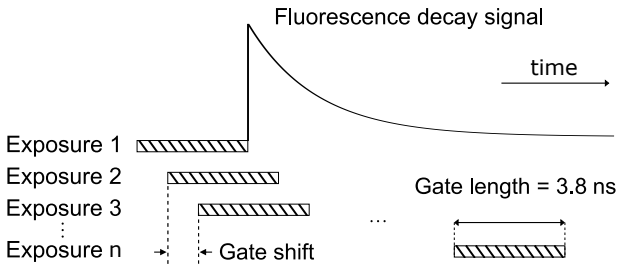

Figure 3.3: Schematic of the gating process. At each exposure (each time-gate position) we measure the dot product of the gate gain with the fluorescence decay signal.

time-resolved imaging capability. Therefore, our experimental setup gathered 0.5 MP frames by scanning an electronic time-gate.

The gate's shape can be found by scanning a narrow pulse (for example, the emission of an ultra-short lifetime sample) across it, or more typically, by scanning the gate over a fixed pulse [5]. Each measurement is the integral of the fluorescence decay multiplied by the gate amplitude, i.e. the dot product between the time-gate and the decay signal. Consequently, the full scan gives the cross-correlation (or equivalently, convolution with a time-inverted gate) of the gate G with the fluorescence signal A:

$$s(t) = (G \star A)(t) = \int_{-\infty}^{\infty} G(\tau - t)A(\tau)d\tau \tag{3.1}$$

The gating operation is visualised in Fig. 3.3.

SPAD parameters The SPAD had a few experimental parameters that needed to be adjusted for. Since our pixels were SPADs, our data experienced pile-up, which resulted in lower peak counts. This phenomenon was independent of the time-gated readout and is purely a property of the SPAD array itself. A raw experimental measurement is compared to its pile-up corrected counterpart in Fig. 3.4(a).

The gate was experimentally measured, and subsequently modelled as a super-Gaussian:

$$g_N(t) = \exp\left[-2\left(\frac{t-t'}{w_N}\right)^{2N}\right] \tag{3.2}$$

where t' is the centre of the gate and w_N is its width, related to the full width at half maximum

[5] If the gate is scanned with respect to a fixed pulse, the gate measurement will be time-inverted, as shown in Appendix Sec. 8.0.5.

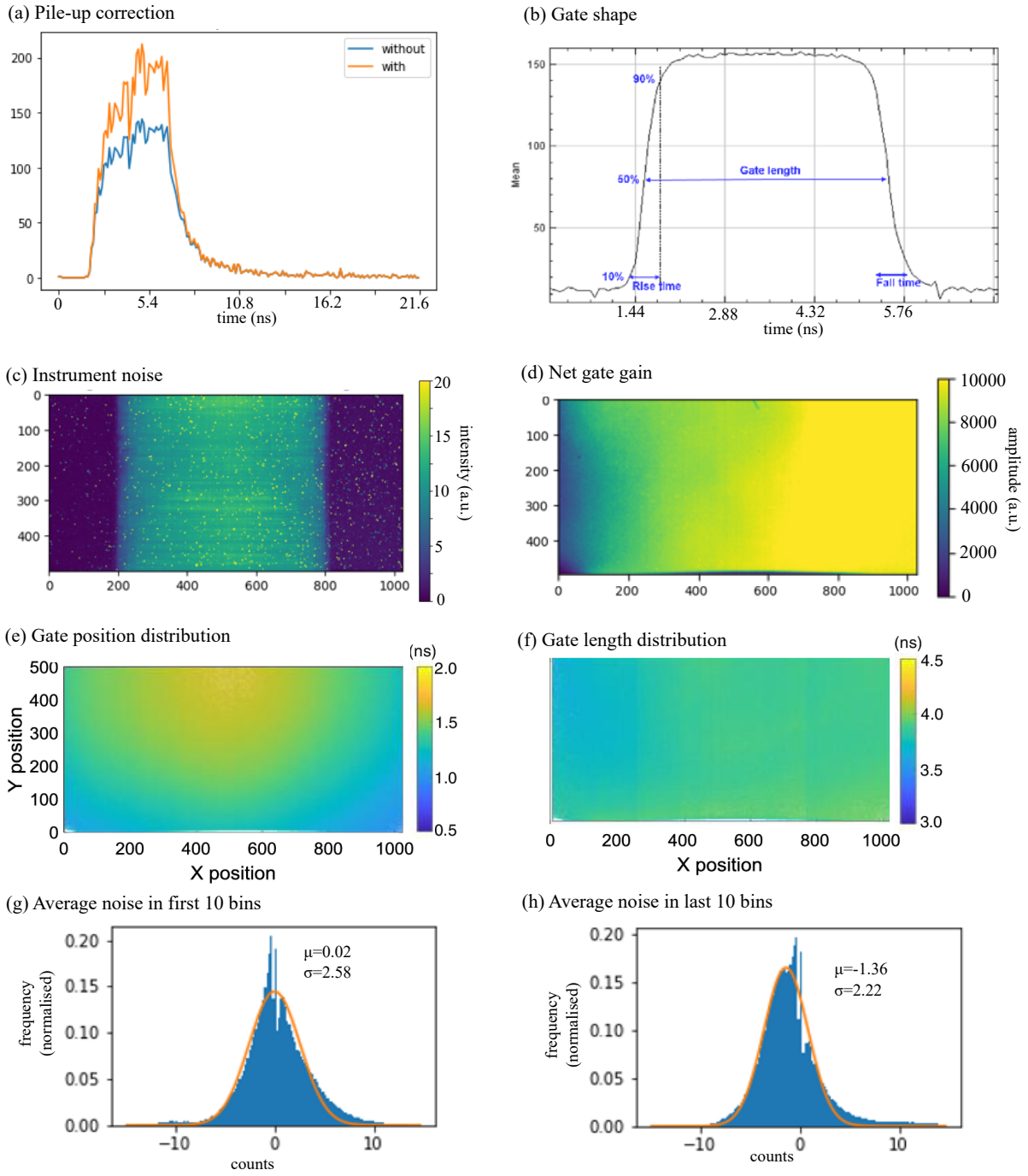

Figure 3.4: **(a)** An experimental decay profile, with and without (raw data) pileup correction. **(b)** Gate profile. **(c)** Pixelwise distribution of the instrument's peak noise level, highest in the central columns. **(d)** Pixelwise distribution of the net gate gain. **(e)** Gate [starting] position, adapted from [26] **(f)** Gate length (average $\sim 3.8ns$) [26] **(g)** Noise from first 10 pixels. **(h)** Noise from the last 10 pixels.

(FWHM) by:

$$w_N = \frac{FWHM}{2(0.5ln2)^{1/2N}} \tag{3.3}$$

The parameter N determines how rapidly the super-Gaussian rises, with N=1 being a regular

Gaussian bell curve and $N \to \infty$ tending to a tophat function. Our gate was 3.8 ± 0.2ns wide with a rise time of 0.55 ± 0.08ns, matched best by N=6. We show an experimental measurement of the gate in Fig. 3.4(b).

Our SPAD array's pixels were also not uniform in their properties. Noise level was uneven across the detector's pixels. Fig. 3.4(c) shows the peak noise in the various pixels of the SPAD array, acquired by observing the signal when the camera is not imaging a sample. Also, the centre of the instrument is noisier than the edges, and we see rows of pixels with high noise. Net readout gain (which is the gate's gain multiplied by its duration) varied from the left side of the detector array to the right - see Fig. 3.4(d). Other sources of uncertainty where the gate position (equivalently, the pulse arrival time with respect to the time-gate) and the gate length, which varied along the detector array, as noted by the designers of the instrument [26] - see Fig. 3.4(e-f). To adjust for these pixel-wise differences, we had to ensure our algorithm was robust to a wide range of experimental parameters simultaneously.

Noise statistics.

Based on experimental observation, we modelled measurement noise in terms of a Poissonian component and an independent and identically distributed (iid) Gaussian component. To find the Gaussian noise distribution, we looked at the data in the first and last ten times bins, which contain negligible fluorescence signal and thus have minimal Poissonian noise contribution. Analysis of three 0.5 MP images (background and pileup corrected) found that the Gaussian component had ≈ 0 mean and ≈ 2.3 count standard deviation. We show our noise analysis in Fig. 3.4(g-h).

The noisy measurement was therefore distributed as:[6]

$$\hat{f}(t) = X + Y \qquad (3.4)$$
$$\text{where } X \sim \mathcal{P}(f(t)), Y \sim \mathcal{N}(\mu, \sigma^2)$$

Experimental forward model. In our experiments, we were imaging convallaria cell samples stained with the fluorescent dye acridine orange (C-AO), and mammalian cancer (fibrosarcoma) cell samples, HT1080, stained with the fluorescent protein Clover (HT-C). The prior is expected to give a mixture of 2 lifetimes, with the component fraction varying depending on the biology of the sample. The latter is assumed to have a single lifetime.

[6] A more physically-accurate forward model would assume $\hat{f}(t) = \max(\hat{f}(t), 0)$ to ensure the non-negativity of counts. Instead, we performed background subtraction on our experimental data, allowing our effective counts to be negative, and modelled this processed data with Eq. 3.4. This resulted in the prior quoted 0 mean Gaussian noise distribution; the unprocessed noise mean would naturally be greater than 0

A mono-exponential fit can be used to approximate the mean lifetime of a bi-exponential mixture. Such a system has two lifetimes τ_1 and τ_2, each occurring with its own relative intensity $\frac{A_1}{A_1+A_2}$ and $\frac{A_2}{A_1+A_2}$ respectively, which we can call the component fractions. A mono-exponential estimate of a bi-exponential system varies such that, the higher the component fraction of the longer lifetime component, the longer the predicted mono-exponential lifetime. Conversely, if the component fraction of the short-lifetime component is higher, the mono-exponential fit will have a shorter lifetime. While mono-exponentials are not a precise replacement for a bi-exponential fit, in practice the reconstruction error in reconstructing a bi-exponential at small photon counts is higher than for mono-exponentials, making the latter preferable in low SNR scenarios [72].

Due to low to intermediate photon counts on the order of 100s or 1000s of photons in total per sample, we thus approximated f(t) with a mono-exponential decay:

$$f(t) = Ae^{\frac{t_0-t}{\tau}} H(t_0) \tag{3.5}$$

where A represents fluorescence intensity at $t = t_0$, which is the arrival time of the excitation impulse, and τ is the fluorescence lifetime. $H(t_0)$ is the Heaviside step function.

We can expand Eq. 3.4 to give the full forward model for the signal at a single SPAD pixel at gate position t_i:

$$\begin{aligned} &s(t_i, \tau) = X + Y, \text{ where} \\ &X \sim \underbrace{\mathrm{P}}_{\text{shot noise}}\left(\sum_{t=0}^{t_n} \underbrace{Ae^{\frac{t_0-t}{\tau}} H(t_0)}_{\text{fluorescence}} \underbrace{e^{-2\left(\frac{t_i-t}{w}\right)^{2N}}}_{\text{gate}} \delta t\right) \\ &Y \sim \underbrace{\lfloor \mathcal{N}(\mu, \sigma^2) \rceil}_{\text{bgd. + digitised thermal noise}} \end{aligned} \tag{3.6}$$

where the $s(t_i, \tau)$ represents an explicit dependence of the signal on the sample's fluorescence lifetime, τ. Here, δt is a normalisation factor - the sampling interval.

Machine learning for fit-free lifetime estimation

Having explored the forward model, the next step is to train an artificial neural network that finds the underlying lifetime of a measured fluorescence. The algorithm pipeline has three main steps:

1. Data step: obtaining and cleaning, or synthetically generating a dataset
2. Model step: Designing, tuning, training and testing the model

3. Pre-processing step: Pre-processing experimental data to match the model's expected input distribution. [7]

Data. We generated our dataset synthetically, since our forward model was well-defined, and we wanted to ensure that our algorithm was robust to a wide, continuous range of parameters, which is difficult to acquire experimentally. The dataset consisted of 4 million temporal decay measurements, split 2:1:1 for training, validation and testing, respectively. The decay curves were generated by randomly sampling decay lifetimes $t_0 \in [0.5, 5]$ (to account for variable sample lifetimes), the decay start $t_0 \in [5, 10]$ ns (accounting for varying readout times of different pixels), the decay amplitude $A \in [2, 32]$ ns (to allow for varying fluorescent intensity in a given bin and therefore variable Poissonian signal-to-noise ratio (SNR)), and finally, variable gate width $w_n \in [3.6, 6]$ ns [8]. Pixel-wise variation in the Gaussian noise were accounted for by sampling the mean and standard deviation randomly, $\mu \in [-2, 2]$ ns and $\sigma \in [0, 5]$ ns. These parameters were fed into Eq. 3.6 to produce the dataset. We sampled 200 time-gate positions t, at a 108ps interval (gate shift) making each measurement a 200-element 1D vector of photon counts. A separate dataset was made, with identical parameters, of 30 time-gate samples at 504 ps intervals, to investigate the influence of the number of samples (and total fluorescent intensity) on estimation error.

Model. Our philosophy was to build a small, fully-connected, feedforward ANN, with only a few layers. This choice makes sense in light of the algorithms used in fit-free lifetime estimation. As we derived in Chapter 2 (Sec. 2.1.7), RLD, phasor analysis and the central moment method all use some fixed linear combination(s) of the input and feed this combination through some simple non-linear functions.

A single node of a fully-connected ANN performs a single linear combination of the previous layer, adds a bias, and finally performs a non-linear function on this sum. Therefore, such a model can easily approximate the mapping performed by conventional fit-free methods. Our method can, however, learn these linear combinations and subsequent non-linear mixings freely, to adjust for uncertainty in parameters such as the gate or noise. Lastly, keeping the model and its parameter space small makes it computationally fast and helps prevent overfitting. These traits motivated our model choice.

We use a custom ANN consisting of an input layer (IL), an output layer (OL), and three

[7] I include this as a separate step from the acquisition and cleaning of training/validation/testing data for several reasons. Firstly, it pertains to using the finished model on new data; training/validation/testing data can be discarded at this stage. Pre-processing must be performed even by the end-user, who will import a pre-trained algorithm. This has implications for computational speed: a real-world user will observe a computational time which is the sum of the pre-processing time and the model runtime. Moreover, models are typically trained on 'nice' data to assist convergence. Experimental data might require some pre-processing to match the distribution of this 'nice' dataset.

[8] This gate-width range exceeded the observed gate variation, nonetheless the method worked well, demonstrating its robustness for semi-blind deconvolution.

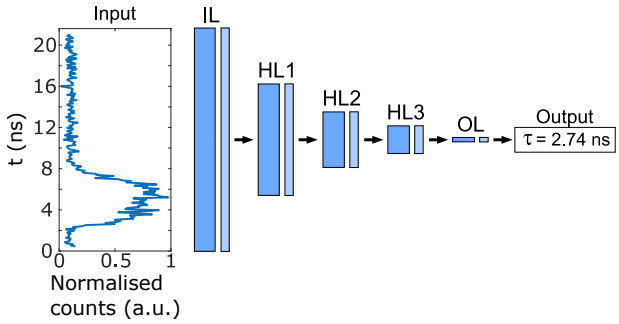

Figure 3.5: Overview of the ANN architecture. The input layer is followed by hidden layer 1 (dense, 100 nodes, ReLU activations), then hidden layer 2 (dense, 50 nodes, ReLU), then hidden layer 3 (dense, 25 nodes, ReLU), and finally the output layer (dense, 1 node, ReLU [0.5,5]).

hidden layers (HLi, with $i = 1, 2, 3$) connecting the IL with the OL, as depicted in Fig. 3.5. Each of these layers is formed of a fully-connected dense layer followed by with rectified linear unit (ReLU) activation function. The IL (with $n_0 = 200$ nodes) is fed with a fluorescence decay signal (normalised to the range [0,1]), i.e. with a 1D vector with as many elements as the number of gate shifts (200 in our work). Then, the output of the IL is fed in cascade through the ANN, while the number of nodes of each subsequent HLi decreases as $n_1 = 100$, $n_2 = 50$, and $n_3 = 25$. Finally, the OL provides an estimate of the lifetime τ of the fluorescence decay.

Training was standard: we used mini-batch gradient descent with a batch size of 128, using adaptive moment estimation (Adam) as our gradient descent algorithm. Our loss function was the mean squared error (MSE) between ground truth lifetime values and the corresponding ANN predictions. The number of training epochs was validated using the validation set, early-stopping the training after 15 epochs without improvement on the validation set. Computations were carried out using *Tensorflow* 1.14.

The validation set and test set had an approximately equal loss, 10% larger than the training set loss. On our 200 time-bin test dataset, we obtained $\sim$ 0ns average error (the estimator was unbiased) and 0.072ns standard deviation. On our 30 time-bin data, we got likewise $\sim$ 0ns mean error, with 0.250ns standard deviation.

Pre-processing. We first performed pileup correction on the data. The pileup corrected intensity I_{corr} is [73]:

$$I_{corr} = -I_{max} \ln(1 - \frac{I_{rec}}{I_{max}})$$

where I_{max} is the maximum possible count we can observe for a given bit depth (e.g. 255 for 8-bit intensity measurements), and I_{rec} is the recorded photon count.[9]

[9]Ulku et. al. note that pile-up correction decreases the SNR of a measurement, such that the pile-up corrected

Experimental data was then background-corrected by subtracting the mean of the first 5 frames from each measurement. These frames are assumed to contain only dark counts and thermal noise since they are measured before the arrival of the laser pulse. Afterwards, the measurement was divided by its maximum (upper bound normalised to 1) before fitting.

3.3 Results

Benchmarking against LSF on experimental data

Our ANN was designed with two objectives: (a) robust, precise and accurate lifetime estimation, and (b) rapid computation. To demonstrate the robustness and low uncertainty of prediction, we apply our method to some experimental datasets, comparing our predictions to a ground truth lifetime fit obtained with least squares fitting.

Convallaria - Acridine Orange. We imaged a sample of Convallaria, dyed with acridine orange (AO). When illuminated at 470nm, AO is expected to express 2 lifetime components: 0.98 ± 0.12ns with a component fraction of 51%, and 3.2 ± 0.13ns with a component fraction of 49% [74]. This may vary depending on whether the dye is in solution or bound to a cell, as well as on the environment of the dye. We can deduce that, since Acridine Orange has a *bi-exponential decay*, variations in lifetime reflect the component fractions of the short and long lifetime components. [10]

First, we got some high photon count measurements acquired over 10 seconds. This dataset was obtained by sampling 30 time-gates with a 504 ps gate shift and exposure of 330 ms per gate. We then used the least-squares (LSF) deconvolution and Artificial Neural Network (ANN) to extract the fluorescence lifetime. The LSQ processing time took 56 minutes, and we got a lifetime reconstruction of 1.29 ± 0.49ns. The ANN, with a processing time of 2.7 seconds, gave similar results, at 1.22 ± 0.27ns. Fig. 3.6(a-c) shows the results. Ergo, we found similar lifetime values 1244X faster.

We also captured low photon count measurements of the same Convallaria sample, but this time over just a 1s acquisition. To achieve this, we reduced the exposure to 33 ms per frame. 58 minutes of LSF yielded a lifetime distribution of 1.20 ± 0.53, whilst our ANN (processing time 2.7 s) gave a global lifetime distribution of 1.28 ± 0.34; the increase in speed is 1289X. The

signal is no longer Poisson distributed and has a higher-than-expected variance [73]. However, our noise analysis was performed on synthetic data, where pile-up was not an issue, hence this problem was avoided.

[10] While the mean of mono-exponential fits reflects the component fraction of the bi-exponential mixture, its standard deviation is also influenced by fluorescent intensity, as consequently, the SNR of the measurement.

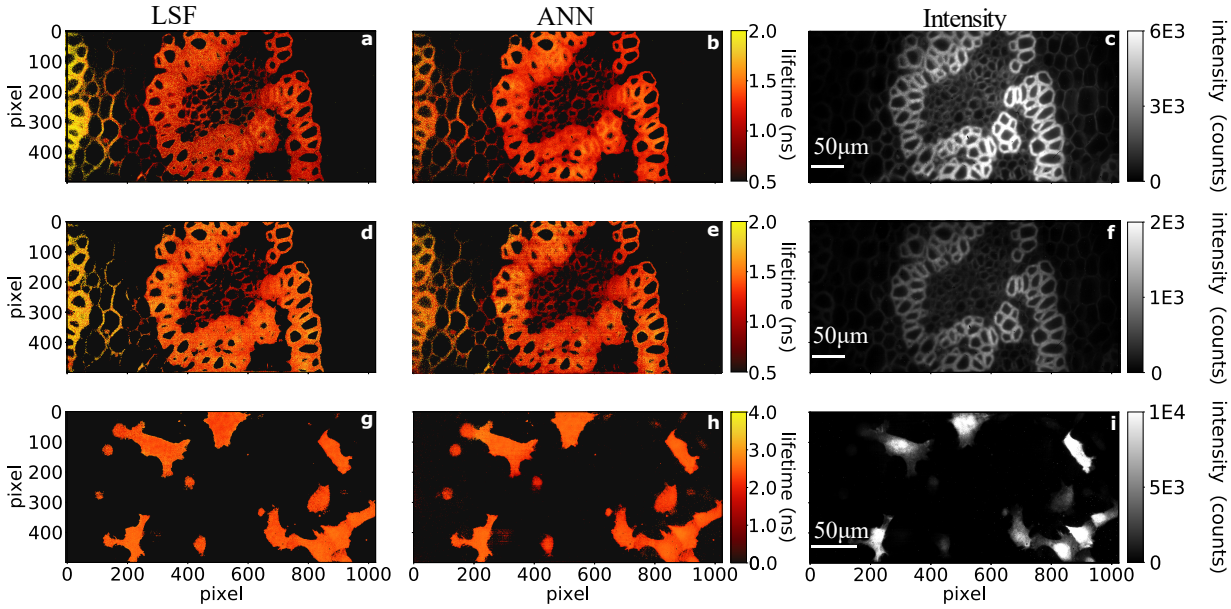

Figure 3.6: Wide-field fluorescence lifetime measurements of Convallaria-Acridine Orange and HT1080 cells. First column: least-squares deconvolution; second column: ANN estimation; third column: temporal sum of pile-up and background corrected intensity data, saturated to reveal dimmer structures. (a-c) We show high photon count measurements of convallaria, from a 10s acquisition. Spatial sampling is 0.47 μm/pixel with a 7% active area fill factor. (d-f) Low photon counts measurements of the same convallaria sample, acquired over 1s. (g-i) Measurements of HT1080 (fibrosarcoma) cells expressing Clover [75]. Spatial sampling is 0.33 μm/pixel. HT1080 was imaged for 400s.

results are shown in Fig. 3.6(d-f).

In the examples shown here, the total photon count in the LPC data falls below 2700 photons per pixel, whereas the HPC data exceeds 8500 (the intensity values in Fig. 3.6 are clipped to make dimmer structures more visible). Therefore, LPC data analysis is more challenging due to the lower signal-to-noise ratio (SNR). Nonetheless, both the LSQ and ANN methods recovered similar lifetimes for both HPC and LPC data. The agreement between the lifetime distributions of the ANN over the HPC and LPC samples is marginally better than for LSQ.

HT1080 - Clover. Convallaria-AO is commonly used for testing FLIM systems, as it yields a strong signal. We also wished to test our system against a simpler *mono-exponential* system, where lifetimes have one mean, and variance is a function of SNR and environmental/biological processes. Therefore, we dyed a sample of fixed HT1080 (fibrosarcoma) cells with pcDNA3-Clover. Clover is a green fluorescent protein, expected to decay with a single lifetime; up to our knowledge, literature does not give a clear estimate of the lifetime of clover samples at 470nm, as it is predominantly excited at the peak excitation wavelength of 505nm, at which point it has a single lifetime component of 3.2ns [76].

The HT1080 cell data was acquired using 200 gates scanned at a 108 ps gate shift. We found the HT1080 cells to be dimmer than the Convallaria cells: HT1080 cells yield around 100 photons per second on average in the brightest region, compared to around 2500 in the brightest region of Convallaria. To account for this, we imaged longer, with an acquisition time of $400s$.

Results are shown in Fig. 3.6(g-i). LSF yields a lifetime distribution of 2.41 ± 0.29 while the ANN yields lifetimes of 2.31 ± 0.34. Although neither method predicts a precisely uniform lifetime distribution, lifetime estimation has inherent uncertainty due to the limited SNR of the system. Likewise, biological measurements always incur some lifetime uncertainty. In other words, since our systems are not biased in any way to output a fixed lifetime for all pixels, our lifetime values are bound to have some spread.

3.6-megapixel mosaic. To further illustrate the computational efficiency of our ANN, we evaluated a 3.6 MP sample of convallaria dyed with acridine orange, made of a mosaic of eight 0.5 MP images. Although the images had a 10% overlap so that they could be stitched together in post-processing, all 4 MP of data was processed by the ANN, taking 36s. The final reconstruction is shown in Fig. 3.7.

Evaluation on synthetic data

Impact of Gaussian noise. In our first experiment, we examined the impact of the Gaussian (thermal) noise on the reconstructions. In summary, we found that the mean of the Gaussian noise biases our ANN, whereas the noise variance influences the uncertainty of the ANN's lifetime estimation.

The experiment went as follows. For lifetime values 1, 2.5, and 4ns, we created five datasets per lifetime, each containing 100 decay profiles. Using the terminology of Eq. 3.6, these 100 decay profiles had the same mono-exponential amplitude A, the same laser arrival time t_0, and the same gate shape; their only difference is that each decay has a different random noise sample.

The samples s had Poissonian $X \sim \mathcal{P}(x)$ and Gaussian $Y \sim \mathcal{N}(\mu, \sigma^2)$ components, such that $s = A + B$. The five datasets had varying Gaussian components:

$$B_1 \sim \mathcal{N}(0, 0^2), B_2 \sim \mathcal{N}(0, 5^2), B_3 \sim \mathcal{N}(0, 10^2), B_4 \sim \mathcal{N}(5, 5^2), B_5 \sim \mathcal{N}(-5, 10^2)$$

Datasets 1-3 seek to compare what happens in the presence of increasing Gaussian noise, i.e. increasing electronic jitter. Table 3.1 columns 1-3 show the results: we see increasing uncertainty in the lifetime estimate, with no clear trend on the change in the mean lifetime error (i.e. the bias).

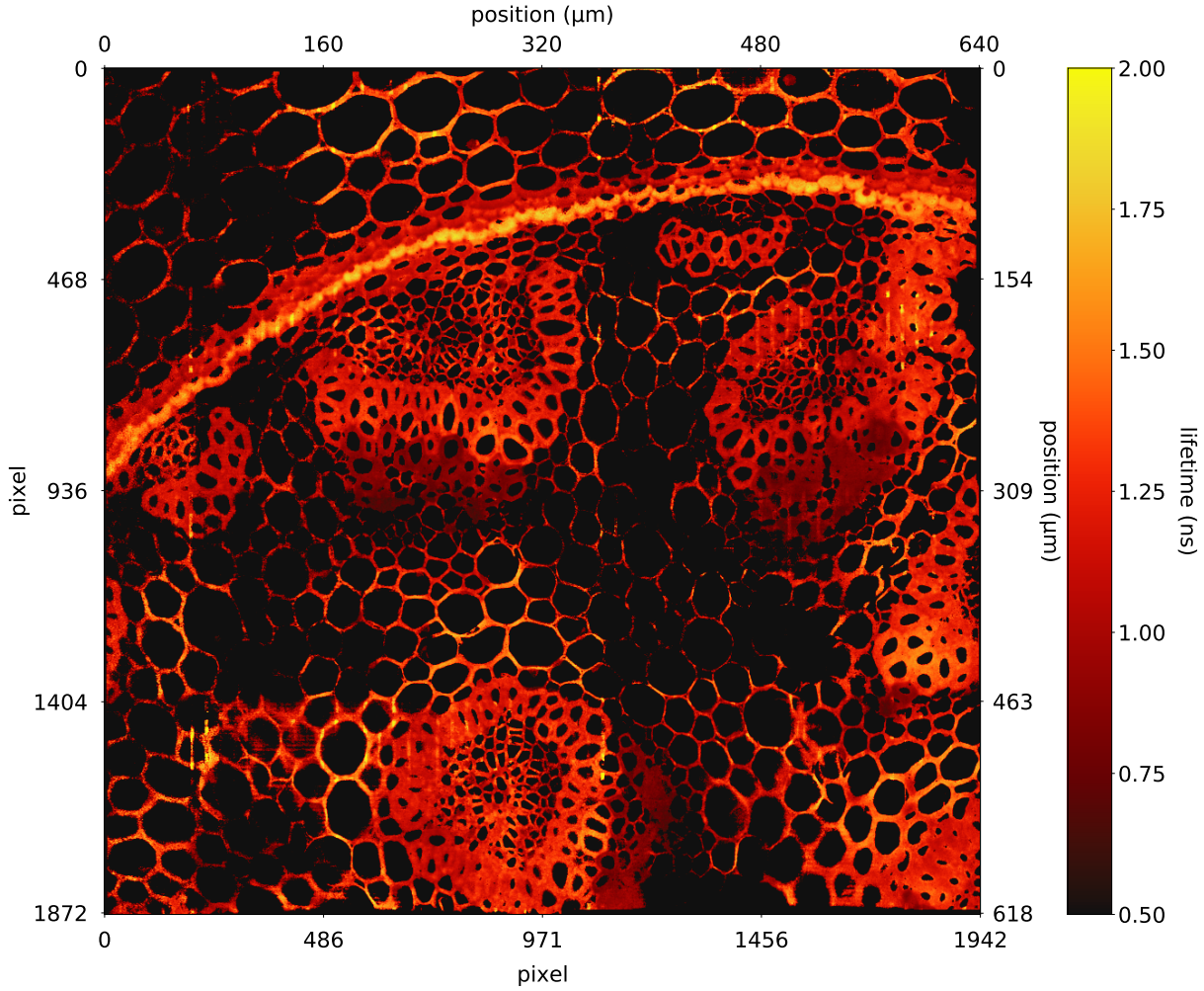

Figure 3.7: 3.64 MP (1875×1942 pixels) image of convallaria stained with acridine orange. The field of view is $\sim 618\mu m \times 650\mu m$ ($0.33\mu m \times 0.33\mu m$ per pixel). Total acquisition time was approximately 16 minutes in HPC mode (at the cost of SNR, this could be reduced to 10–20 s by operating in low photon count mode). The mosaic image was stitched using ImageJ BigStitcher [77].

Datasets 4-5 examine the effect of non-zero mean Gaussian noise, amounting to improper background subtraction. The results are shown in Table 3.1 columns 4-5: we observe that adding positive mean Gaussian noise gives a positive bias, whereas adding negative mean Gaussian noise gives a negative bias on the lifetimes. This matches our expectations: for instance, if a high background is added *to all* frames in the decay measurement, the signal will appear nearly flat over time, so an algorithm might think it has a long lifetime.

Impact of lifetime. Our second experiment evaluated the ANN's performance as a function of lifetime. We generated 9 sets of synthetic data using Eq. 3.4, each set containing 10 curves. The 9 sets were distinguished by their lifetimes, each having a fixed lifetime $\tau \in \{0.75, 1.25, ..., 4.75\}$. Two parameters were different between the nine datasets: the lifetime and the amplitude of the

GT	P(x)	P(x) + N(0,25)	P(x) + N(0,100)	P(x) + N(5,25)	P(x) + N(-5,100)
1	0.04±0.05	0.04 ±0.10	0.02 ±0.20	0.09 ±0.10	-0.01 ±0.17
2.5	0.01 ±0.06	0.00 ±0.10	0.03 ±0.15	0.06 ±0.11	-0.11 ±0.13
4	-0.01 ±0.07	0.00 ±0.13	-0.04±0.17	0.07 ±0.11	-0.18 ±0.17

Table 3.1: We show the distribution of lifetime prediction *error* as a function of measurement noise, where error = reconstruction − ground truth. The mean error gives the bias of the estimate, whilst the standard deviation of error gives estimation uncertainty. **(Column 0)** Shows the ground truth lifetimes, which are 1, 2.5, and 4ns. **(Columns 1-3)** The mean and spread of the lifetimes reconstructed from the noisy measurements. The noise model assumes a Poissonian distribution with added i.i.d. 0 mean Gaussian noise of increasing variance (0, 25, 100). The 0 mean Gaussian noise does not significantly bias the results in any scenario, but the uncertainty of our reconstructions increases with the noise level. **(Column 4)** We have Poissonian noise and added Gaussian noise of mean 5, variance 25. We observe positive bias, and a similar spread as in column 3, where the Gaussian variance was the same. **(Column 5)** Adding noise with negative mean results in a negative bias; we see similar uncertainty as in Column 3.

fluorescence decay. The latter was changed to keep constant PSNR: that is, the peak gate measurement value was kept constant. The 10 curves within the datasets were generated from the same set of parameters and differentiated only by randomly sampling their noise distribution.

The results are summarised in Fig. 3.8(a). We observe that the ANN is biased towards outputting values near the centre of the label distribution. We also observe that shorter lifetimes produce a lower lifetime uncertainty. Importantly, this experiment shows that our ANN yields a lifetime distribution of similar accuracy and precision as LSF, for a wide range of lifetimes.

Impact of intensity (i.e. Poissonian noise). We also examined the influence of SNR on

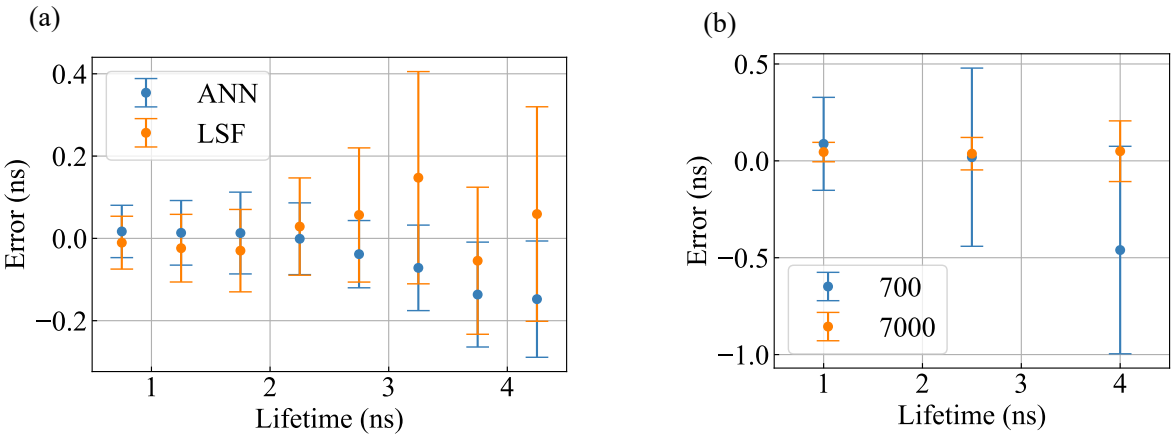

Figure 3.8: **(a)** We show lifetime predictions from the ANN benchmarked against LSF, for a range of lifetimes. Our estimator has similar uncertainty as LSF. However, its mean is slightly biased towards the centre of our training distribution, i.e. short-lifetime estimates are positively biased, while long-lifetime predictions are negatively biased. Uncertainty increases with lifetime. **(b)** Higher intensity creates higher SNR and lower prediction uncertainty.

prediction uncertainty, by contrasting the predictions of samples of 1, 2.5, and 4 ns lifetime with low SNR (total counts approximately [270, 700]) vs high SNR (total counts approximately [3000, 6000]). The noise distribution on these datasets was $s = X + Y$, where $X \sim \mathcal{P}(x), Y \sim \mathcal{N}(0, 5)$. Gaussian noise was constant to isolate the effect of Poissonian noise, i.e. fluorescent intensity.

Fluorescent intensity only impacts the Poissonian noise in our signal. Since the variance of a Poissonian distribution equals its expectation, the standard deviation of the measured counts in a given time bin is the square root of the intensity. Thus, the SNR associated with the Poissonian component of our noise scales as $1/\sqrt{\mathrm{N}}$, where N is the intensity of emitted fluorescence. Fig. 3.8(b) shows our results. Higher intensity creates lower prediction uncertainty, and higher lifetimes have higher uncertainty for given intensity. These observations both line up with the theoretical (Cramér–Rao) lower bound on the standard deviation of a shot-noise limited lifetime estimate [78]:

$$\sigma_\tau = \frac{\tau}{\sqrt{N}} F$$

where F is the photon economy, a detector-dependent factor.[11]

3.4 Discussion

This chapter presents a simple ANN to perform fit-free lifetime estimation, offering rapid and precise estimate of the fluorescence lifetime from the data stream of a 0.5 megapixel (500×1024 pixels) time-gated SPAD array. The ANN is trained on synthetic data that encompasses a wide range of experimental parameters. The method's accuracy (bias) and precision (uncertainty) are benchmarked on synthetic data. This technique enables real-time (up to 1 Hz) FLIM imaging with a SPAD array. To demonstrate this, we successfully applied the ANN to experimental data, reaching similar outputs as our implementation of the LSF gold standard at $\sim 0.1\%$ of the computational time.

The mosaic image shows clear structural lifetime variation, despite the fitting algorithm operating pixel-wise. Hence, we can safely say that most of the lifetime variation in this image is biological and not just noise/estimation uncertainty. That said, there are some artefacts (vertical stripes). These artefacts reflect columns of SPADs that have different properties than their neighbours. This also indicates that our ANN is not perfectly robust to variations of SPAD prop-

[11] See Appendix Sec.8.0.7 for derivation.

erties. In retrospect, I see two plausible explanations. The artefacts may have been a fault in the machine learning pipeline, or it might just go to show that different sets of parameters produce similar measurements, introducing degeneracy in the inverse problem of deducing lifetime from. This was not fully explored in this work and could seed future research.

Our model predicts lifetime one pixel at a time, giving it more flexibility than alternatives operating simultaneously on a larger FOV. We can therefore scale our estimator to FLIM measurements of any size or shape. However, this also means that alternative neural network architectures that operate on multiple pixels in parallel may be faster than our method. On the note of computational speed: although we showed intensity thresholded lifetime maps in this chapter, the ANN fitted all the pixels and thresholding was only applied in post-processing; hence the quoted computational times are not dependent on the sparsity of the sample or the choice of threshold. Further on the note of computational time: our LSF code was not thoroughly optimised; a GPU running a Laguerre-expansion-based implementation would be faster than our algorithm, decreasing the advantage of our ANN. Nonetheless, the ANN is undeniably a fast approach.

Regarding our evaluation of the impact of Gaussian noise (3.3.2), we generated mono-exponentials with constant amplitude A to create the synthetic dataset. A mono-exponential decay with lifetime τ and amplitude A has an intensity $I = A\tau$ - see Appendix Sec. 8.0.6. Thus, constant A means the fluorescent intensity is directly proportional to the lifetime. This explains why the spread of our prediction uncertainty over lifetime did not increase as much as expected from 1 to 4 ns: the uncertainty of a lifetime predictor is directly proportional to the square root of fluorescent intensity and inversely proportional to lifetime[12]:

$$\sigma_\tau \sim \frac{\tau}{\sqrt{I}} = \frac{\tau}{\sqrt{A_0 \tau}} = \frac{\sqrt{\tau}}{\sqrt{A_0}}$$

With regards to our synthetic dataset, we pointed out that we had made 2 models, one using 200 time-gates and another using 30 time-gates; we achieved a test set root-mean-squared error (RMSE) of 72ps on the prior, and 250ps on the latter. This is true, however, we must note that the datasets do not stand on equal footing. As we noted, the synthetic 200 time-gate data is sampled at 108ps gate intervals, whilst the 30 time-gate data used 504ps intervals. The signal amplitude was the same, not the total intensity; essentially, we assumed equal acquisition time *per gate*, not in total. Therefore, the 200 time-gate data had on average $200/30 = 6.67$ times higher total fluorescent intensity than the 30 time-gate data. In retrospect, the comparison between these two datasets is therefore not entirely fair, as the fundamental uncertainty of a lifetime estimator scales

[12]This relation is complicated by the presence of Gaussian noise, which is typically not accounted for in the Cramér-Rao lower bound of FLIM estimators.

as the square root of the total intensity measured by the system . Had we scaled the 30 time-gate data to match its intensity to that of the 200 time-gate data, we'd expect $\frac{200}{\sqrt{6.67}} = 96$ps standard deviation (if our estimator follows to CR lower bound), which is a lot closer to the 200 time-gate data than our quoted metric.

Chapter 4.

Encoding time in space: cavity FLIM

Summary. The current FLIM paradigm is to form an image of the fluorescent sample on the detector. This works like normal microscopy or camera imaging: one point on the object plane (sample) is captured on one point of the image plane (detector). The full time trace of the fluorescent decay is captured on the same pixel in the image.

Here we change this paradigm, using a system of mirrors to form an optical cavity that images time in space - spreading the temporal FLIM decay out over multiple pixels. Our novel FLIM setup allows us to capture fluorescence in a single shot but this comes at the cost of mixing spatial and temporal information, which makes it challenging to estimate the fluorescence lifetime of the sample. We analyse the inverse problem and demonstrate a physics-inspired CNN that solves this task. Our work was published in [2].

4.1 Introduction and related work

Need for FLIM speed. The field of fluorescence lifetime imaging (FLIM) has seen significant advances in recent years, however, the widespread adoption of FLIM in clinical settings is still hindered by the limited field of view (FOV) and slow acquisition speeds of commercial systems[79]. The primary limitation of traditional methods is that point-scanning is slow for high pixel resolution, precluding the acquisition of instantaneous full FOV information. To address this issue, recent research has focused on improving the acquisition speeds of confocal scanning systems using techniques such as resonant scanning and spinning-disc systems[80]. The detection systems are also being improved; conventional systems use photomultiplier tubes (PMT) and time-correlated single photon counting electronics (TCSPC), but recently, single-photon avalanche diode (SPAD) alternatives are also being explored. Nonetheless, for applications requiring dynamic scenes, large FOVs or plane-illumination schemes such as light-sheet imaging or total internal reflection fluorescence (TIRF) microscopy, wide-field systems are the optimal solution.

Single-photon avalanche diode (SPAD) arrays offer wide-field imaging in both TCSPC [81] and time-gated operation (see previous chapter). Wide-field FLIM systems can also use microchannel-plate (MCP) gates, acting as intensifiers for arrayed PMT detectors [18] or Charge-Coupled Device (CCD) cameras [82, 83]. An MCP-based intensifier multiplies incoming photons, improving the quantum yield of the detector and offloading some of the timing and sensitivity requirements

from the camera itself. So, intensifier-based systems benefit from high-fill factors and low noise, and are ideal for low-light imaging applications.

Space-time imaging.

Time-resolved imaging typically samples a signal into a sequence of frames, effectively generating a video showing the temporal evolution of the fluorescence emitted by the sample. TCSPC fills in the frames of the video in parallel by time-tagging incoming photons and adding a count to the correct frame. Time-gated sampling fills in the frames one by one.[1]. Instead of such video-based modalities, we now consider imaging techniques that encode the time-of-arrival of light onto pixel position on the detector, mapping time to space.

Streak imaging In the field of streak imaging, we spread out the temporal information of a scene spatially, by offsetting the detected light over a pixel array as a function of time [84]. Streak cameras are used to detect ultrafast temporal events, which are too fast even for regular time-resolved cameras to measure; their resolutions are in the pico [85], femto [86], and even attosecond range [87].[2] Applications include measuring events such as laser ablation, or laser-induced plasma formation. Streak camera technology is also used in bio-imaging, including fluorescence spectroscopy.

Streak cameras, shown in Fig 4.1(a), work similarly to cathode ray tubes, seen in Fig 4.1(b). Light from the sample is imaged onto a photocathode, converting photons into a proportionate number of electrons. The electron beam passes between charged plates and gets deflected by the electric field between them. The voltage across the plates is increased over time. Consequently, the electrons shift vertically by an amount that depends on their arrival time. The electron beams are then multiplied by a multichannel plate and converted back into photons by a phosphor screen. Therefore, the spatial distribution of detected light will match the temporal pattern of the fluorescent samples. This camera is similar to an iCCD; but instead of varying the voltage across the MCP to control the intensifier gain (creating a time-gate), we vary the voltage across electrodes in the intensifier, creating a basic cathode ray tube.However, while a cathode ray tube measures the voltage across the plates by measuring how much the electric field offsets a continuous electron beam, we use a known voltage ramp to measure the temporal distribution of the electron beam (equivalently, the light from the sample).

A streak camera avoids confusion between time and space in the output image by using a

[1] These imaging modalities are introduced in Chapter 2, Sec. 2.1.3 - 2.1.3

[2] Attosecond resolution streak cameras are not in wide use, and only exist in very niche applications, where they can be used to infer attosecond properties of x-rays. Their design principle also differs from the description in this thesis.

slit to image only a single line of the sample at a time. The slit is placed before the accelerating electrodes, cropping out the light from above and below the line of interest. Thus, assuming a horizontal slit and a vertical electron sweep, all light that we observe at our detector originates from one row of the sample, and the vertical offset from this line indicates the time-of-arrival of light at the given horizontal position, as shown in the output of Fig 4.1(a).

There exists a *wide-field streak imaging* alternative to the line scanning methods described above, known as compressed ultrafast photography (CUP) [90]. In CUP, a full frame is streaked instead of just a line sample. To distinguish the vertical sample position from the time of flight, a DMD encodes the sample in space. If the sample is sparse in space (in gradient domain), this allows the reconstruction of the spatiotemporal distribution of the sample from a single shot. This approach has been fine-tuned to image both DMD frames at once, in conjunction with an external CMOS camera that detects fluorescent intensity [91]. However, CUP and related approaches rely on streak cameras and leave the need for setups with simpler or cheaper components.

Optical cavity imaging. An alternate method of imaging time in space uses an optical cavity. An optical cavity is a resonator (something which makes light bounce back and forth), typically made of mirrors.

A laser cavity is a practical example of such a resonator. A basic laser cavity uses two mirrors (one partially reflective) to trap light in a round-trip path, increasing the net path length of light in the gain medium and allowing mode-locking. Its operation is visualised in Fig. 4.2(a). A

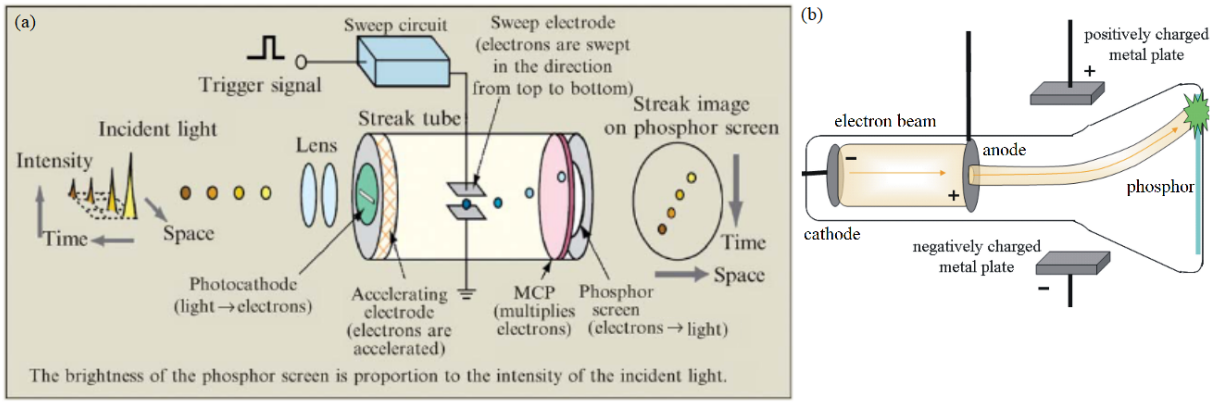

Figure 4.1: (a) Schematic of a streaking fluorescence lifetime measurement, from [88]. A time trace is shown separated in time and space. A photocathode converts photons into electrons, which are then swept vertically by an external electric field. The field ramps up over time, hence the time of arrival of electrons (and the light which made them) gets encoded in the electron beam's vertical position: electrons that arrive earlier experience a weaker field, and ones that arrive later feel a stronger field. The electrons are multiplied and converted back to light. Overall, in this setup, photons which arrive later are swept downward. (b) Schematic of a cathode ray tube, adapted from [89]. Its operational principle is analogous to that of a streaking camera.

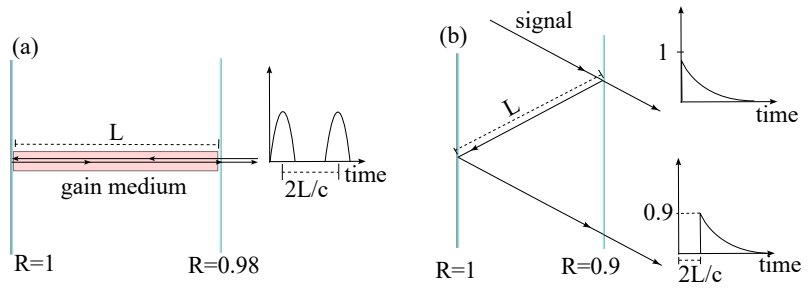

Figure 4.2: Overview of optical cavities. (a) Schematic of a mode-locked/pulsing laser cavity. (b) Schematic of a cavity made of a reflective and partially reflective mirror, in which light is introduced at an angle. The first signal (light which escapes immediately) arrives at a detector on the other side of the mirror at time $t = 0$. The second signal (light which escapes after 2 reflections) arrives at the same screen shifted in time by the time of flight of light in the cavity. Since there is no gain medium to increase the intensity of light, the intensity of the second signal also drops compared to the first.

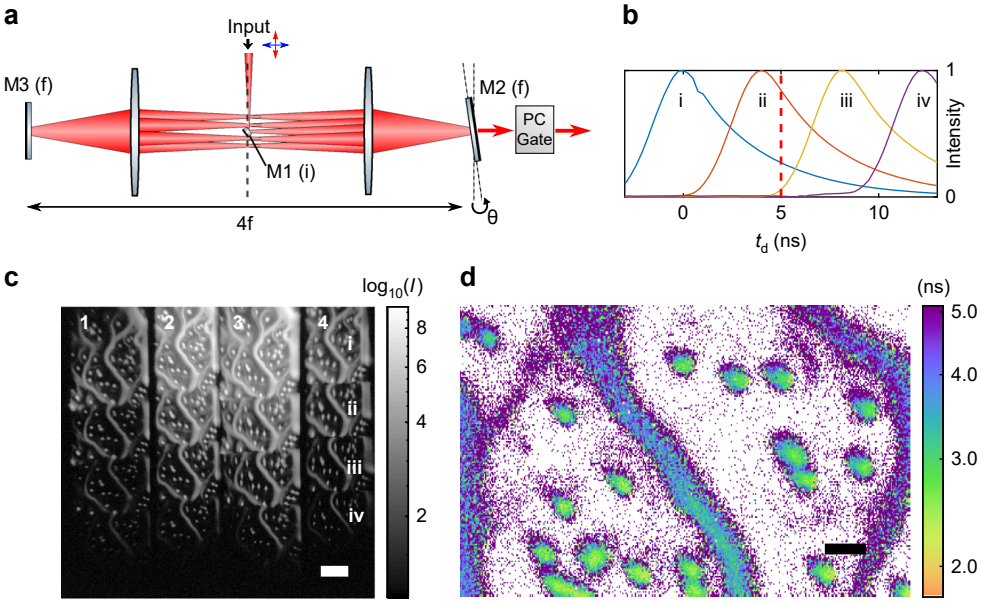

Figure 4.3: 4f optical cavity based FLIM, a single-shot time-gate scanning approach, based on an optical cavity for parallel scanning of multiple gate positions with respect to the signal. The approach and image are from [92]. (a) Schematic of a 4f optical cavity (f is the focal length of the lenses in the cavity). (b) The output signal over time. The red line shows where the first gate (reading from 0 to 5ns) ends and the second gate (reading from 5ns onwards) starts. The optical cavity shifts the signal over the gate, outputting multiple relative gate positions in parallel. (c) Shows an image of the camera readout. Columns (1-4) relate to how the authors use a Pockels cell for gating; columns 1 and 3 show the same signal, and their sum is the intensity in the first gate. The same holds for columns 2 and 4, which show the second gate. Rows (i-iv) show the various time-shifted signals, which are shifted vertically on the output. (d) Lifetime reconstruction from this time-gated data.

fraction of light exits the cavity through the partially reflective mirror, forming a visible beam.

There are several ways in which a cavity can map ToF to spatial coordinates. Fig. 4.2(b)

shows a simple schematic. In this scheme, light travels in an empty cavity at an angle, and 10% of it exits the right-hand mirror on each reflection, forming the output. Outputs at different vertical positions are shifted in time. This process does not directly encode information about the temporal properties of the signal, it simply delays the decay by the ToF of light in the cavity.

Bowman et. al. demonstrated a FLIM system using a Pockels cell as a time-gate combined with a 4f optical cavity that parallelises gate scanning [92]. The excitation path is normal, but the emission is imaged into an optical cavity, creating spatiotemporally offset copies/replicas of the emission image. The approach is summarised in Fig. 4.3. The cost of this approach is that 16 replica images of the sample form on the detector, therefore potentially limiting total FOV or resolution.[3]

4.2 Method

In this work, we demonstrate a method that combines the wide-field nature of CUP-based streak imaging and the parallel time-gate scanning of 4f optical cavity imaging. As with CUP, the temporal and spatial dimensions of fluorescent emission overlap across our detector's FOV, creating uncertainty in fluorescence lifetime estimation. To address this, we developed two solutions to extract fluorescence lifetime images: gradient-descent-based inverse retrieval and a physics-inspired ANN.

In this chapter, I discuss the experimental setup, the forward model mapping spatiotemporal fluorescence decay data onto our detector, and our algorithms, particularly the ANN. We validate the system on experimental data, and analyse the noise model of our measurements to derive the lower bound on reconstruction uncertainty with Bayes' theorem.

Setup

Fig. 4.5(a) illustrates our setup. A pulsed laser diode[4] illuminates and excites the sample through a dichroic beamsplitter and a microscope objective[5]. The same objective records emission from

[3] We have 8 time-gate samples, each of them having its intensity split between the horizontal and vertical beams' PBS output ports, giving a total of 16 images. If the emission signal is high and the collection SNR is sufficient, one might discard half of these images, since the horizontal and vertical emission contains degenerate information. This assumes the underlying fluorescent emission is unpolarised; this assumption may or may not be appropriate, depending on the sample [93].

[4] Horiba DeltaDiode DD-485L, emitting at 470 nm with 1 MHz repetition rate

[5] Zeiss 40× 0.75NA microscope objective

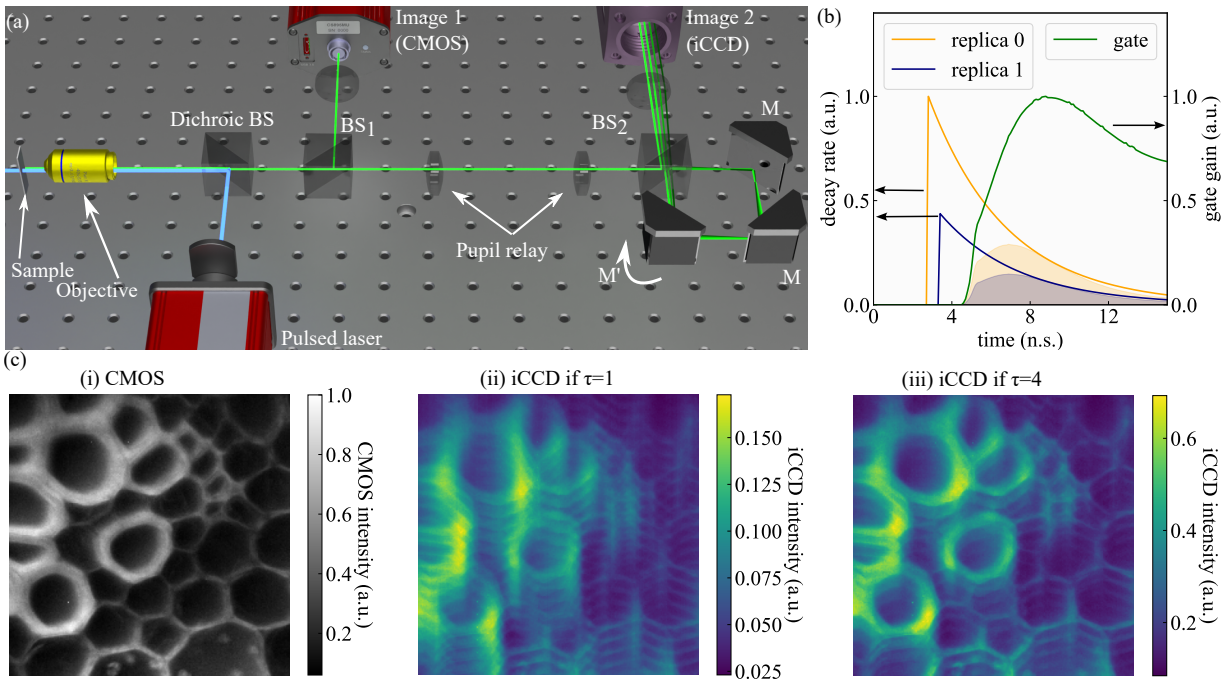

Figure 4.4: (a) Schematic of our setup. An optical cavity composed of a beamsplitter BS_2, two right-angle mirrors $M \times 2$ and a tilted mirror M' creates replicas of the signal. The replicas are offset in time by the round-trip ToF of light in the cavity and offset in space by the angular tilt of M'. (b) Schematic of the replicas and the iCCD gate (shown cropped to the first 15ns, for contrast). The measured fluorescence is the integral of the gated replicas, shown shaded. (c) We illustrate how our system detects fluorescence lifetime. (i) This CMOS image. (ii) The corresponding iCCD image if the lifetime is 1ns. (iii) The iCCD image if the lifetime is 4ns.

the sample. A 30:70 beamsplitter cube[6] imaged 30% of the light onto a CMOS camera[7] through a lens [8]. A pupil relay [9] was used to image the pupil of the microscope objective into our optical cavity. To summarise this part of the setup: 30% of sample fluorescence is imaged on a CMOS camera, and the remaining 70% (a far-field/Fourier space image) enters an optical cavity.

The optical cavity consists of a 50:50 beamsplitter[10] and 3 mirrors[11]. The beamsplitter cube accepts the remaining 70%, reflecting ∼ half of this to an iCCD[12]; we can call this the initial beam or the $0th$ replica. The beamsplitter transmits the other half into the cavity. Two mirrors[13] reflect

[6]Thorlabs BS019
[7]Thorlabs CS895MU
[8]f=200mm achromatic doublet (Thorlabs AC254-200-A-ML)
[9]Relay consists of a pair of f=250mm achromatic doublets (Thorlabs AC254-250-A-ML)
[10]The beamsplitter ratio is 50:50 nominally, we measured 51.5:48.5 experimentally at our given wavelength (Thorlabs CCM1-BS013/M)
[11]The mirrors are mounted on a kinematic mount (Thorlabs KM100CP/M) to allow fine adjustment of replica separation and angle; this adjustment had to be accurate within 0.1% for the cavity to function properly.
[12]Andor iStar 334T; it was triggered at 500 kHz (on every second laser pulse)
[13]Right-angle turning prism mirrors (Thorlabs CCM1-E02/M) were used

the signal by 180°, then a tilted mirror[14] sends the light back to the beamsplitter. Approximately half of this signal escapes the cavity at a slight angle compared to the initial beam, while the other half does another round-trip in the cavity. Again, half of this signal escapes, and so on.

Each successive beam is offset in space by a given angle compared to the previous one. The beams are in Fourier space and are re-imaged using a lens[15]. The lens converts the Fourier plane angular offset of light to an image plane pixel offset on the camera.

This system projects a set of replicated images of the object onto the iCCD, shown in Fig. 4.5(b). Each image replica is offset spatially from the previous by a few microns due to the angular offset of light in the cavity, and temporally by a few hundred picoseconds due to the time of flight of light in the optical cavity. Furthermore, each replica has ≈ 50% lower intensity than the previous one, due to the beamsplitter. Fig. 4.5(c) illustrates the measurements from samples of two different lifetimes with the same total fluorescent intensity. The higher the lifetime, the higher the peak intensity on the iCCD, as shown by the colourbar limits, and the faster the replicas decrease in intensity.

Forward model

A uniform sample of fluorescent molecules will decay according to a mono-exponential probability function. Our optical cavity replicates this decay signal and retards each replica in time by ∼0.59 ns, as shown in Fig. 4.5(b). The tilted mirror in the cavity shifts the retarded signal in space. Consequently, the iCCD intensifies different portions of the signal on different pixels.

The noise-free time-folded lifetime signal at a pixel $s_{i,j}$ on the iCCD visualised in 4.5(b)) is:

$$\begin{aligned} s_{i,j} = r_0 &\int_0^\infty G(t) A_{i,j} e^{\frac{t_0 - t}{\tau_{i,j}}} H(t - t_0) dt \\ + r_1 &\int_0^\infty G(t) A_{i,j-1y} e^{\frac{t_0 + t_c - t}{\tau_{i,j-y}}} H(t - t_0 - t_c) dt \\ &+ \dots \\ + r_N &\int_0^\infty \underbrace{G(t)}_{\text{gate}} \underbrace{A_{i,j-Ny} e^{\frac{t_0 + N t_c - t}{\tau_{i,j-Ny}}} H(t - t_0 - N t_c)}_{\text{fluorescence}} dt \end{aligned} \tag{4.1}$$

In words: if we assume the cavity replicates the signal vertically, shifting subsequent replicas *downwards*, this forward model tells us that a given pixel s_i, j captures the 0^{th} replica of the

[14] We used an elliptical mirror, mounted with a length-wise extended 3D printed mirror mount (Thorlabs H45E1)
[15] f=500mm achromatic doublet (Thorlabs AC254-500-A-ML)

same point on the sample, plus the first replica from a point *above*, and the second replica from a point even higher above on the sample, etc. The replicas are increasingly dim; the drop in intensity depends on how many round-trips light took in the cavity, denoted by $r_1, ..., r_N$ for the first N replicas. The splitting ratio of BS' at the given excitation wavelength was experimentally measured as 51.5:48.5.

$G(t)$ is the temporal profile of the time-gated intensifier on the iCCD camera. A is the amplitude of the decay, τ is the corresponding lifetime. $H(t-\alpha)$ is the Heaviside step function. The N terms represent N round trips; we determined N=6 to be a good cut-off point, as further round trips had too poor SNR to be resolved from noise. The image shear, measured in pixels per round-trip, is $y\mathbf{j}$. At each round-trip, the replicas are also delayed by the cavity round-trip time, t_c.

In parallel to the iCCD acquisition, a CMOS camera also acquires an image where each CMOS pixel i, j records an intensity

$$q_{i,j} = k \int_0^\infty A_{i,j} e^{\frac{t_0 - t}{\tau_{i,j}}} dt \tag{4.2}$$

which corresponds to the time-integrated fluorescence recorded at each pixel. Here, k is a factor that represents the amplitude ratio of the measurement obtained on the iCCD and CMOS cameras. This arises due to the product of the splitting ratio of BS_1, the pixel sizes and fill factors of the two cameras, and their respective quantum yields at the given emission wavelength. As such, k is constant over our experiments.

iCCD noise model. To figure out the noise model of our images, let us first consider how the cameras work. An intensified charge-coupled device (iCCD) is made of two elements: a multichannel plate (MCP) based intensifier, and a CCD. Light strikes a photocathode, converting photons into photoelectrons. These photoelectrons collide into the walls of channels in a multichannel plate (MCP); the collisions free more electrons, cascading to create an electron cloud. The voltage across the MCP controls how many electrons are produced in the MCP per photoelectron, i.e. the gain of the system. Finally, this electron cloud is converted back into photons by a phosphor screen. We can then measure this light with a CCD.

S_r is a variable that denotes the iCCD signal originating from replica r of some fixed point in our FOV. S_r assumes a value $\hat{s_r}$ in a particular measurement,[16]. We can approximate S_r as:

$$S_r \sim \mathcal{N}(s_r, n) \tag{4.3}$$

[16]Note: s_r is **not** s_{ij} from Eq. 4.1 which is the net signal on a given iCCD pixel. Rather, s_r is the intensity of a given replica of a sample. We can ignore which pixel it lands on.

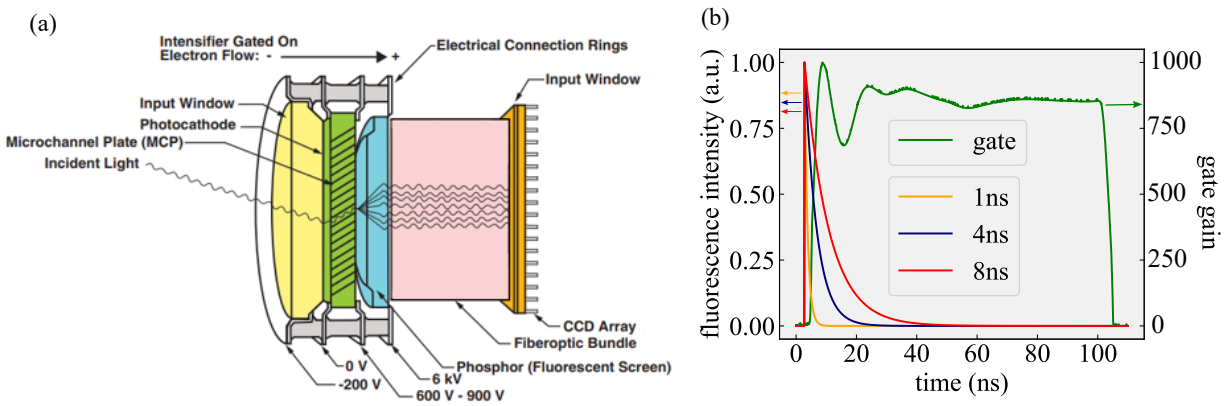

Figure 4.5: **(a)** Schematic of an iCCD, from [94]. **(b)** Gain (gate) of our iCCD, compared to a series of reference mono-exponential decays of 1, 4, and 8ns lifetime. The gate is $\sim$ 100ns long.

where n is the noise level of our measurement.[17]

We can describe s_r as the integral of the intensified signal:

$$s_r = \int_t M(t) D_{QE} P(t) dt \tag{4.4}$$

where M is the iCCD gain (i.e. the gate); D_{QE} is the detector quantum efficiency (basically the likelihood of detecting a photon, ergo the likelihood of generating a photoelectron and a subsequent electron cloud from a photon); and $P(t)$ is the signal (the number of photons falling on the pixel at time t) which is dependent on the sample lifetime τ[95].

The iCCD amplifies the Poissonian noise which originates from the quantised nature of photon emission events in the sample. It also amplifies dark current noise (as well as dark current itself) and clock-induced (spurious) charge noise. The noise on the CCD is referred to as readout noise, and it is not amplified, since the intensifier precedes it. Hence the total noise level, n, is:

$$\begin{aligned} n &= \sqrt{\sigma^2_{readout} + F^2 \int_t M^2(t)(\sigma^2_{signal}(t) + \sigma^2_{dark} + \sigma^2_{clc})dt} \\ \sigma^2_{signal}(t) &= D_{QE} \cdot P(t) \end{aligned} \tag{4.5}$$

where F is the noise factor of the amplification process itself. Our Andor iStar 334T uses an 18x-73 intensifier; according to the spec sheets [96], its quantum efficiency D_{QE} is around 20% above 500 nm, dark current noise is 0.03 e^-/s after noisy amplification, readout noise 8 e^-/s, clock induced charge noise 0 e^-/s.

[17] Note: In reality, S_r has both a Gaussian and Poissonian component, whose relative contributions vary with signal intensity. At low intensities, the Gaussian component dominates, while at large intensities, the Poissonian component may be approximated with a Gaussian.

4.3 Solving the inverse problem

To invert this forward model, we first pre-process our data and then apply some FLIM estimator.

We consider estimators for three different scenarios:

1. The simplest case is when replicas do not overlap, reducing our task to estimating the lifetime of a regular time-gate scan; fitting-based methods work well for such data.
2. A more challenging lifetime estimation task is when various replicas overlap. To solve this, we can either use inverse retrieval or machine learning.
3. The most challenging task is when replicas overlap but we have no CMOS image, reconstructing lifetime from the iCCD only. This we approach with machine learning.

Pre-processing and evaluation

The pre-processing pipeline to get from raw iCCD and CMOS images to lifetime maps had 3 steps. First, the iCCD and CMOS frames were background subtracted, with a separate background measurement. This is particularly important for the iCCD which has a high dark current; the CMOS had very little dark noise. Second, the iCCD image was rotated such that its replicas were vertical, like in our forward model.[18] Third, the CMOS image was transformed to match the iCCD FOV.

We note that the CMOS's pixel pitch in the object plane was smaller, and its image was neither in the same orientation, nor exactly co-aligned with the iCCD FOV. Therefore, we used an affine transform, linearly mapping the CMOS to the iCCD FOV by co-aligning a set of reference points on the two detectors.

The CNN pre-processing had three extra steps. The images were resized to match the replica separation of the training data, both images were normalised, and then the iCCD intensity was scaled by some constant k. This accounts for the difference in the intensity output of the two cameras when imaging the same scene.

For a single image, our pre-processing algorithm (for the CNN) has a wall time of 21.6 ± 0.1 ms and our trained ANN takes 71.1 ± 6.5 ms to estimate the lifetime from the pre-processed data. Hence the evaluation pipeline can run in real-time in parallel with a $\leq$10 Hz measurement.

[18] The replicas can arrive at any angle based on the relative orientations of the iCCD to the tilted mirror in the optical cavity.

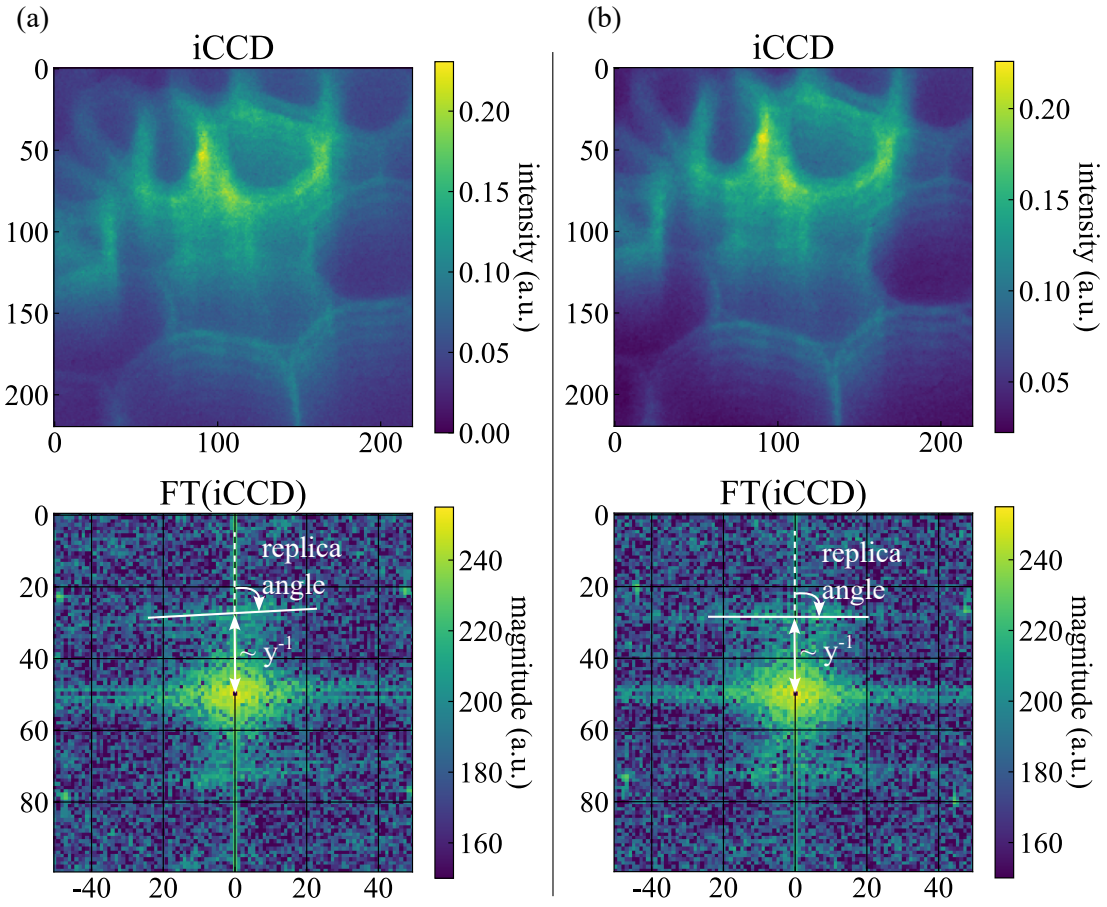

Figure 4.6: (a) An example of an iCCD image where the replicas are tilted by 5° from vertical. The replicas look approximately vertical by the naked eye, but the 2D FT reveals their precise angle and separation. (b) The same iCCD image, rotated to give vertical replicas.

Measuring replica separation and shear direction via 2D Fourier transform. For sparse data such as beads, it is straight-forward to estimate replica separation and the angle along which replicas are sheared. However, for complex samples such as cells, the overlap of the various replicas and the lack of sharp edges makes this task difficult to do by eye. Instead, these parameters can be found experimentally from the 2D Fourier transform of the iCCD image. The shear direction is important in our forward model, and offset from the desired angle can lead to artefacts in the form of vertical bands of higher/lower lifetimes. The impact of replica tilt on reconstructions is explored in the Results Sec. 4.4.2.

Fig. 4.6 demonstrates how replicas appear on the 2D Fourier transform (FT) of the iCCD image. Vertical replicas create horizontal bands above and below the central frequency cluster. The normal vector of these bands points in the same direction as the replicas, therefore the angle of the bands is offset from the replica angle by 90°. Further, the offset of the bands from the centre is inversely proportional to the replica separation along a given direction, e.g. the vertical offset scales as y^{-1}, where y is the vertical replica separation - see Eq. 4.1.

Non-overlapping replicas

When the replicas do not overlap, this system of equations forms a set time-gated FLIM measurements, where each replica is a given time-gate position, scaled by a scalar drop in intensity (introduced by the beamsplitter in the optical cavity). We can correct for the intensity drop by multiplying each replica with a factor, reducing the task to deconvolving a time-gated measurement. Lifetime can then be obtained using established algorithms (see Sec. 2.1.3); fitting-based deconvolution methods such as least-squares fitting or MLE are preferable to fit-free approaches for systems like ours, where the gate is long and non-uniform.

Overlapping replicas: inverse retrieval.

The IR pipeline is a process that improves our estimate of the lifetime map by iteratively comparing the expected iCCD signal to the measurement and adjusting the lifetime map accordingly. This is done using gradient descent to minimize the mean squared error between the expected and measured signals. However, this process can get stuck in local minima, so we use an L2-norm regularizer to constrain the IR. The optimization problem we are trying to solve is finding the optimal lifetime map, represented by $\hat{\tau}$, by minimizing the cost function $C(\tau)$ which is the sum of the data fidelity term and the regularizer, subject to the constraint that the lifetime map must be non-negative. Therefore, the inverse retrieval is a non-linear ridge regression problem:

$$\begin{gathered} \hat{\tau} = \arg\min_{\tau} C(\tau), \text{ where} \\ C(\tau) = \|\mathbf{P}(\tau) - \hat{s}\|_2 + \alpha\|\tau\|_2^2 \\ \text{subject to } \tau \geq 0 \end{gathered} \tag{4.6}$$

We minimise this cost function iteratively, via gradient descent. We initiate the optimization with a random guess for the lifetime map τ. We then take a step in the negative gradient direction:

$$\tau^{(n+1)} = \tau^{(n)} - \beta[\nabla_{\tau} C(\tau)]_{\tau=\tau^{(n)}} \tag{4.7}$$

Eventually, we reach the input τ that minimises C; see Appendix Sec. 8.0.8 for details. Inverse retrieval is a computationally slow but analytically tractable method for lifetime estimation in this scheme[19].

[19]The IR approach is analogous to least-squares fitting and maximum likelihood estimation in regular fluorescence lifetime fitting, i.e. we fit the measurement with an expected forward model, minimising some cost function iteratively via gradient descent)

Overlapping replicas: dilated convolutional neural network

We have developed a dilated CNN as an alternative and faster approach to retrieve experimental lifetime maps, instead of using the numerical inversion method. The CNN takes two inputs: the iCCD image and the CMOS image, and predicts the lifetime of the sample.

Data. 10,000 pairs of 270x220 pixel intensity and lifetime seed maps were used to generate 10,000 220x220 synthetic CMOS and iCCD image pairs. We used the forward model in (8.14), with model and noise parameters obtained from experimental data. The iCCD and CMOS images are smaller than the seed maps to allow replicas to enter the iCCD field-of-view from parts of the sample outside the detector limits. The network is trained on 7,500 synthetic input-label triplets (two inputs: 220×220 iCCD + 220×220 CMOS, label: 120×220 lifetime[20]), validated on 1,250 and tested on 1,250.

For the intensity seed maps, we upsampled images from the CIFAR-10 dataset, shown in Fig. 4.7(a)(i), and thresholded these by shapes made of lines (generated by mixing digits from the MNIST dataset), displayed in Fig. 4.7(a)(ii). This created line-like structures with varying intensity levels, which were then convolved with a relatively broad Lorentzian point-spread-function (Fig. 4.7(a)(iii)) to simulate out-of-focus planes, and remove sharp edges introduced by thresholding, giving the intensity seed map ((Fig. 4.7(a)(iv). For the lifetime seed maps, we upsampled different CIFAR-10 images - see (Fig. 4.7(a)(v)). We passed these intensity-lifetime pairs through the forward model, and cropped the outputs to allow replicas to enter the iCCD field-of-view from above - see Fig. 4.7(b)(i-ii).

Physics-inspired model architecture. We aimed to design a simple, physically sensible network - see Fig. 4.8 - to process our data and predict fluorescence lifetime, as in Fig. 4.7(b)(iii-iv). The model of choice is a fully convolutional neural network (CNN) that uses 2D convolutional layers with dilated kernels. The use of dilation in the kernels ensures that the feature maps are only dependent on the inputs in a causal way.[21]. The final Conv2D layer's (3x3) kernels mimic sliding window binning, which is commonly used in lifetime fitting to increase SNR.

[20]The lifetime label is only the central section of the FOV, as the network can see all replicas entering and leaving this region, making it the best to predict

[21]This notion of causal dependence might sound confusing. As mentioned earlier, our optical cavity overlaps replicas with one another, shifting replicas in space by 'y' pixels. This gives us **relation 1**: a pixel, say pixel 0, in the image plane (on the iCCD), depends on the same point, say point 0, of the *sample* in the object plane, as well as the point at -y, -2y, etc. Equivalently, we observe **relation 2**, that the properties of a pixel 0 of the sample (in the object plane) are described by pixels 0,y,2y,... on the iCCD. Substituting relation 1 into relation 2, we see that the lifetime estimate in a pixel 0 depends on pixels {0,y,2y,...} of the iCCD, which depend on pixels {(...,-2y,-y,0),(...,-y,0,y),(...,0,y,2y)} This helps to minimize over-fitting and aligns with the physics of the problem. Since the convolutional kernels maintain this dilated shape throughout the network, this causal dependence persists until the deeper layers. Hence, causal dependence of the lifetime in pixel 0 of the object plane (sample) is described appropriately by a dilated kernel, whose receptive field covers elements {...,-2y,y,0,y,2y,...} of the image plane.

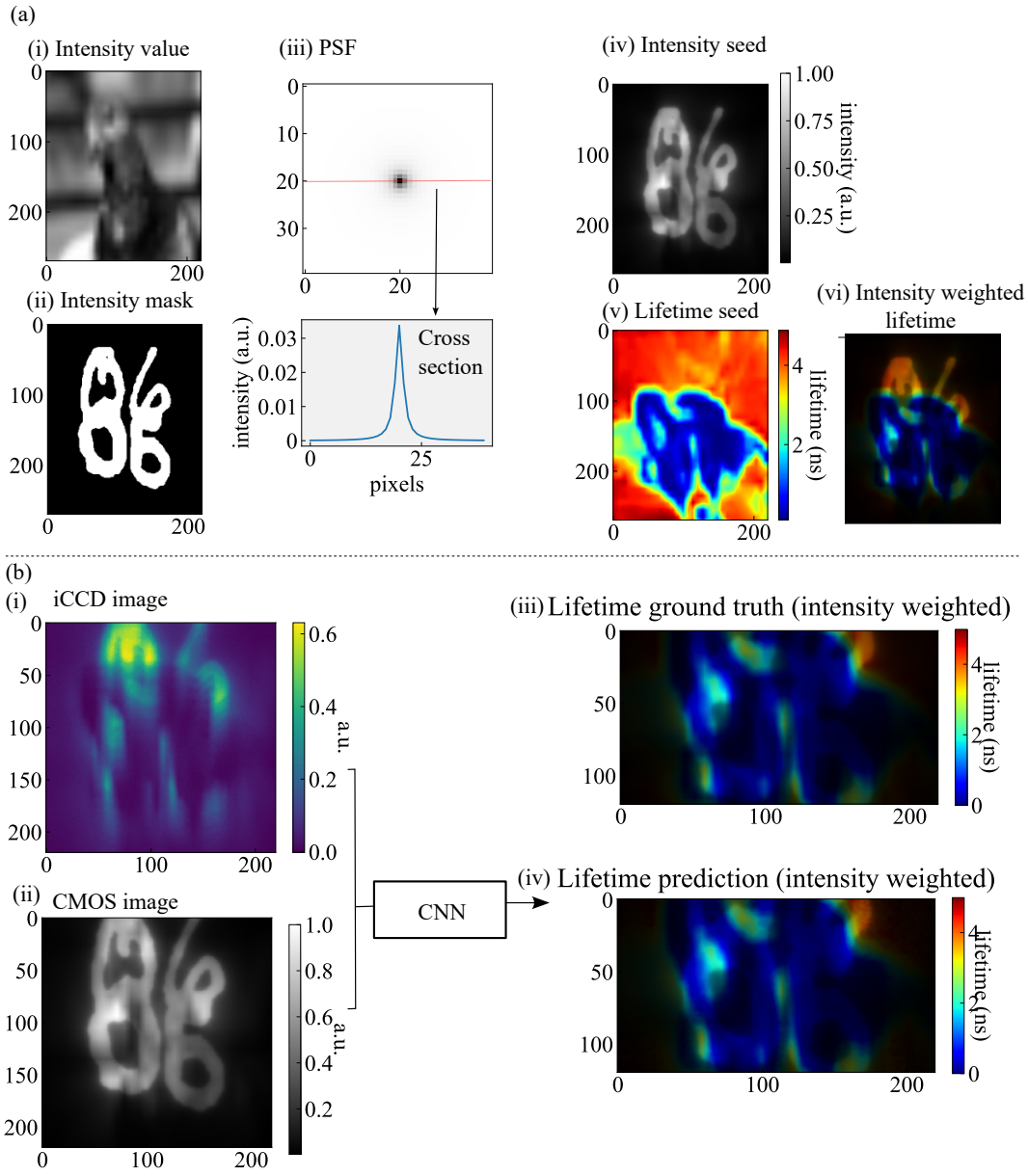

Figure 4.7: **(a)(i)** A synthetic intensity value map. **(ii)** A sharp synthetic intensity threshold map. **(iii)** We show the PSF, which is convolved with the mask to give a blurred mask (not shown separately). **(iv)** Intensity seed, generated by multiplying the intensity value image with the blurred mask. **(v)** Lifetime seed. **(vi)** Intensity weighted lifetime. **(b)(i-ii)** Passing the intensity and lifetime seed images through the forward model gives the iCCD image (the CMOS image is simply the intensity image, cropped). These inputs are used to train a CNN, which aims to predict the lifetime at the centre of the field of view, where incoming replicas and outgoing replicas are fully visible. **(iii)** Shows this region of the lifetime map. **(iv)** Shows the prediction of the CNN.

The CNN architecture is designed to match the physical parameters of the system and process the data efficiently. The two inputs are concatenated at the very beginning of the network because the CMOS data alone cannot provide insight into lifetime. So, feature extraction layers are not needed for the CMOS data. Similarly, the CMOS data adds valuable context to the iCCD image, so we do not process it on a separate branch either.

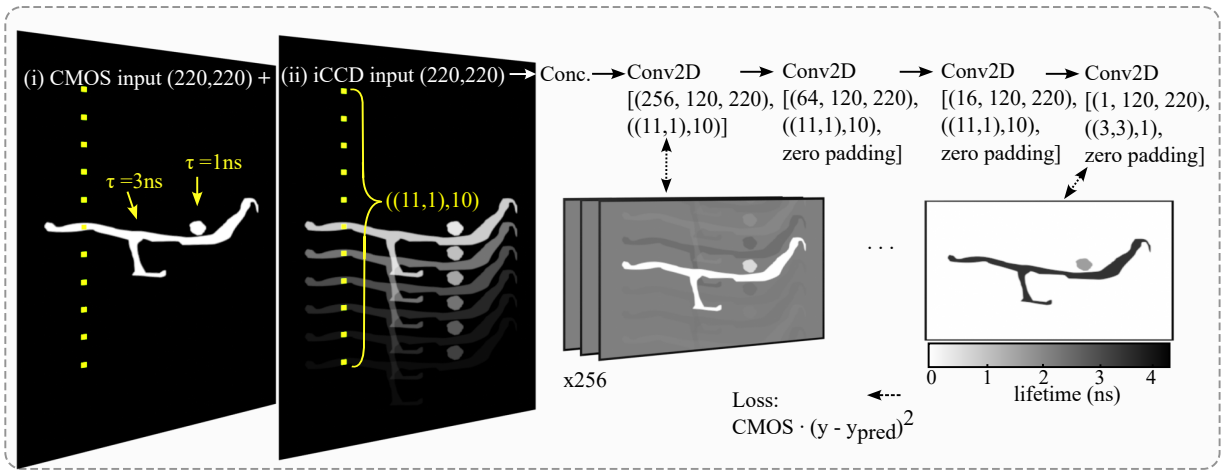

Figure 4.8: Schematic of our neural network.

After concatenation, the network convolves the inputs with a set of kernels. First, 256 dilated convolutional kernels are applied, without zero-padding. Each kernel's size is (11,1), designed to match our assumption of 6 replicas, with the kernel centred on the 0th replica. A dilation rate of 10 matches the pixel separation of replicas, creating a sparse receptive field of 101 pixels.

Another convolutional layer is applied sequentially, outputting 64 feature maps, followed by a third layer that outputs 16 feature maps. The kernel sizes are still (11,1) with dilation rates of 10. These first 3 convolutional layers perform a type of blind deconvolution, doing much of the processing. Lastly, 3x3 kernels output the final lifetime map. These 3x3 kernels help make the predictions smooth in a local 3x3 neighbourhood, informing each pixel of the lifetime of its neighbours. This is inspired by sliding window binning strategies, but instead of using a simple sum, our final layer uses a learned kernel. This reduces image artefacts that might be present if each pixel were treated independently. The output feature map contains the lifetime prediction.

Training. Our training loss was intensity weighted mean-squared error. In other words, for predicted lifetime maps $\hat{y}$ and the corresponding ground truth lifetime labels y, the CNN minimised:

$$L = \frac{\sum_{n=0,i=0,j=0}^{N,I,J} \mathrm{CMOS}_{n,i,j}(\hat{y}_{n,i,j} - y_{n,i,j})^2}{N \times I \times J} \tag{4.8}$$

where N is the batch size, I and J are the width and height of the image in pixels, and CMOS is the intensity field of view corresponding to the lifetime prediction.

Weighting the loss with intensity encourages the network to prioritise predicting the lifetime correctly for bright regions of the sample, and allows for more error in dim areas. The motivation for this loss function is that higher intensity areas have higher measurement SNR, and thus can be expected to be predicted with greater certainty. We visualise this by comparing the lifetime

maps in Fig. 4.7(e-f). It is difficult to accurately estimate lifetime in regions of very low intensity (even though these regions have some lifetime too).[22]

The model is trained for 200 epochs using TensorFlow v2.5.0, taking ≈ 6 hours on a GeForce RTX 2080 Ti GPU, using standard mini-batch (batch size 5) gradient descent (Adam optimiser) via back-propagation of our custom loss function, given in Eq. 4.8. Our test set loss was $0.031 ns^2\alpha$, where α is the arbitrary intensity unit in synthetic intensity images. If we evaluate only the mean-squared-error of lifetime reconstruction, without intensity weighting, our test set error is $0.37 ns^2$, so our estimation uncertainty (root-mean-squared error) is $\sqrt{0.37} = 0.61 ns$. If instead we threshold the predictions and ground truth lifetime maps to only evaluate pixels with intensity $> 10\%$ of the peak value within the FOV, MSE is $0.019 ns^2$, corresponding to an RMSE of $0.14 ns$. Our findings indicate that the CNN learns to reconstruct bright regions correctly.[23]

Overlapping replicas, iCCD only: dilated CNN

Our forward model shows that the various iCCD replicas act like distinct gate positions in a time-gated FLIM system. Hence, they carry enough information to allow lifetime estimation without the CMOS image. The question is, how much does prediction uncertainty increase if we ignore the CMOS?

To answer this, we performed a test on our bead dataset. For this, the neural network was retrained without the CMOS input branch.[24] See Appendix Sec. 8.0.9 for details; in short, we find that including the CMOS is important to maintain contrast between the lifetimes of the two bead populations. Therefore, our work focuses on lifetime prediction with both a CMOS and the time-gated iCCD, and all results shown follow this paradigm.

Uncertainty analysis

We use a Bayesian model and assume minimal noise in the CMOS to calculate the probability density function of the true sample lifetime given a lifetime estimate obtained from noisy iCCD measurements. This allows us to determine the lifetime estimation uncertainty of our system. Let us consider a signal of lifetime τ, creating an integrated CMOS measurement q and a series of noisy iCCD measurements $\hat{s} = (\hat{s}_1, \hat{s}_2, ..., \hat{s}_3)$. Here, $\hat{s}_r$ is the noisy iCCD signal in the r^{th}

[22] Regular fitting schemes often use intensity thresholding to decide which pixels to fit, and which have too low signal for fitting; our approach is similar, using a linear weighting instead of a hard threshold.

[23] See Appendix Sec. 8.0.11 for training and validation loss curves.

[24] We note that the CMOS was still used for the network training loss, but this only affects the training phase, since we do not need to calculate the loss to get predictions.

replica. We assume the CMOS has an SNR far better than the iCCD camera, therefore we ignore the CMOS in our noise calculations. Accordingly, we assume that the uncertainties in lifetime predictions originate from the iCCD only, and the probability density function of our predictions likewise.

Bayes' theorem states:

$$\begin{aligned} \text{posterior} &= \frac{\text{likelihood} \times \text{prior}}{\text{marginal}} \\ p(\tau|\hat{s}) &= \frac{p(\hat{s}|\tau)p(\tau)}{p(\hat{s})} \\ &= \frac{p(\hat{s}|\tau)p(\tau)}{\int_\tau p(\hat{s}|\tau)p(\tau)d\tau} \end{aligned} \tag{4.9}$$

Our prior, $p(\tau)$, assumes uniform probability density over lifetimes between 0.02 and 20ns. The distribution is uniform as we wish the system to be unbiased, while the limits are needed for the sake of numerical calculation. $p(\hat{s}|\tau)$ is derived from our noise model (for fixed total signal intensity), while the denominator is a normalisation term.[25] We derived the noise model in Eq. 4.5, as $S_r \sim \mathcal{N}(s_r, n)$

Since s_r is the noise-free signal of replica r and is bijectively dependent on lifetime (see Eq. 4.1), this distribution is equivalent to $p(\hat{s}_r|s_r) = p(\hat{s}_r|\tau)$ - see Fig. 4.9(a-b) for $p(\hat{s}_0|\tau)$ and $p(\hat{s}_3|\tau)$. Since our $\hat{s}_r$ is monotonically proportional to lifetime gives some noisy lifetime estimate,[26] we use the Jacobian transformation to write:

$$p(\hat{\tau}_r|\tau) = p(\hat{s}_r|\tau)\left|\frac{\delta\hat{s}_r}{\delta\hat{\tau}_r}\right|$$

Referring to Eq. 4.9, this is our 'likelihood'. We find the Jacobian $\left|\frac{\delta\hat{\tau}_r}{\delta\hat{s}_r}\right|$ numerically. Following Eq. 4.9, we then multiply $p(\hat{\tau}_r|\tau)$ by $p(\tau)$[27] and normalise[28], giving $p(\tau|\hat{\tau}_r)$. These probability distributions are shown in Fig. 4.9(c-d) for replicas 0 and 3. Finally, from $p(\tau|\hat{\tau}_r)$) we estimate

[25] Our calculations cover $\tau \in [0.02, 20]$ at uniform instances on a logarithmic scale, to provide better accuracy for low lifetimes.

[26] $\hat{s}_r$ is just a number since it is the measurement from a single replica; since s_r grows monotonically with lifetime, $\hat{s}_r$ is associated with a single noisy lifetime estimate.

[27] This is important since we evaluate lifetime instances on a logarithmic scale, but we wish to assume $p(\tau)$ has uniform prior probability density over [0.02,20] ns

[28] Normalisation refers to dividing by the integral in Eq. 4.9. Graphically, this means that every column in Fig. 4.9(c-d) must sum up to one: for any measured lifetime, the total probability of the true lifetime being within [0,20]ns is assumed to be 1. If our measured lifetime is close to 20ns, this causes our estimate to be negatively biased, as we make the assumption that the lifetime has to be less than 20ns; simultaneously, this causes its uncertainty to be artificially low. Equivalently, if we predict lifetimes at 0.02, which is lower range of our prior, our probability distribution of the true lifetime is positively biased. In fact, the range [0.02,20]ns was chosen to account for these numerical limitations, 'padding' our target probability distribution of [0.1,10]ns, so that our final result Fig. 4.9(f) could be evaluated without producing bias-effects at the lower and upper extrema.

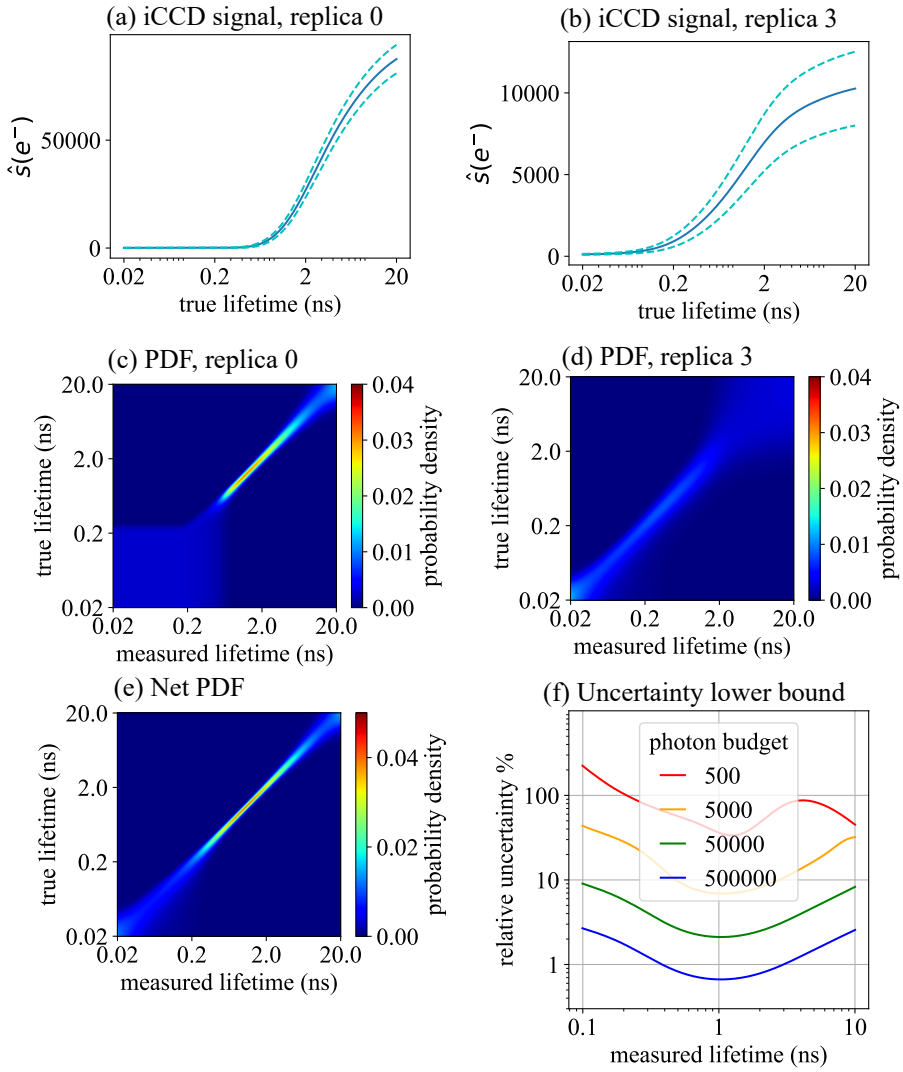

Figure 4.9: Illustration of our uncertainty analysis. **(a-b)** A sample of a given lifetime produces replicas on the iCCD, with intensity distributions given by Eq. 4.3. The photon budget incident on the iCCD is 5000 in these examples. The solid line shows the mean value μ, and the dotted lines show $mu \pm \sigma$. **(c-d)** From the iCCD values, the PDF of measured lifetime vs true lifetime is calculated for each replica. **(e)** The PDFs of various replicas are multiplied and renormalised to give the PDF of the prediction. The prediction uncertainty for a given lifetime is the vertical width of the PDF at the corresponding measured lifetime value. This PDF is derived for 5000 photons. **(f)** The [relative] uncertainty lower bound is shown for 4 photon budgets incident on the iCCD.

$p(\tau|\hat{\tau})$, assuming that each replica has independent noise:

$$p(\tau|\hat{\tau}) = p(\tau|\hat{\tau}_0, \hat{\tau}_1, ..., \hat{\tau}_n) = \prod_i p(\tau|\hat{\tau}_i)$$

The PDF is re-normalised since it is sure (probability of 1) that any measured lifetime corresponds to *some* true lifetime. Finally, we can estimate the relative uncertainty for a given lifetime

estimate:

$$\frac{\Delta_{\hat{\tau}}}{\hat{\tau}} = \int_t p(\tau|\hat{\tau})(\tau - \hat{\tau})^2$$

In Fig. 4.9(e), we estimate the probability of the true lifetime given our noisy measurements and use this to derive relative uncertainties of lifetime estimation for a photon budget of 5000. We repeat this pipeline for photon budgets of 500, 50k, and 500k, plotting the results in Fig. 4.9(f). We find that, with a photon budget of 5000, our lifetime prediction error is less than 10% for lifetimes between 0.48 and 2.8 ns. With a 10× larger photon budget of 50000, this range expands to 0.1 to 10 ns.

We note that this analysis just gives a lower bound: the real estimation uncertainty is higher, due to thermal noise and added Poissonian noise on the CMOS, as well as uncertainty introduced by overlapping replicas on the iCCD. Similarly to RLD [97], our measurements have minimal uncertainty around some lifetime domain, in our case ≈ 1 ns[29].

4.4 Experimental results

Beads - Yellow-Green & Dragon Green

We prepared samples of fluorescent beads of 2 μm (Merck, L4530) and 4 μm (Bangs Laboratories Inc., FSDG006) diameters. The bead samples were prepared by drop-casting isopropanol-diluted bead solution on a microscope slide, and washing with isopropanol to remove weakly adhered beads.

Yellow-Green dye (adhered to the 2 μm Merck beads) has been reported with lifetimes of 2.1 ns [98]. Separate reports of Dragon Green dye (4 μm Bangs Laboratory beads) excited at 485 nm claim 3.4 ns [99] and 4 ns lifetime [100]. We validated the lifetime of our samples using a separate TCSPC SPAD (FLIMera, HORIBA), and found lifetimes of 2.1 ± 0.05ns and $4.1 \pm 0.1 ns$ - see Appendix, Sec. 8.0.10 for details.

We used these beads to validate our cavity-based system, imaging them with a 1s exposure time. Fig. 4.10 shows results for a FOV containing several 2 and 4 μm beads. Our methods successfully distinguish the two lifetime distributions. We fitted bimodal skewed Gaussians to intensity-weighted histograms of the pixel values (the weighting is used to remove dark pixels). The CNN found lifetime populations of 2.0 ± 0.1 and 4.2 ± 0.7 ns, which are in good agreement

[29] The point of minimal relative error, i.e. the minimum of the plotted curves, is affected by the position of the gate with respect to the excitation pulse (moving the gate later is better for measuring longer lifetimes, earlier for measuring shorter ones), and the time delay between each sheared image (a longer time delay will mean replicas arrive at the gate later, optimising the system for resolving long lifetimes).

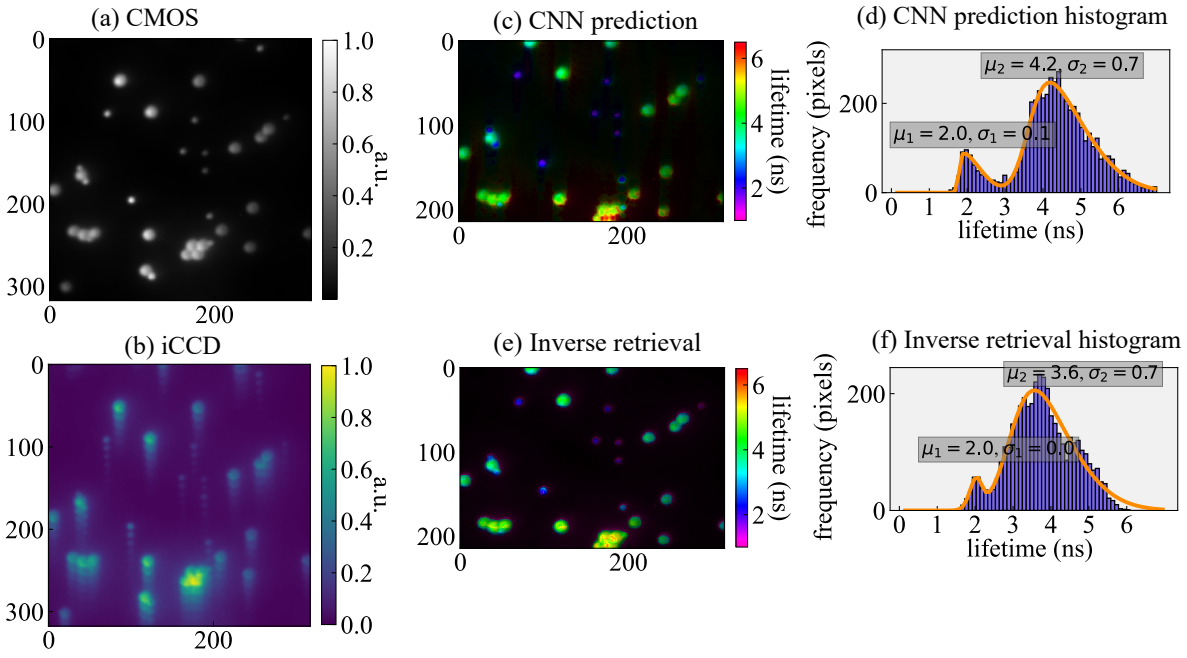

Figure 4.10: Fluorescence lifetime of beads acquired with our system. **(a,b)** We show a mixture of 2 and 4 μm beads on the CMOS and iCCD, **(c,d)** showing clear bi-modal lifetime distributions. **(e,f)** In orange we show a bimodal skewed Gaussian fit, with distributions of 2.0±0.1 and 4.2±0.7 ns, and 2.0 ± 0.0 and 3.6 ± 0.7 ns for the CNN and IR, respectively.

with our FLIMera validation distributions for the small and large beads, respectively. IR gave values of 2.0 ± 0.0 and 3.6 ± 0.7 ns, validating the CNN. The 2ns population has a smaller lifetime variance than the 4ns one, which is in line with our expectation from Fig. 4.9(f).

Convallaria - Acridine Orange

Next, we prepared Convallaria cells stained with Acridine Orange, obtained from Johannes Lieder GmbH & Co. KG (Catalogue number As3212). We imaged a variety of samples, all with similar properties; here we show two regions of interest. In the first, shown in Fig. 4.13(a), the CNN and IR retrieved lifetime distributions of 1.30 ± 0.09 and 1.29 ± 0.11 ns, respectively. Across the second region, they predicted 1.34 ± 0.11 and 1.30 ± 0.15 ns, respectively. These values are in agreement with our previously published results of 1.29 ± 0.49 ns, shown in Chapter 3 in 3.6(a). Also, the CNN and IR mostly agree on local lifetime patterns, cross-validating each other. We also note that these samples are not sparse like the beads, hence their time-folded replicas overlap on the iCCD, and yet our approach is capable of predicting their lifetimes.

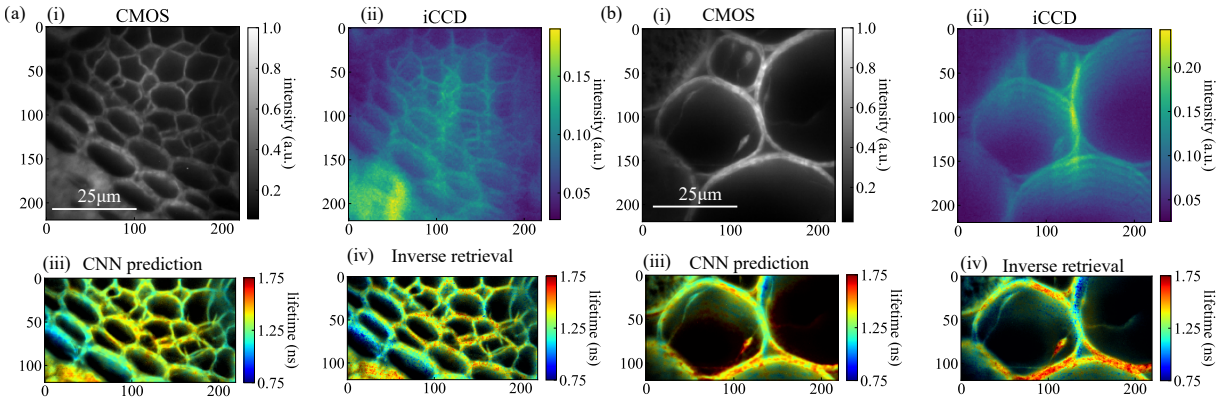

Figure 4.11: Lifetime map of Convallaria - Acridine Orange. The CNN and inverse retrieval estimate similar lifetimes patterns, but the CNN output is a little smoother, with more vertical artefacts, whilst the inverse retrieval has more granularity but fewer artefacts. These differences can be attributed to the final layer of the CNN, which acts as a form of (learned) sliding-window averaging, and the IR regulariser, which suppresses high-lifetime artefacts in low-intensity regions. **(a)(i-ii)** The CMOS and iCCD images of the first sample. **(iii-iv)** The CNN and IR lifetime reconstructions. **(b)** We show the same for a different sample.

Impact of replica tilt

Below we show results that demonstrate the impact of tilted replicas (shear not matching the expected direction), on further Convallaria Acridine Orange samples. Fig. 4.12(a) shows data where the optical cavity sheared replicas by 3° from the vertical, and corresponding reconstructions. Artefacts appear on either side of vertical structures. This is to be expected, since if replicas are tilted, the forward model predicts higher signal on one side of vertical structures, and lower on the other side, than is observed in the tilted iCCD image. Fig. 4.12(b) shows the same FOV, computational rotated to give vertical replicas. Reconstruction artefacts are suppressed. Fig 4.13 shows the same for a sample with a 5° shearing tilt from vertical.

4.5 Discussion

In this study, we present a novel method for obtaining fluorescence lifetime images, utilising a time-folded optical cavity together with an iCCD camera and a CMOS camera. Our setup samples different delays of the signal on different pixels of an iCCD gate, simulating time-gated measurements captured in parallel in a single snapshot. To predict sample lifetimes from this scheme, inverse retrieval and a physics-inspired neural network were explored.

We explored how much impact the CMOS has since it is not de facto necessary for lifetime

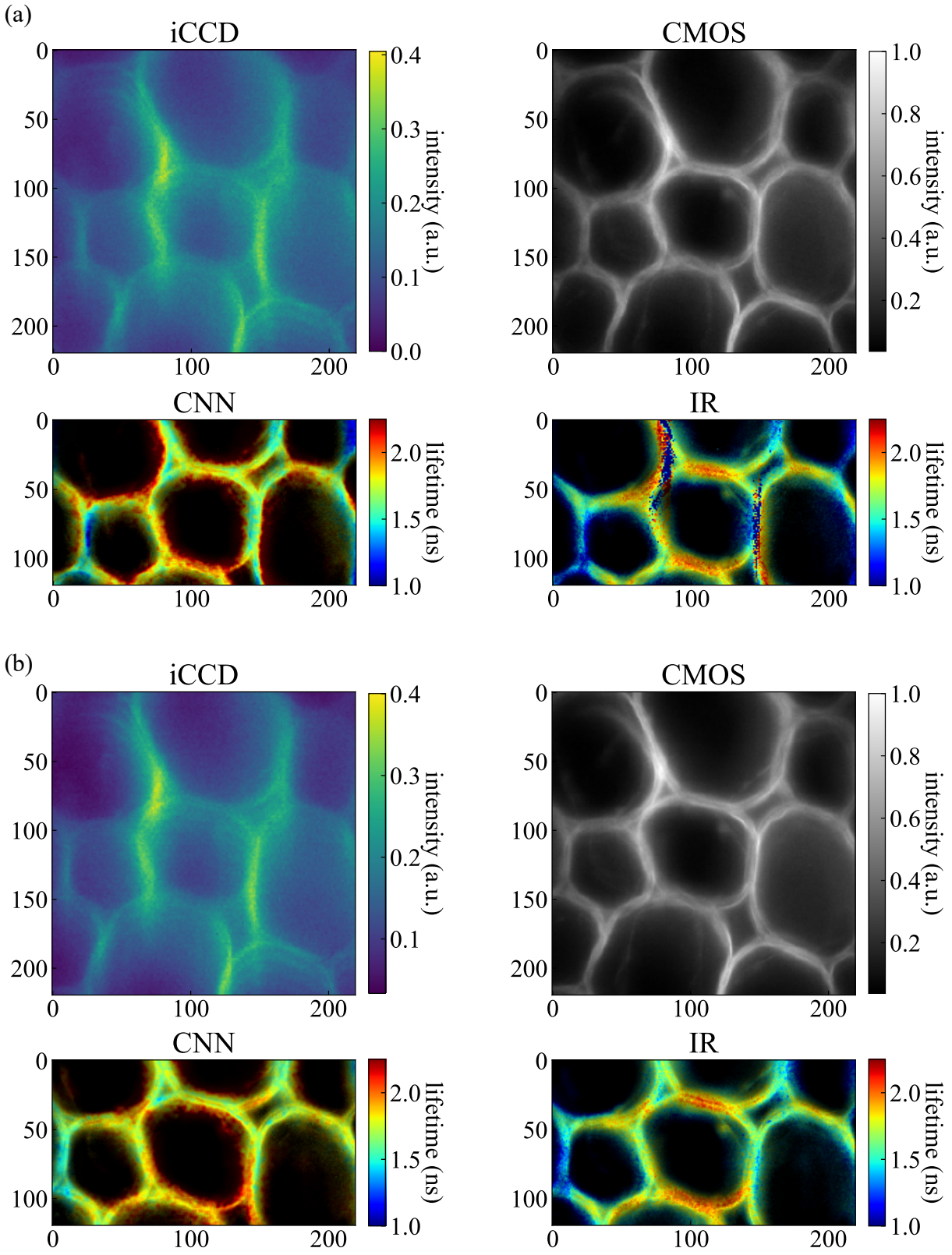

Figure 4.12: **(a)** Data with replica shear tilted $\sim 3°$ from vertical, and corresponding CNN and IR reconstructions. **(b)** The same sample with replicas sheared along the vertical axis, and corresponding reconstructions. Artefacts are suppressed.

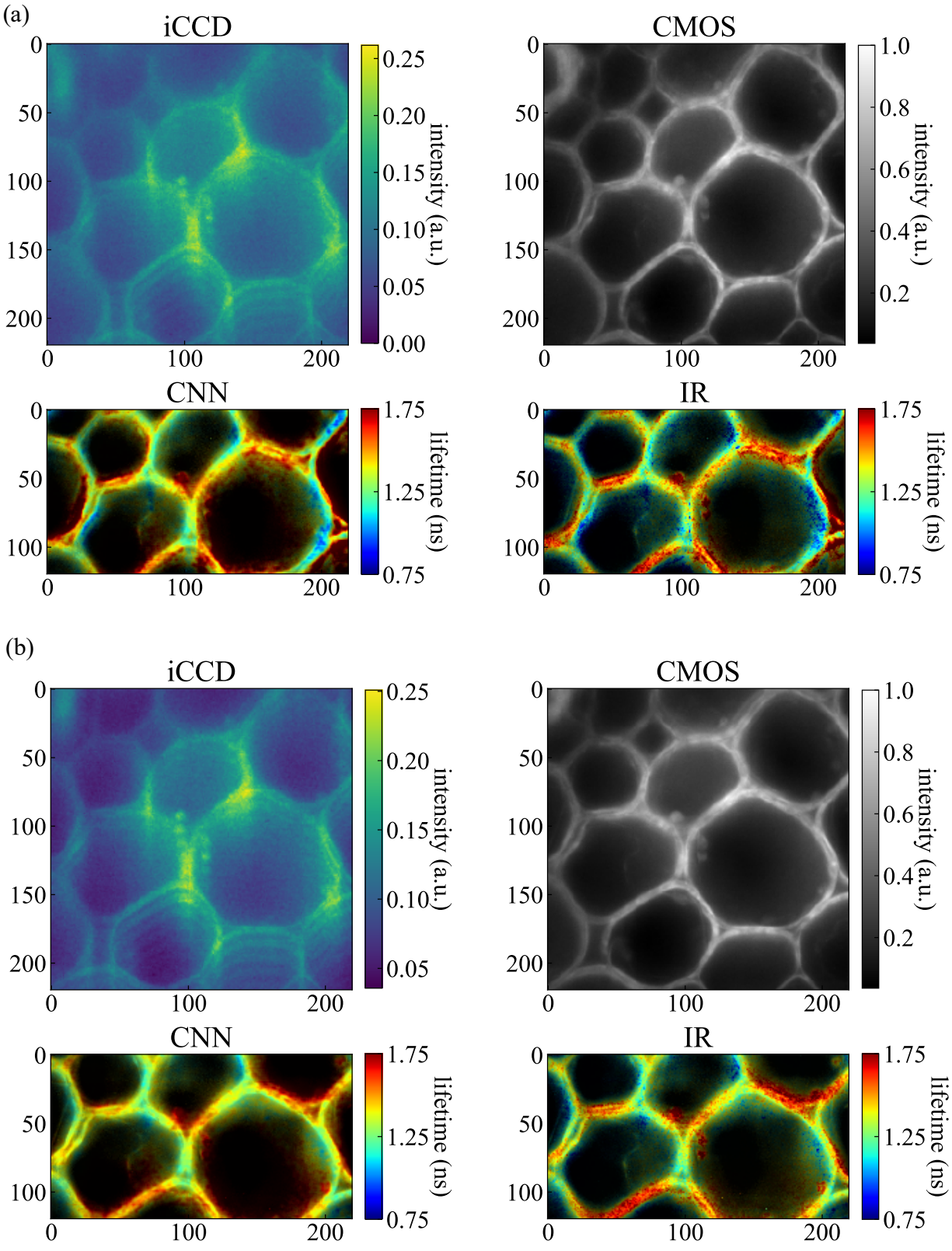

Figure 4.13: **(a)** Data with replicas tilted ∼ 5° from vertical, and corresponding CNN and IR lifetime estimates. **(b)** and corresponding CNN and IR lifetime estimates.

reconstruction, and observed that iCCD-only estimation of fluorescent bead lifetimes suffered compared to iCCD-CMOS data fusion estimation. We note that our iCCD-only results came from regular measurements made by our setup, therefore the comparison was not entirely fair. This is because the 30 : 70 beamsplitter was not removed for the iCCD-only test, so it unnecessarily directed 30% of the signal to the CMOS. Nonetheless, the estimation error increased so markedly, that we concluded that our estimators strongly rely on the pixel-wise ratio of the CMOS to the iCCD. For this two-camera scenario, Bayesian analysis derives the relative uncertainty of lifetime prediction for a wide range of lifetimes.

We validated our approach using fluorescent beads and tested it on cellular samples, obtaining lifetimes within the expected ranges. The simultaneous spatial and temporal shearing means that all of the information required for generating a wide-field image is obtained in a single measurement. This provides an advantage over scanning-based or line-streaking-based approaches.

On the other hand, it is experimentally difficult to create the optical cavity, as the adjustment of the mirrors has to be very accurate ($\sim \pm 0.1°$) over several degrees of freedom to get linear replicas. Follow-up studies may bypass this by using a custom pre-manufactured optical cavity. Furthermore, the SNR of individual measurements is decreased by the beamsplitter since the cavity merely redistributes fluorescent signals in time and space, it does not amplify them. The parallel acquisition of multiple distinct signal positions on the time-gate drops the SNR of each signal compared to what we would have without the cavity. Since fluorescence lifetime measurements are typically SNR limited (that is, the acquisition time needed for good SNR is usually long), the exposure time needed to acquire the same total fluorescent intensity (i.e. SNR) is the same with and without the cavity.

As the experimental cost and difficulty of setting up the cavity are not counterbalanced by improved SNR, optical cavity-based FLIM approaches are unlikely to become popular in the near future (in my opinion). Nonetheless, I do believe that the optical cavity approach would be advantageous for non-scannable time-gates.[30]

Some questions regarding our cavity-based approach remain open. For instance, it is unclear how signal sparsity affects the quality of reconstructions. Vertical artefacts in reconstructions indicate that severe overlap of the iCCD replicas hampers lifetime estimation, but the extent of this effect remains unknown, and it is not observed in synthetic data.

Lastly, since the cavity acquires multiple time-gated measurements, our method could allow single-shot multi-exponential retrieval, similar to multi-exponential RLD. I believe this would be the most promising avenue for a follow-up study.

[30] A non-scannable time-gate is one with a fixed temporal response or one which is difficult to alter electronically. In this case, the cavity allows for time-gate scanning without having to change the imaging path.

Chapter 5.

FLIM super-resolution: single-sample image fusion upsampling

Summary. In this chapter, we look at computational single-image super-resolution (SISR), a field of signal processing where we take an image and try to increase the number of pixels in the same field of view. Existing approaches typically use some form of interpolation, inverse retrieval (IR), or are example-based. Here, we report a novel IR-example-based hybrid algorithm dubbed single-sample image fusion upsampling (SiSIFUS). This method relies on data fusion to improve upon existing super-resolution techniques; we successfully apply it to FLIM data.

5.1 Computational super-resolution

Interpolation

Interpolation is the most basic form of SISR, assuming a simple forward model: the unknown high-resolution (HR) target is decimated (sampled at regular intervals) to give our low-resolution (LR) image. In other words, our measurements are sparse points on the high-resolution target, so the HR target can be estimated by connecting these LR points. This is called interpolation; we will look at a few interpolation methods, noting that they do not add any new information to our system, and therefore the choice of method is use-case dependent. Fig. 5.1 shows a summary of various interpolation methods for 2D data.

Nearest, bilinear, and bicubic. These are some of the most commonly used schemes. Nearest interpolation assigns each pixel to match the value of its nearest neighbour, with some policy for when two or more neighbours are equally far. Bilinear interpolation estimates HR pixels using a weighted sum of the nearest LR neighbours; if the HR point is horizontally between 2 LR pixels, i.e. it lies on a row of the LR image, then it is evaluated from the horizontal line connecting its LR neighbours. Similarly, if an HR pixel has vertical LR neighbours, the vertical line connecting them gives the HR estimate. If instead, the HR target lies between 4 LR pixels, then we first find the values of its nearest HR neighbours that lie on either a row or column and then interpolate between these. Bicubic interpolation works much the same, except we fit a cubic polynomial to 4 nearest measured pixels within a row or column.

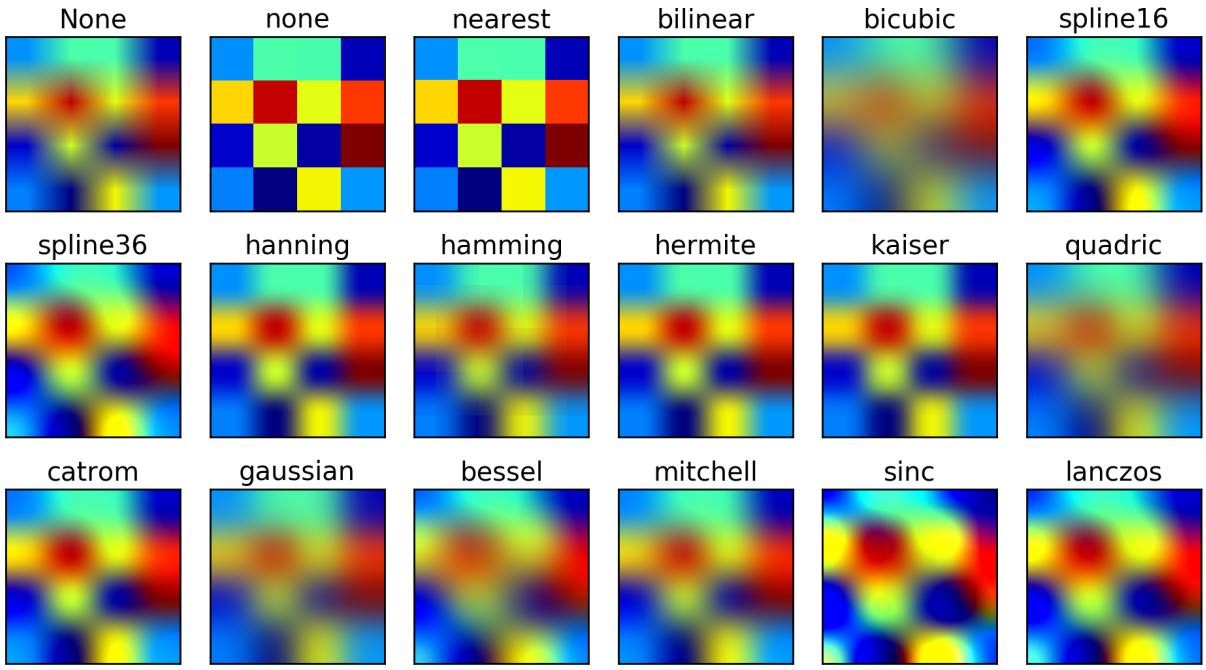

Figure 5.1: Schematic of various interpolation methods used by Matplotlib, connecting a 4×4 series of samples [101].

Spline. Spline interpolation is another commonly used technique, which ensures that the upsampled image has smooth gradients. Polynomials of some order are fitted to small subsets of points in the LR image. At the knots joining neighbouring subsets, the neighbours must have equal value, derivative and second-order derivative, ensuring the fit is continuous. The net collection of piecewise polynomials is called a spline.

Sinc interpolation. This well-known form of interpolation is derived from signal processing, using Nquist's theorem. The theorem states that a function with bandlimit B can be completely recovered from samples at 1/(2B) intervals. So, if we sample at an interval larger than 1/(2B), we are undersampling, but if our detector obeys the Nyquist criterion, we can recover the high-resolution signal losslessly.

We treat our samples $y_i = f(t_i)$ as a Dirac comb at positions t_i, scaled with amplitude y_i. We then convolve with sinc(t/T), where T is the sampling interval. The output is a sum of sinc functions, each at the position of one LR sample and scaled with the corresponding intensity. Equivalently, we can take the Fourier transform of our image, zero-pad it to the desired HR size, and perform an inverse Fourier transform.

Unlike electrical signals in wires, where we can control the bandlimit of sent messages, images have an ill-defined, sample-dependent bandlimit (on the spatial frequencies in the object plane). In microscopy, this is often higher than the imaging system's modular transfer function

(MTF), upper bounded by the diffraction limit.[1] In such cases, the image is considered under-resolved.

Lanczos interpolation. This is closely related to sinc interpolation, but instead, it uses a finite-sized, scaled sinc function to counteract the infinite support of sinc(t/T) over t. Lanczos interpolation is therefore a numerical improvement, as we cannot directly convolve our LR image with an infinitely long function.

Inverse retrieval

Inverse retrieval methods[2] guess the high-resolution image by simultaneously enforcing reconstruction constraint(s) and prior(s). The reconstruction constraint defines that the detector D imaging the HR target x must give to the LR measurement y. D is the net downsampling operator performed by the imaging system, typically a cascade of decimation/binning/blurring. The inverse model is typically ill-posed, meaning any one of an infinite number of HR targets could be downsampled to give y. Priors $P(x)$ regularise this task, allowing us to select the most likely HR option. The super-resolution task will then be:

$$\hat{x} = \arg\min_{x}\{||\mathrm{y} - Dx||_2 + \alpha P(x)\} \tag{5.1}$$

for some weighting factor α.

Some IR-based approaches focus on optimising the downsampling model. One might spread high-resolution information onto a smaller number of pixels using an engineered point spread function, enabling SISR from low pixel-count sensor data [102]. Similarly, defocusing the HR sample spreads information onto the sensing pixels [27]. Blurring the image plane onto a wide area of the detector plane increases the receptive field of each detector pixel. This means that even a sparse camera (i.e. one with significant dead space between pixel active areas) can collect some information from the regions between pixels.

Other IR-based approaches have a larger focus on the prior. Implementations sometimes enforce priors on the gradients in the image, encouraging characteristic edge shapes profiles [103], or by facilitating that the image should be sparse in gradient space [104]. An interesting

[1] MTF is the spatial frequency response of an imaging system. It compares the spatial frequencies seen by the detector to the spatial frequencies in the original sample. The MTF is typically high for low frequencies and falls off for high frequencies. For example, in images of natural scenes, large features like the shape of a mountain, whereas small details like individual grass blades on the mountain are blurred. The later the fall-off point, the better the camera resolution.

[2] In literature, super-resolution via inverse retrieval is often called reconstruction-based super-resolution.

prior constraint is that of low-rank image approximation. This posits that if one divides a natural image into small patches, many patches are similar, and others are linear combinations of one another. Consequently, singular value decomposition (SVD) of the patch matrix yields few SVs with large coefficients, so this matrix is low-rank and the image is compressible. If the image is noisy, however, the image patches become more dissimilar, so the patch matrix has a higher rank. Hence, reconstructions are encouraged to be compressible (like noise-free natural images) using low-rank priors [105].

Compressed sensing provides a framework for reconstructing HR data from far LR measurements through numerical optimisation. CS focuses on both the downsampling operator and the prior; refer to Chapter 2 Sec. 2.3.2 for an introduction to CS. Fig. 5.2 shows some examples of the impact of the choice of measurement matrix and sparsity basis on the reconstruction quality

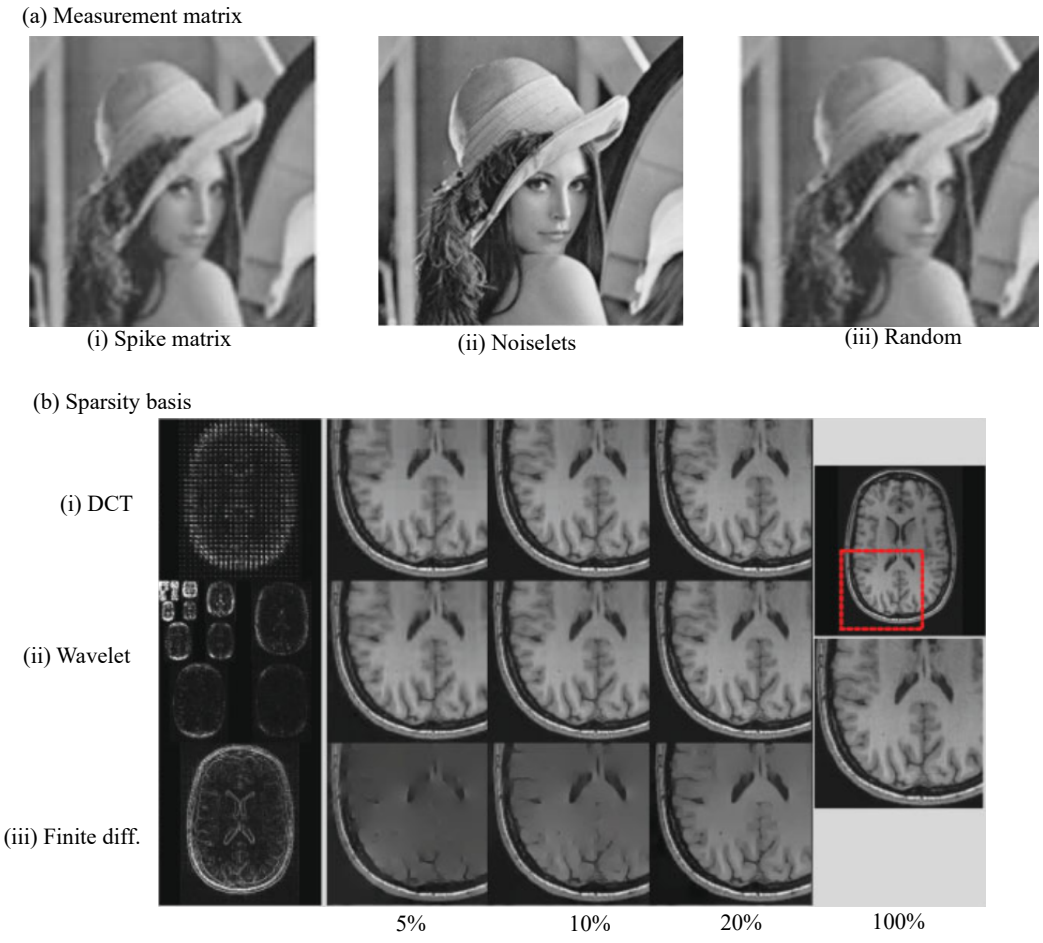

Figure 5.2: **(a)** Extreme example showing the impact of the measurement matrix on compressed-sensing super-resolution of an image, from [106]. **(i)** A spike matrix was used as the measurement matrix, **(ii)** uses a noiselet matrix, **(iii)** a random matrix. **(b)** Impact of the sparsity basis on MRI image reconstruction. A brain image is shown in **(i)** DCT, **(ii)** wavelet, and **(iii)** finite difference (i.e. gradient) basis, alongside reconstructions from various percentages of the largest components in these bases.

of images.[3]

CS assumes the data is sparse in some basis and that there is incoherence between the downsampling matrix (measurement) and the sparsity basis (prior)[107]. Incoherent measurements can be achieved in several ways, ranging from sampling the scene in the frequency domain (as in MRI [108]), sampling random DMD projections of the image plane (Hadamard basis)[109, 110], or using random diffusers to replace a lens in an imaging setup [111].

Such schemes are ideal when the sample is sparse and we are interested in recovering small features that might fall between detector pixels. However, the overall light intensity collected by such a camera does not increase compared to a regular sparse measurement. If the same amount of light is used to make measurements of more pixels, the pixel-wise SNR may decrease. Moreover, many methods proposed in SISR literature introduce complexity in both the optical setup and the processing algorithm. Even if the experimental and processing pipelines are in place, CS reconstruction programs often still consume significant time and electricity.

Example based

Example-based[4] SISR methods "hallucinate" high-resolution details from a training set of HR/LR image pairs. I differentiate between example-based methods that use classical inverse retrieval and deep learning; this subsection looks at classical methods.

Neighbour embedding. This field of super-resolution uses a set of high-resolution training images to upsample the measured low-resolution image. The training images are downsampled to get a corresponding low-resolution image set; we then split the HR and LR sets into patches. Ultimately, these HR-LR patch pairs form our training dictionary; the considerations for making such a dictionary were established in [112].

We then divide the measured low-resolution image into patches. For each patch, we find the most similar training LR patches; these are its nearest neighbours, giving the method its name. The simplest similarity metric is the Euclidean distance of pixel values, but most algorithms use feature-similarity, e.g. the root-mean-squared error between the patch gradients [113]. These nearest neighbour training LR patches all have corresponding HR patches, which are linearly combined to upsample the target patch. Fig. 5.3 illustrates this scheme. The weights of this linear combination (weighted sum) should minimise reconstruction error, i.e. the downsampled linear combination should give the measured LR patch.

[3] Across different target images or applications, results will vary, as measurement matrices and sparsity bases are better suited for some data than others

[4] Example-based methods are sometimes called learning-based, but to avoid confusion with machine learning/deep learning, I refer to them as example-based.

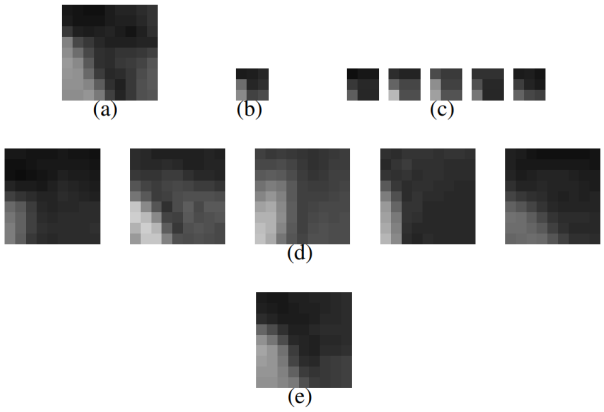

Figure 5.3: Neighbour embedding SR, adapted from [113]. **(a)** Shows the target (ground truth) HR patch. **(b)** Shows the LR patch. **(c)** Shows 5 nearest neighbour LR patches in the training set; whilst **(d)** shows the corresponding 5 nearest neighbour HR patches. **(e)** The prediction patch is a linear combination of the 5 HR patches.

Sparse coding. Neighbour embedding suffers from a globally fixed number of neighbours. Like NE, sparse coding (SC) methods minimise the difference between a measured LR patch and a dictionary of LR training patches, typically by minimising distance in some feature space such as gradient space. Unlike in NE, we jointly optimise a sparsity term (L1-norm of the weights of the neighbour patches) alongside this feature distance term. The joint optimisation allows the algorithm to choose how many training patches to combine in the upsampling process (instead of limiting the search to the 'N' most similar patches) while still encouraging the number of patches to stay small.

More advanced implementations operate on overlapping patches to avoid edge effects. Overlapping regions are upsampled by jointly optimising the corresponding patches. The choice of patches in the training dictionary strongly affects reconstruction. The fewer elements there are in the dictionary, the faster the upsampling process, but the dictionary must be diverse

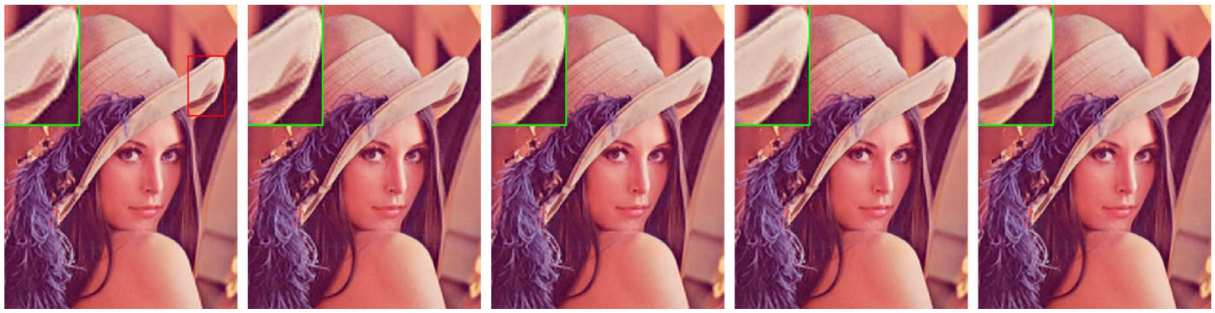

Figure 5.4: Impact of [trained] total dictionary size on sparse coding SR of Lena, from [114]. From left to right, the first four images use 256, 512, 1024, and 2048 trained patches. The last one uses 100,000 untrained patches, from which the training process selected the prior mentioned subsets.

enough to match our LR images. 'Training' the patch dictionary helps achieve these conflicting objectives[114, 115].

Anchored neighbourhood regression. ANR is a method for speeding up SISR. Like NE and SC, it is an example-based method that uses a dictionary of HR and LR patches for upsampling. Searching through an extensive dictionary is time-consuming; moreover, much of the dictionary may be redundant or unused for a given set of images. It makes sense to instead offload time-consuming parts of the algorithm to offline pre-processing, and only solve a simpler regression task online [116]. Briefly: for each dictionary element, a neighbourhood of similar patches is computed, much like in NE, offline. For the actual online upsampling step, the LR image is divided into LR patches (like in NE and SC), each of which is matched to the most similar dictionary patch (called the anchor). Then, a single SC-based step is performed between the LR patch and the anchor's pre-computer neighbourhood.

The key weakness of example-based SISR is that vastly different training HR patches can correspond to identical LR patches, causing ambiguity in the reconstruction process. The greater the downsampling factor of the HR patches (and equivalently, the greater the SR upsampling factor), the worse the ambiguity. Accordingly, a hallucinated patch is not guaranteed to give contextually sensible high-resolution details.

Self-similarity. Self-similarity approaches seek to alleviate this problem by finding example patches in the given image itself. Self-similarity is a type of SISR where we look for similar features within the image itself, instead of using an external dictionary. Although this does not guarantee that hallucinated features are correct, and it increases the computation time of online processing, it does ensure that HR features are from inside the sample's patch distribution.

Fig. 5.5(a) illustrates the concept of self-similarity - we see repeating features at different scales in the image. Say we want to super-resolve the small blue boxed bush in the measured LR image (top-left). We can shrink the image and look for bushes of the same size as our target in the smaller images; note that we have higher resolution versions of these bushes in the original LR measurement.[5] We upsample the target bush with these HR versions. We can use a similar process to upsample the window in red using the similarly shaped door.

Fig. 5.5(b) shows the workflow. We start with an LR measurement I_0. If we want to super-resolve some small target patches (dark green, dark red), we sequentially look for reference patches with similar features in downsampled versions of the LR image ($I_{-1}, I_{-2}, I_{-3}, ...$). These correspond to larger reference patches in the original LR image (light green, light red). The large patches replace the small targets, but these reference patches have variable sizes depending on which downsampling level they originate from. So, the desired HR patch size is reached with

[5] Since the image has been downsampled, bushes near the camera now match the size of the small target bush.

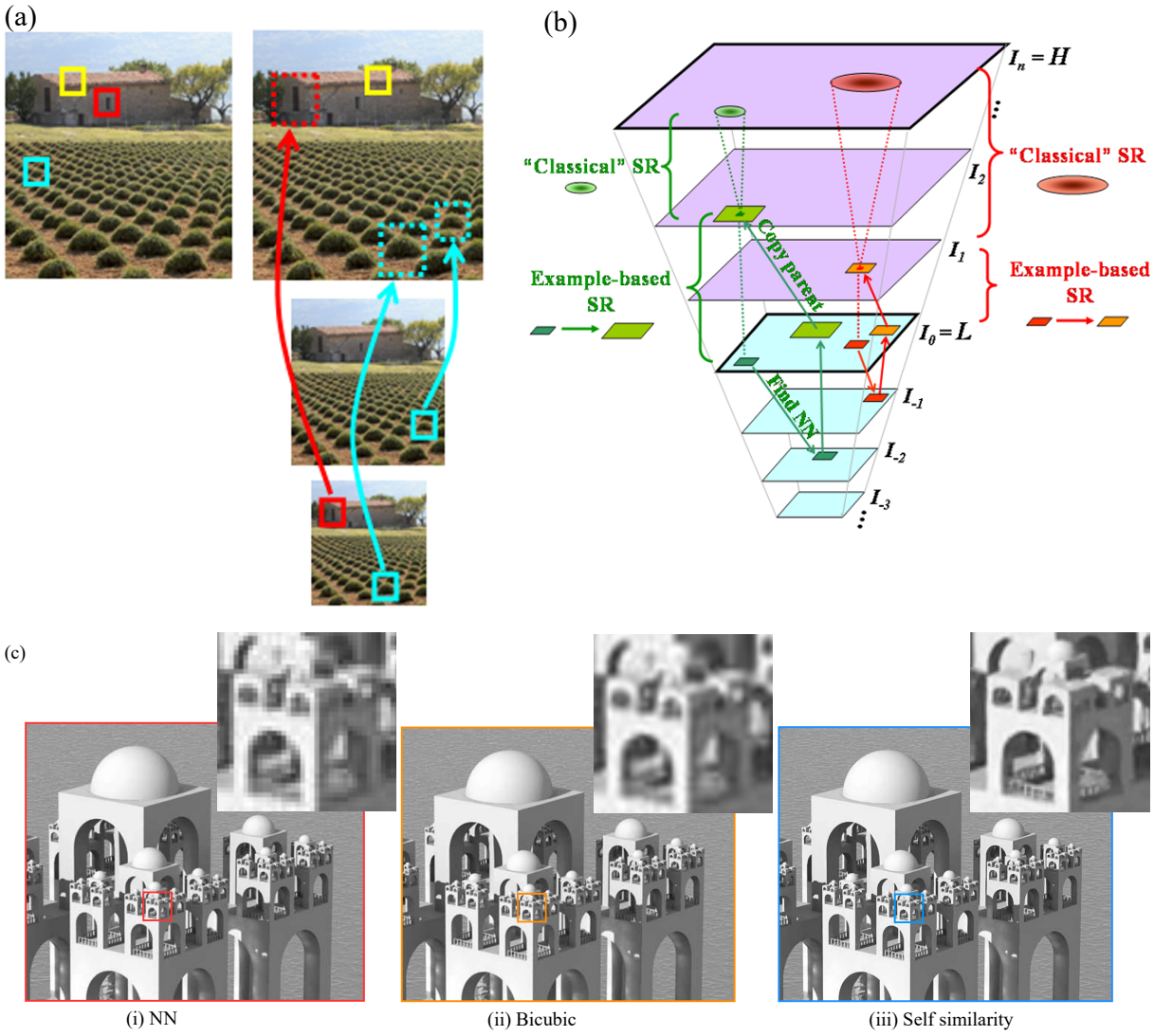

Figure 5.5: Overview of self-similarity based SR; images from [117]. **(a)** Illustration of similar features in the same image across different scales. **(b)** Schematic of the self-similarity SR workflow, where we go from I_0 to I_n. **(c)** Benchmarking of self-similarity SR to nearest neighbour and bicubic interpolation.

interpolation/IR-based SR. Fig. 5.5(c) shows an extreme example of self-similarity SR in action on a fractal-ish image.

Deep learning super-resolution.

The field of machine learning has uprooted image processing in the past decade. In particular, deep learning (DL) based super-resolution is witnessing explosive improvements year-to-year. DL-SR algorithms can be classified in many ways. However, to minimise the 'black box' nature of this review, I will approach DL-SR by drawing analogies with classical SR. See [118] for a

more in-depth review, and [119] for a review of real-world SR, focusing on overcoming real-life challenges, which is relevant to experimental optics.

CNN. The basis of most DL-SR is the convolutional neural network (CNN)[120], popularised for this purpose by SRCNN [5]. The authors demonstrated that a CNN trained end-to-end on an SR task (low-resolution image input, high-resolution image target) performs patch extraction, representation and non-linear upsampling, as shown in Fig. 5.6(a). The authors also observe that the convolutional kernels learned by a CNN trained on a broad dataset form an efficient upsampling set similar to a trained patch dictionary, with individual filters optimised for given tasks - see Fig. 5.6(b), where filters (a-e) resemble traditional edge detectors, (f) resemble a texture extractor, and (g-h) are like Laplcians/Gaussians. The authors further demonstrate that an SRCNN is analogous to a sparse coding scheme.

Supervised. Based on the CNN backbone, a series of supervised architectures have been developed that learn increasingly complex modelling of SR. These architectures range from very deep CNNs [121], through residual CNNs [122], to residual autoencoders (i.e. U-Nets) [123]. Often, the LR image is first interpolated (e.g. via bicubic interpolation), and only the high-frequency residual is computed by the SR algorithm; the final image is reconstructed by adding the residual

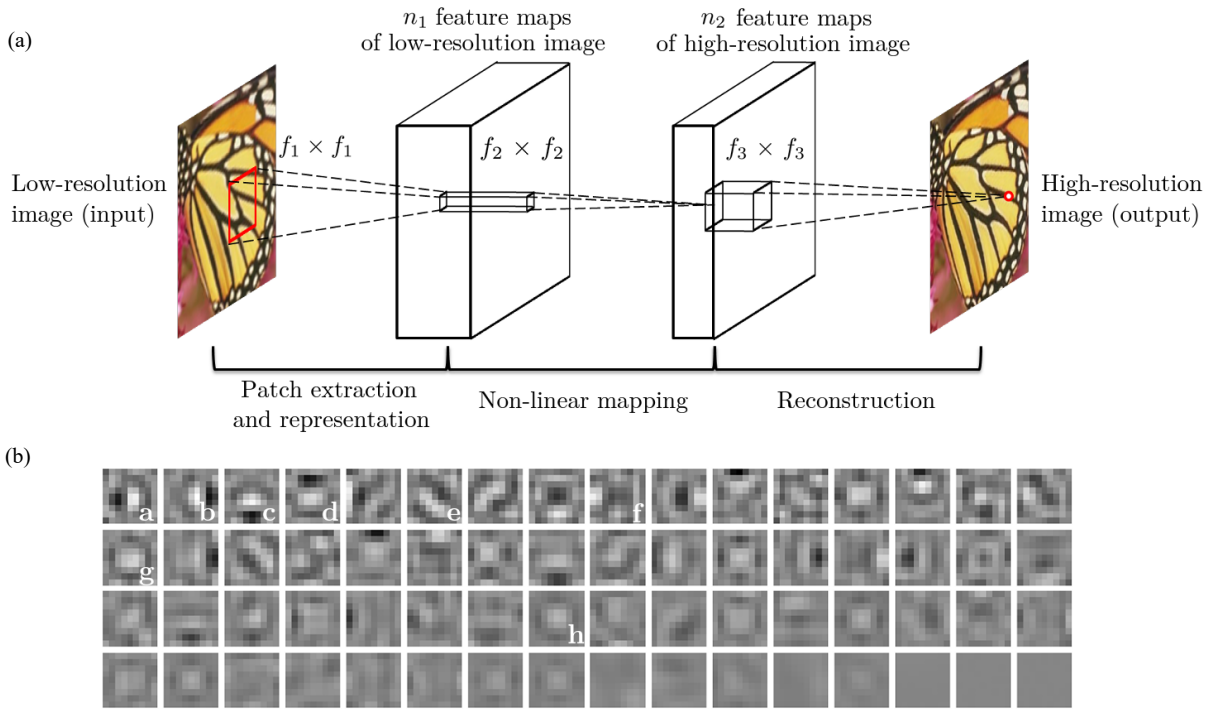

Figure 5.6: SRCNN algorithm and convolutional kernels, from [5]. **(a)** Schematic of the SRCNN algorithm, drawing analogies to classical example-based SR via patches. **(b)** The kernels of the first layer, shown organised by variance. We see various filters that match traditional feature extractors, such as edge detectors at various orientations.

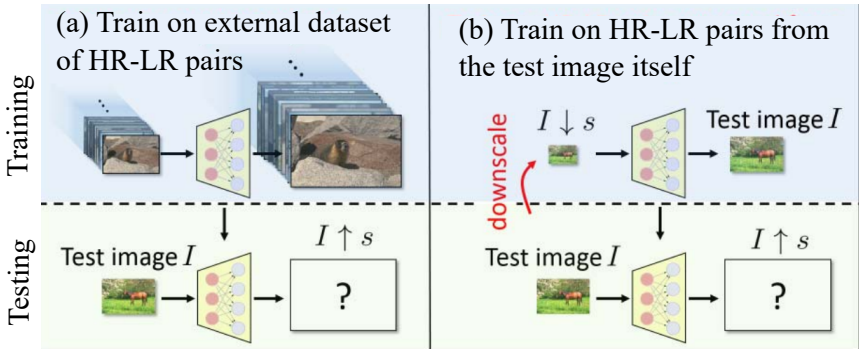

Figure 5.7: Image from [125]. (a) Shows the supervised DL-SR pipeline, where a neural network is trained on an external dataset of LR-HR pairs. (b) Shows the unsupervised (self-supervised) pipeline, where an image is downsampled and trained to upsample back to itself. In testing, the same convolutional kernels are used, allowing the network to adapt to the larger size of the input than in training.

to the interpolated image [6]. This lifts the requirement for the SR algorithm to keep track of the mean, low-frequency features and allows it to focus on HR features only. Another popular mechanism in DL-SR is channel attention, which models inter-dependencies of feature maps within a given layer [124].

Self-supervised. A series of self-supervised algorithms have also been developed for SR. Zero-shot super-resolution (ZSSR) demonstrates this well [125]. Fig. 5.7 compares supervised SR to self-supervised algorithms like ZSSR.[7]. Self-supervised algorithms are essentially deep learning analogues of self-similarity-based classical SR; a neural network learns to map a downscaled version of the LR input back to its original size with a set of convolutional kernels. These same kernels are applied to the image at its original size, super-resolving it.

common side-task of SR is degradation modelling, where we assume some unknown degradation of the HR image, typically blurring, happens before decimation [126]. The task is then to estimate the degradation kernel. Degradation modelling is typically treated separately from the SR task; once found, the degradation kernel is plugged into some existing SR pipeline. In the presence of HR-LR image pairs, this task is a well-studied inverse retrieval problem [103, 127], however, if we only see the degraded LR image, it is less straight-forward to learn the degradation kernel in an unsupervised manner. Several solutions to blind degradation modelling have been proposed with ML, including generative models like GANs [128, 129].

Generative modelling. In recent years, generative models have completely transformed SISR. Generative modelling is a field of machine learning that focuses on learning the distribution of a training dataset. This allows us to generate new, synthetic images of a similar distri-

[6] The residual is the difference between the interpolated image and the target HR image.
[7] We note that despite the figure seemingly depicting a dense neural network, a CNN is used

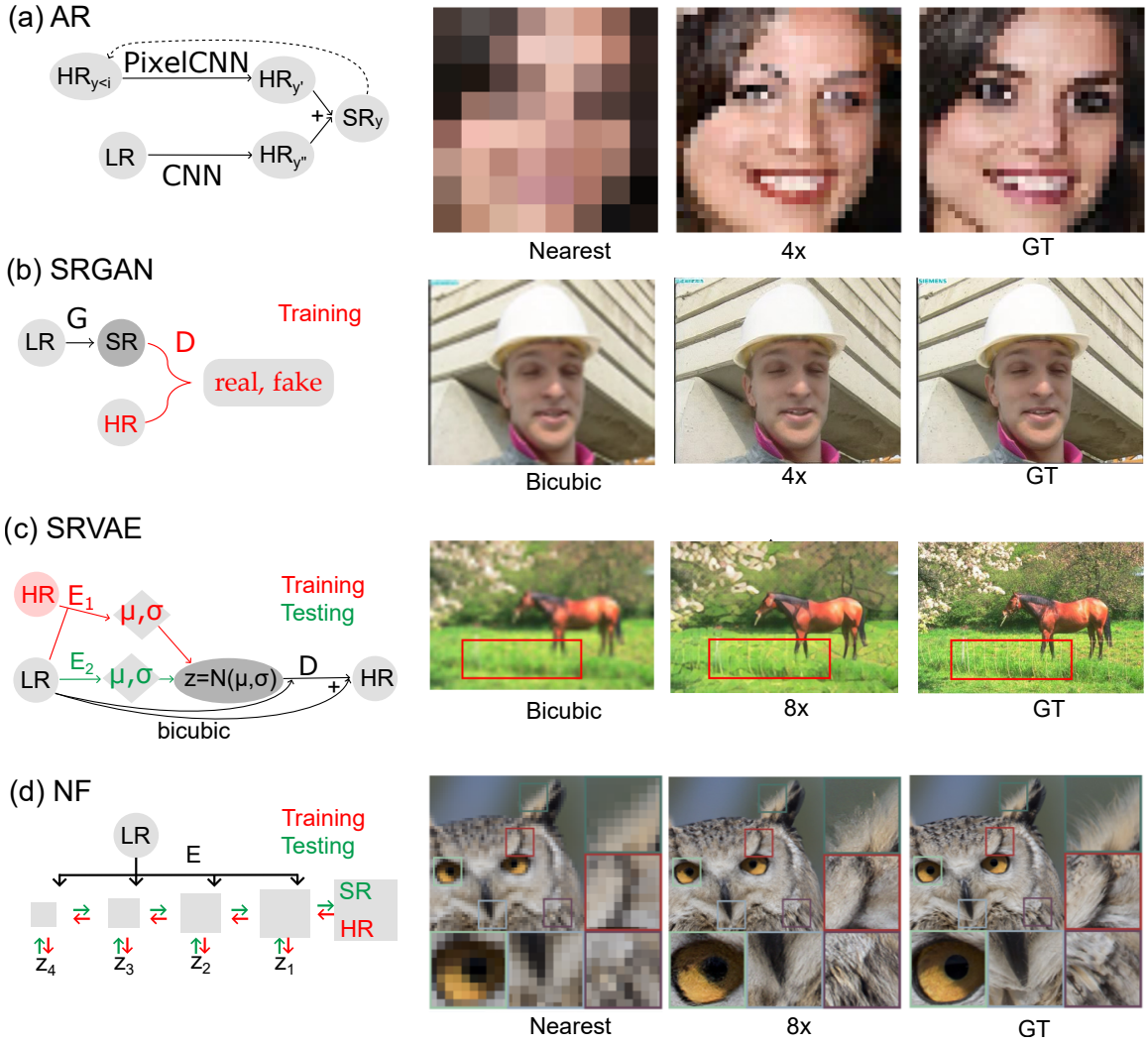

Figure 5.8: Types of generative SR. **(a)** Autoregressive SR, from [131]. **(b)** SRGAN, from [6]. **(c)** SRVAE, from [132]. **(d)** SRFlow, from [133].

bution as our training set. A common application is generating fake images of human faces. See [130] for a review of generative modelling.

The core idea of generative modelling is that we want to sample a distribution of HR images $x \sim \hat{p}(\hat{x})$, which estimates the distribution of our training samples $x \sim p(x)$, such that $\hat{p}(\hat{x}) \approx p(x)$. These samples should not be random but conditioned on the LR image. There are several approaches to achieve this; we will briefly consider the most popular ones used in SISR.

Autoregressive models. Autoregressive models explicitly track the interdependence of pixel values, predicting each pixel sequentially based on the previous ones. This relies on the chain rule of probability, which allows one to calculate the probability of an image $\mathbf{x}$ made of pixels x_i

based on the joint probability density of the pixels, using only conditional probabilities:

$$p(\mathbf{x} \in \mathbb{R}^n) = \prod_{i=1}^{n} p(x_i | x_{i-1}, ..., x_1)$$

$$\left[\text{e.g. } p(\mathbf{x} \in \mathbb{R}^4) = p(x_4, x_3, x_2, x_1) = p(x_4|x_3, x_2, x_1) \cdot p(x_3|x_2, x_1) \cdot p(x_2|x_1) \cdot p(x_1)\right]$$

$$\Rightarrow \; -\ln p(\mathbf{x}) = -\sum_{i}^{n} \ln p(x_i | x_{i-1}, ..., x_1) \tag{5.2}$$

The negative log-likelihood can be minimised directly to capture the joint probability distribution of the data. However, the image generation process is slow, as pixels are generated in sequence. Moreover, the ordering of the pixel sequence will impact the output, yet there is no *a priori* obvious best choice.

A famous example of autoregressive image generation is PixelCNN [134]. PixelCNN uses convolutions to parallelise modelling pixel interdependence, speeding up the AR training process. In SISR, AR models can super-resolve small images where the slowness of sequential prediction is less of a setback. An example is shown in Fig. 5.8(a), from [131]. The authors use a PixelCNN (initialised with the first pixel of the HR image, which is also the first pixel of the LR measurement) to predict the second pixel. Meanwhile, a regular CNN upsamples the LR measurement, containing a different guess for the second pixel. The super-resolved estimate for the second pixel is the sum of the two guesses. This estimate is fed back to the PixelCNN, outputting the third pixel; the regular CNN's output for the third pixel is added, giving the super-resolved third pixel. This process is repeated, creating the full image.

Generative adversarial networks. Generative adversarial networks (GANs) create photorealistic images by training to fool a discriminator network, whose job is to discern fake images from the real images in the training set. A GAN is made of two networks, a generator and a discriminator. In the forward pass (the training evaluation step), the generator draws random samples from some noise distribution, processing these to generate SR images. Then, the discriminator takes a mixed batch of generated and ground truth images and labels each with a 0s and 1s, respectively.

The backward pass (the training update) takes two steps. First, we update the discriminator's weights to classify training images as 1s and generated images as 0s. In the second step, we freeze the discriminator (fix its weights) and update the generator weights instead, aiming to produce images that the discriminator classifies as 1 (i.e. to trick the discriminator into thinking the generated images are real). By repeating these two steps, the discriminator becomes better and better at modelling from the training set distribution and determining whether or not an image comes from this distribution. Meanwhile, the generator improves at creating images that match the training distribution better and better.

Fig. 5.8(b) demonstrates the super-resolution GAN (SRGAN) approach, popularised by Ledig et al [6]. The generator is conditioned on the LR image instead of noise, outputting an SR estimate. Training is done in two alternating steps. First, the discriminator is trained to classify SR images as 'false' and HR images as 'real'. Then, the frozen discriminator acts as a loss function for the generator, which tries to fool the discriminator. The generator loss also has an image similarity component (e.g. MSE) to recover the HR target instead of random HR samples. Most SISR GAN approaches are deterministic: the generator is some CNN-based model that outputs a given output for a given input.[8]

Variational autoencoders. Variational autoencoders encode (compress) an image, producing a compressed vector of means and standard deviations. These means and standard deviations parameterise a normal distribution. This image-specific Gaussian is sampled, giving a low-dimensional, probabilistic embedding. From this, the decoder reconstructs the HR image. We encourage the latent space of the full training dataset to approximate some prior, typically a unitary Gaussian, with sets of similar training images populating subspaces of this Gaussian. Consequently, unlike a regular autoencoder, the VAE has a continuous latent space, whose decoded samples vary smoothly in feature space. For details, see [135].

Multiple VAE approaches exist for SISR. Here, we focus on the implementation in [132]. Similarly to many classical SR algorithms, this VAE focuses on estimating high-frequency residuals of interpolation (i.e. the difference between the bicubic interpolated LR image and the HR target). See Fig. 5.8(c) for an overview of this method. During training, the HR and LR images are encoded together onto latent features. We input these latent features together with the LR image into the decoder, which predicts the high-frequency residuals of the target. These are added to the bicubic upsampled LR image to give the final SR estimate. A separate encoder, E_2, is trained to approximate the same distribution as E_1. eThis allows us to bypass the E_1 branch in testing.

Normalising flows. Normalising flows (NFs) find mappings from Gaussian noise and the distribution of training images. The training step maps training data, sampled from the complex image distribution, onto a normal distribution through invertible functions; in testing, we can then sample the Gaussian and run it through the inverse function to obtain new complex samples. For more details, see [136].

SRFlow was the first NF implementation that achieved an advantage over GAN-based methods [133]; its outline is shown in Fig. 5.8(d). The authors decompose the SR task into multiple 2×2 upsampling sub-steps; to make these invertible, the number of feature channels is decreased by $4x$ at each upsampling step. An encoder E is used to deterministically super-resolve the LR input to various sizes. The encoder can be any differentiable architecture; it need not be invert-

[8] It is just trained with an adaptive, dataset-specific, perceptual loss function (the discriminator).

ible, generative, or probabilistic. In the training phase, we split the HR images into the encoded images and variational feature maps, which transform into Gaussian noise through invertible steps. Then in the testing phase, we can take various random noise samples, get the corresponding variational feature maps, and add these to the deterministic encodings, creating probabilistic SR images.

Diffusion models. Diffusion models have become very popular in recent years. Similarly to NF models, they learn cascades of invertible noising/denoising operations; however, these are not simple analytically invertible functions but autoencoders, vastly increasing the capacity of diffusion models to capture image distributions. Google's SR3 is a very impressive implementation of diffusion-model-based SISR [137]. During the bulk of our super-resolution work, diffusion models were still in their infancy, so we did not consider them for our application.

Generative modelling for FLIM SISR. As shown above, generative modelling is a powerful tool for creating realistic-looking SR images. I experimented with generative modelling for our data-fusion-based super-resolution task, specifically GANs and VAEs. However, such models seek to capture the distribution of images and therefore require large amounts of data. We realised that our FLIM dataset was comparatively limited in volume (900 FLIM images), diversity (3 probes, 2 of which have constant lifetimes) and SNR.[9] These issues prompted us to venture away from data-hungry methods like generative modelling, and instead focus on approaches that extract information from a single sample.

Single-sample image fusion upsampling - our algorithm.

There are some unique challenges associated with super-resolving FLIM data compared to e.g. natural images:

1. Lifetime estimation uncertainty makes ground truth images noisy.
2. Different fluorophores produce different lifetime distributions.
3. Statistical studies of FLIM image properties are limited.
4. There is a lack of large open-source datasets; difficult to mine data online.

[9]The probe with variable lifetime was a FLIM-FRET probe, Rac1-Raichu (see Fig. 2.3). Our data showed lifetime variance, whose spatial distribution did not match our expectations, and seemed quite random and speckly. Furthermore, we expected a long-lifetime population (non-FRET lifetime) and a distinct short-lifetime population, but we usually observed single population modes. This suggests the SNR of our signal may have been poor, so lifetime estimation uncertainty may have exceeded the difference between the populations, removing contrast. Another possibility is that the probe was globally on or off, and we were actually measuring a single, noisy population.

5. Few fluorophores produce a bright (high SNR) emission, that also exhibits lifetime variation; hence it is difficult to create a diverse FLIM dataset.

6. It is difficult for an end-user to judge FLIM SISR quality in absence of HR ground truth, whereas we can estimate reconstruction quality of e.g. human faces.

In this paper, we propose a learning-based method of upsampling fluorescence lifetime images, which we call single-sample image fusion upsampling. Our method combines ideas of self-similarity-based patch learning with data fusion. To our knowledge, the SR algorithm described here is novel.

Image quality metrics

A central aspect of super-resolution is the ability to judge the quality of an upsampled image. This is done by comparing the HR reconstruction to an HR ground truth measurement. We discuss some image similarity metrics, ranging from pixel-wise comparison, through comparing low-level features such as edges in small patches of the image, to differentiating images based on high-level, contextual features.

Pixelwise metrics (esp. PSNR). Pixelwise metrics evaluate the difference between two images on a per-pixel basis. Common examples are mean absolute error (MAE) and mean squared error (MSE). Peak signal-to-noise ratio (PSNR) is commonly used in SR, as it scales with both MSE and the dynamic range of an image. It thus generalises MSE over images of different intensities, or over cameras that discretise light at different levels.

These metrics are straightforward, but they reflect human perception poorly. They infamously under-penalise image blur, which produces low pixel-wise error whilst deteriorating perceptual quality significantly. Conversely, they are sensitive to transformations like single-pixel translations or gentle image warping, which the human eye can hardly detect and which does not significantly affect our perception.

Structural similarity index measure (SSIM). A commonly used metric that approximates human vision is the structural similarity index measure (SSIM) [138]. SSIM is calculated on two windows, x and y, colocated in two images. It measures the luminance l, contrast c, and structure s in that window:

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c1}$$
$$c(x, y) = \frac{2\sigma_x\sigma_y + c_1}{\sigma_x^2 + \sigma_y^2 + c1}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x \sigma_y + c3}$$

$$SSIM(x,y) = l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma}$$

where μ denotes mean pixel value, σ^2 denotes variance ($\sigma_x^2 y$ mean covariance), and c1, c2 are stabilisation variables; α, β, γ are weighting factors. Luminance measures the difference in mean intensity between the windows, contrast measures the difference in variance between the windows, and structure measures how strongly the pixels in the two windows co-vary. The window size is typically a sliding 11×11 square weighted by a Gaussian of standard deviation 1.5, as proposed by the original authors of the metric [139].

SSIM varies between -1 and 1; 1 indicates a perfect match between the windows, 0 means no similarity between them, and -1 signifies that the pixel values are perfectly anti-correlated. The net SSIM of the image is the mean of the SSIMs of the various windows.

There are several offshoots of SSIM. A commonly used variant is multiscale SSIM, in which the image is iteratively low-pass filtered (blurred) and decimated (sparsely downsampled) by a factor of 2 at each iteration, and SSIM calculated at each of these scales [140]. A less widely used variant is multicomponent SSIM, which decomposes an image into components (edges, texture regions, smooth regions), computes SSIM on each component, and gives a combined SSIM by weighting these component-SSIMs [141].

Whilst SSIM is fast, tractable, and generally outperforms pixel-wise metrics, it looks at very low-level features of image patches, and still considers pixel-wise covariance to calculate structure.

Learned perceptual image patch similarity (LPIPS).

A more sophisticated image quality metric is learned perceptual image patch similarity (LPIPS)

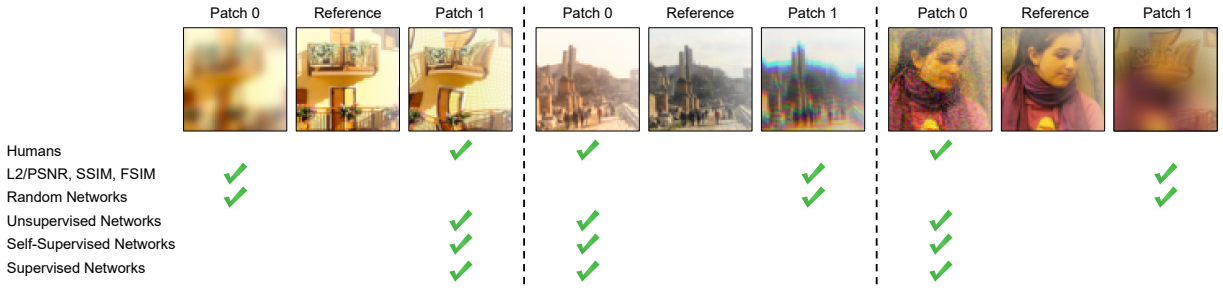

Figure 5.9: Examples of metrics comparing images to their corrupted counterparts; green ticks show which corrupted image is rated as more similar to the reference. Image from [142].

[142]. Image-processing neural networks trained on broad image datasets perform generic feature extraction in their earlier layers. This can be exploited by taking a pre-trained neural network and propagating our images of interest through the first couple of layers, giving low-level feature maps of these images, and computing the distance between these feature maps. This idea is similar to SSIM, except we look at a lot more features than just luminance, contrast and structure as defined above.

Fig. 5.9 demonstrates the difference between how pixel-wise metrics and low-level metrics like SSIM evaluate image similarity, versus how humans and deep neural networks evaluate image similarity. LPIPS is designed based on the observation that DNNs create similar abstractions of image similarity as we humans do. It uses a convolutional neural network, typically AlexNet, trained in a supervised manner on the ImageNet dataset for image classification. The network is frozen and images of interest are input into it. The outputs of a pre-defined set of layers are extracted, forming feature maps. We then compute the cosine distance between these feature maps, this is the LPIPS score. A lower score indicates better perceptual similarity, with 0 indicating a perfect match. LPIPS has offshoots, e.g. we can enforce distinct inputs to map to distinct features (injectivity), giving Deep Image Structure and Texture Similarity (DISTS) [143].

Whilst feature space measures can offer advantages over pixel-wise metrics, they can incur a loss of information about the original image. They can hence stop being a true metric, since a distance metric d, acting on two inputs x and y has the property $d(x, y) = 0 \iff x = y$. Fig. 5.10 demonstrates this; we compare a variety of state-of-art image feature similarity metrics in reconstructing a reference image from noise. The implementation starts with a noisy guess, shown in Fig. 5.10(a), and iteratively minimises the difference between the guess and the reference using gradient descent. Optimising specific features causes the loss of information about other aspects of the image, hampering the ability of many feature extractors to properly measure perceptual image similarity [144]. In this work, we use LPIPS as our main image similarity metric, which is perceptual, flexible, and nearly injective.

5.2 Method

We upsample fluorescence lifetime images by fusing data from two sensing modalities: high-resolution intensity and low-resolution FLIM. Our method leverages statistical correlations between the two measurements to produce a data-driven prior that informs an IR algorithm about the non-sampled points of the scene.

In this study, we introduce two methods for single-sample image fusion upsampling (SiSIFUS). The first approach, referred to as local SiSIFUS, relies on the existence of local correla-

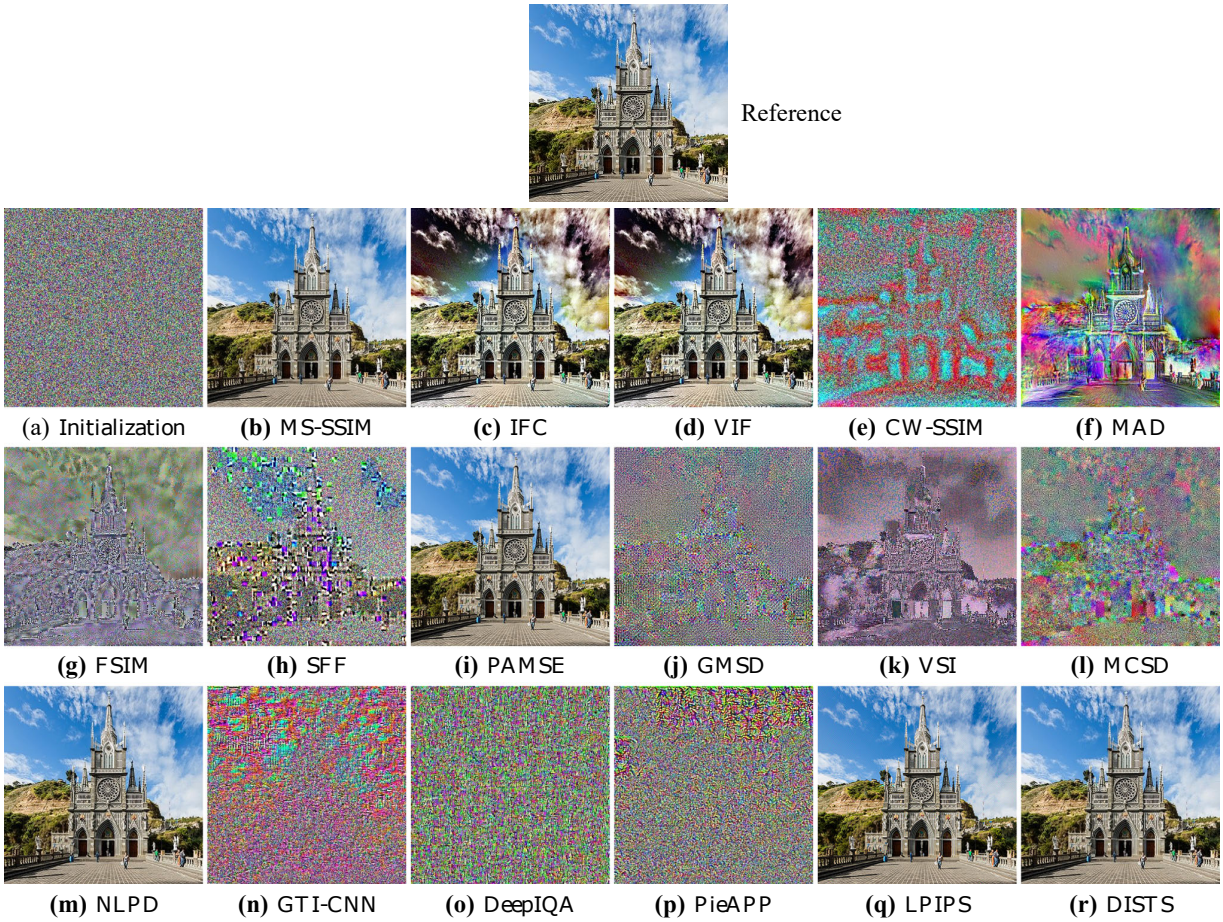

Figure 5.10: Comparison of various image quality metrics, adapted from [144]. Starting from a noisy guess ((**a**)), we try to recover a reference image by minimising various losses ((**b-r**)) comparing the guess to the reference.

tions between LR FLIM pixels and the corresponding intensity pixels, giving local 1D mappings from intensity to lifetime. This concept is similar to those applied in LiDAR SR from RGB and intensity data. [145, 146, 147].

The second method, global SiSIFUS, capitalises on the morphological properties of the sample's fluorescence lifetime. This assumes that similar intensity patches within our FOV have similar lifetimes. Our implementation incorporates a neural network to anticipate fluorescence lifetime from intensity patches, identifying global morphological similarities in the intensity image that can best explain the available FLIM samples. Similar SISR approaches exist, exploiting self-similarity [117], non-local neural networks [148], and self-supervised image morphological clustering [149]. Our method improves upon these by introducing data fusion. To avoid reconstructions being biased towards an out-of-distribution HR training set, our proposed method learns exclusively from the given measurement pair.

Inverse model

Let us consider two images showing the same field-of-view of a fluorescent sample. One is an LR image ($m \times n$ pixels), showing FLIM $\tau_{LR} \in \mathbb{R}^{+m,n}$. The other is an HR image ($M \times N$ pixels), showing fluorescent intensity $I_{HR} \in \mathbb{N}^{M,N}$. The ratio of HR:LR image dimensions, i.e. the downsampling factor in the forward model, or the upsampling factor in the inverse model, is some integer ≥ 2.

Our forward model is very simple. We assume the HR FLIM sample is sparsely sampled (decimated) by some known operator $\mathbf{A}$ to give the LR FLIM image, ignoring any blurring or noise incurred by the optics and detector, $\tau_{LR} = \mathbf{A}\tau_{HR}$.

The SISR inverse model estimates the HR FLIM sample that produced our LR measurement. For this, we optimise a cost function akin to example-based algorithms like NE. Our cost function has 3 terms. The first one asserts data fidelity, ergo that our HR FLIM guess $\hat{\tau}_{HR}$ can be downsampled to give τ_{LR}. Second, we have a learning-based term, which says that our $\hat{\tau}_{HR}$ should be similar to a learned HR prior $\hat{\tau}_p$. The third term is a total-variation regulariser, which encourages sparse finite differences in the HR guess.[10]

Hence, our optimisation objective is:

$$\begin{aligned} \tau_{HR}* &= \arg\min_{\hat{\tau}_{HR}} C(\hat{\tau}_{HR}), \text{ where} \\ C(\hat{\tau}_{HR}) &= \underbrace{\|\mathbf{A}\hat{\tau}_{HR} - \tau_{LR}\|_2^2}_{\text{data fidelity}} + \underbrace{\gamma\|\hat{\tau}_{HR} - \hat{\tau}_p\|_2^2}_{\text{learned prior}} + \underbrace{\alpha\|\mathbf{D}\hat{\tau}_{HR}\|_1}_{\text{TV regulariser}} \text{ such that } \hat{\tau}_{HR} \geq 0 \end{aligned} \tag{5.3}$$

We initialise optimisation with an HR guess equal to the learned prior. This guess is then optimised over 20 iterations with the alternating direction method of multipliers (ADMM).

Local correlation priors (LCP)

Local priors operate on local neighbourhoods; these are independent LR FLIM patches and their corresponding HR intensity patches. Such priors are motivated by observations that FLIM and intensity are often locally correlated (e.g. calcium imaging, van der Linden et al [12], Supplementary Movie 5; SNARE imaging, Verboogen *et al.* [151], Fig. 1 (c) 2^{nd} row). Also, under certain conditions, lifetime and intensity are explicitly linked, albeit in the presence of covari-

[10] This describes our *a priori* belief that the FLIM image should have a few sharp gradients. We use the anisotropic form of TV [150].

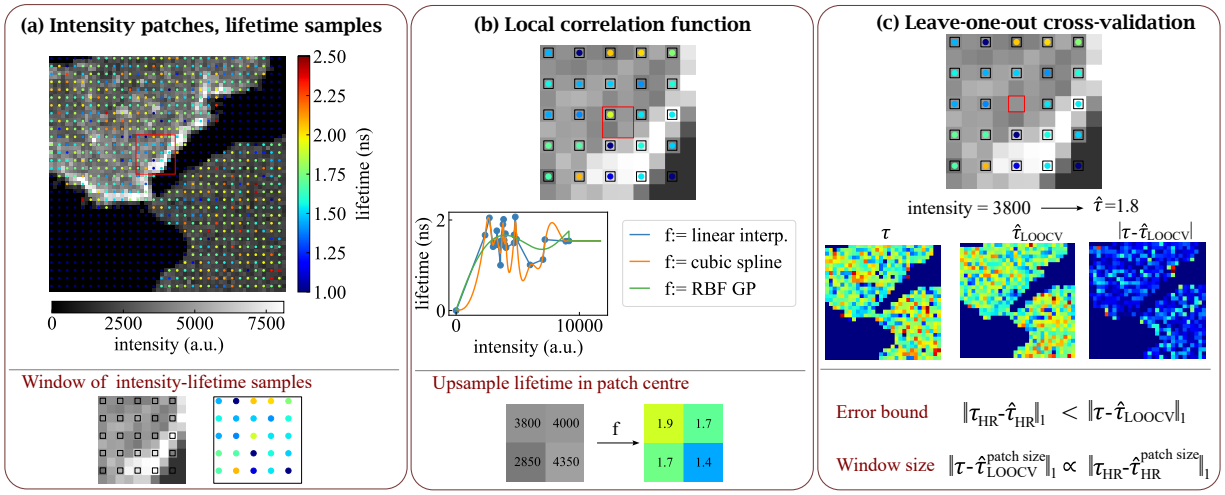

Figure 5.11: **(a)** Illustration of HR intensity and corresponding LR fluorescence lifetime. At the bottom, an HR intensity - LR lifetime patch pair in a given window of the FOV. **(b)** We fit some function to a set of intensity-lifetime pixels in a given window and predict the central pixels. **(c)** Schematic of leave-one-out cross-validation (LOOCV), where the central intensity-lifetime pixel pair is removed for analysis, its value guessed via our algorithm and validated against the measured value. This is repeated for all patch centres, i.e. all pixels in the LR FLIM measurement.

ates. For instance, in FLIM-FRET (see Chapter 2 Sec. 2.5) they are directly proportional, with fluorophore concentration acting as a covariate.

Fig. 5.11 (a) shows an example of an HR intensity image with the LR FLIM image overlayed on it. This image is broken up into windows, each window containing one central LR pixel. We note that pixels at the edges of the LR FLIM image are not centred on their windows. Having found these windows, each one is processed independently.

In a given window, the intensity and lifetime pairs are vectorised and fitted with a function, f, such that $\hat{\tau} = f(I)$. Thus, our lifetime estimate $\hat{\tau}$ for pixel $(\lambda i + x, \lambda j + y)$, is:

$$\hat{\tau}_{\lambda i+x,\lambda j+y} = f_{i,j}(I_{\lambda i+x,\lambda j+y}) \tag{5.4}$$

with samples $i \in \{0, 1, ..., m-1\}$ and $j \in \{0, 1, ..., n-1\}$, and $0 \leq x, y < \lambda$; $f_{i,j}$ is the fit obtained on window (i, j).

Fig. 5.11 (b) demonstrates this LCP algorithm. We look at the co-registered intensity-lifetime pixels in a given window and fit them with some function (we used linear interpolation for our results). This function then maps the centre of the HR intensity patch to the corresponding HR FLIM values.

Leave-on-out cross-validation (LOOCV). We can use leave-one-out cross-validation to make

some *a priori* assumptions on the reconstruction quality. For this, we remove the central intensity-lifetime pixel pair of each window, and run our algorithm on all windows, reconstructing only the left-out lifetime value. This gives us a low-resolution lifetime map τ_{LOOCV} matching the pixels of τ_{LR} - see Fig 5.11 (c). Sec. 5.3 gives further details.

Global morphological priors (GMP)

We developed a second algorithm to exploit features that are globally repeated in the scene, akin to self-similarity SR. We find that fluorescent samples, particularly in the field of bio-imaging and microscopy, often contain repeating features with similar lifetime properties. For example, cells commonly absorb and express fluorophores in specific subcellular structures, like the cell membrane [152], vesicles [153], or the nucleus[154]. These structures can be sampled on multiple detector pixels. It makes sense to try to use the 'known' samples (where both the CMOS feature shape and the corresponding lifetime are sampled clearly) to estimate the 'unknown' samples (which are on dead regions of the FLIM detector and hence have unknown lifetime).

We approach this with global morphological priors (GMPs), that learn to map high-resolution intensity patches (in bio-imaging contexts, these patches describe morphology) to lifetime at the

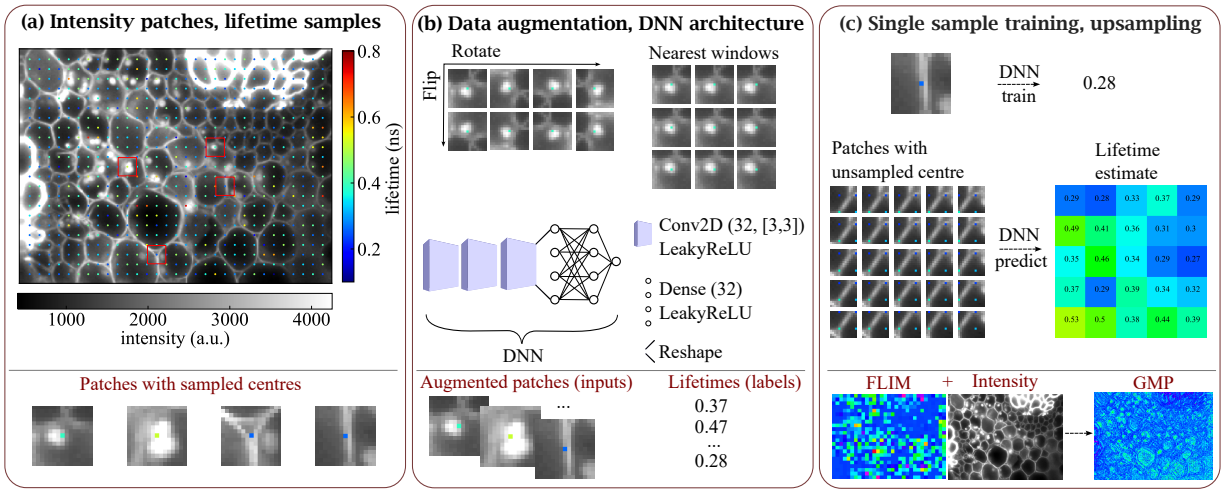

Figure 5.12: **(a)** Fluorescent intensity from the convallaria - acridine orange samples is shown, with sparsely downsampled lifetime samples overlayed. We extract intensity patches from this image; a few of them correspond to a central lifetime sample. Such patches are training data, which we can use to predict the central lifetime of the rest of the patches. **(b)** Training inputs (patches) are augmented via rotation and mirroring. They can be further augmented by adding their nearest neighbours to the training set; since these neighbour patches are unlabelled, their lifetime is assumed to equal that of the sampled patch. The DNN architecture is simple, consisting of 2D convolutions followed by fully connected layers. **(c)** Finally, the trained DNN evaluates patches with unsampled centres, thus super-resolving the lifetime image.

patch centre. Fig. 5.12(a) shows a fluorescent intensity image of Convallaria rhizomes dyed with convallaria with FLIM samples overlayed. The idea is to learn to correlate these training patches to lifetime, assuming this correlation will generalise to test patches (those with un-sampled centres).

Dataset augmentation. We augment our training set to improve the likelihood of generalisation, as shown in the top-left of Fig. 5.12(b). Firstly, we assume that fluorescence lifetime is invariant to the orientation of the intensity patch, so we mirror and reflect the patches to increase the dataset diversity. This gives us 8× more samples than what we started with. We note that some processes may cause lifetime statistics to depend on the orientation of a patch in our FOV (for example a directional external field like sunlight, an electric field, temperature gradient or gravity). In this case, two identical-looking but rotated patches would exhibit different lifetimes, so our dataset augmentation process would harm SR quality.

A second dataset augmentation approach assumes that lifetime varies slowly over the FOV. Therefore we add (unlabelled) patches neighbouring known patches to our training set, and artificially label them with the same lifetime as their sampled neighbour. Our method is shown in the top-right of Fig 5.12(b). If we add the nearest neighbours, this gives us $3 \times 3 = 9\times$ as many samples as we originally have - combined with the invariance-based augmentation, 72× as many. However, this approach is inherently noisy unless the lifetime distribution is flat, so it is only applicable if the lifetime varies slowly. Since we do not know whether this is true *a priori*, we only augment our data this way when the training set size is very small (in our case, for upsampling factors of 8×8, 16×16.

Single sample training mitigates training set bias. As described in Sec. 5.1.5, FLIM SR has a series of challenges, in particular the lack of datasets and statistical studies of FLIM. This makes it difficult to create example-based SR methods with an external, pre-made dictionary. Moreover, the difficulty of assessing a super-resolved image without seeing a reference HR image makes it difficult to judge whether our SR image incorporates out-of-distribution features. Therefore, we use a self-similarity-based approach. We extract all our training examples from the given image (or equivalently, a set of FOVs of the same sample). This scheme ensures SR does not use out-of-distribution intensity-lifetime correlations. We note that this does not ensure our SR image is correct, just that it generates samples from the same patch-lifetime distribution as is observed in the LR sample.

We train the DNN on 13×13 patches, with a batch size of 100 over 150 epochs, using standard gradient descent (ADAM optimiser [155]) via backpropagation, with MAE as the training loss. On an NVIDIA GeForce RTX 2080, upsampling an image containing 125×125 intensity patches corresponding to the same number of lifetime pixels, with 8× data augmentation for rotation and reflection invariance, took $\sim$ 25 minutes of training.

5.3 Results

LCP function comparison. We performed a study of what function is best for LCP. The functions we tried were Kriging interpolation (radial basis function Gaussian processes, taking the mean of the fit), B-spline fitting (linear, quadratic, cubic), and interpolation (nearest, linear, cubic). We compared the SR quality using MAE and LPIPS on four biological samples at upsampling factors of 2,4,8, and 16x. Fig. 5.13 shows our results; based on this study, we used linear interpolation for LCP.

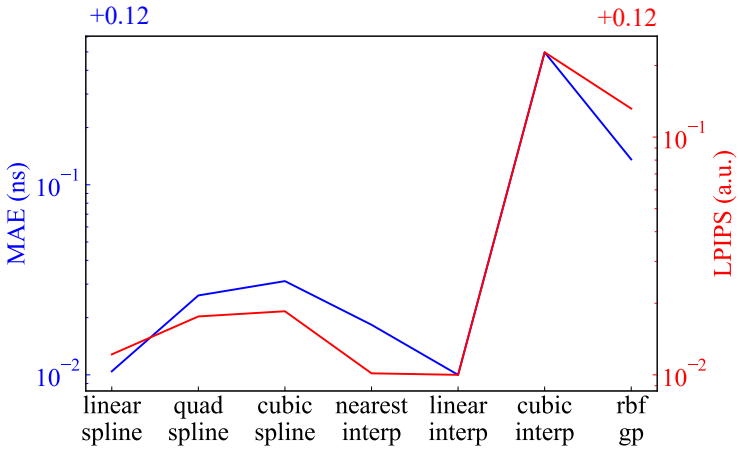

Figure 5.13: MAE and LPIPS for various LCP functions. The plotted values show the mean across 4 samples and 4 upsampling factors. Linear interpolation gives the best performance overall.

Local SiSIFUS for 2x upsampling. We demonstrate our approach on an SK-OV-3 ovarian cancer cell validation sample dyed with Rac1-Raichu, a FRET probe consisting of an Clover donor and an mCherry acceptor (see Fig. 2.3). It was recorded with a custom widefield microscope, which uses HORIBA Scientific's FLIMera SPAD array [156] to register a 192×128×326 fluorescence datacube of the sample. The same object plane is imaged in parallel by a high-resolution (2048 × 2048) sCMOS camera. These images are then computationally co-registered to match their fields of view perfectly. Lifetimes were fitted to the datacube using least squares deconvolution.

The intensity image is shown in Fig.5.14(a). We decimated (sparsely sample) the datacube to give the low-resolution lifetime images τ_{LR} at various downsampling factors, like the 2× downsampled one shown in Fig.5.14(b). This LR image was upsampled using local SiSIFUS, with a 5×5 window LCP. The super-resolved image is then compared to the τ_{GT} using LPIPS.

Fig. 5.14(c) shows LPIPS as a function of upsampling factor between the ground truth and bilinear interpolation in FLIM space (FS) vs LCP SiSIFUS, Fig. 5.14(d-f) respectively. In FS interpolation, first the lifetime of the LR fluorescence datacube is estimated, then the low-resolution

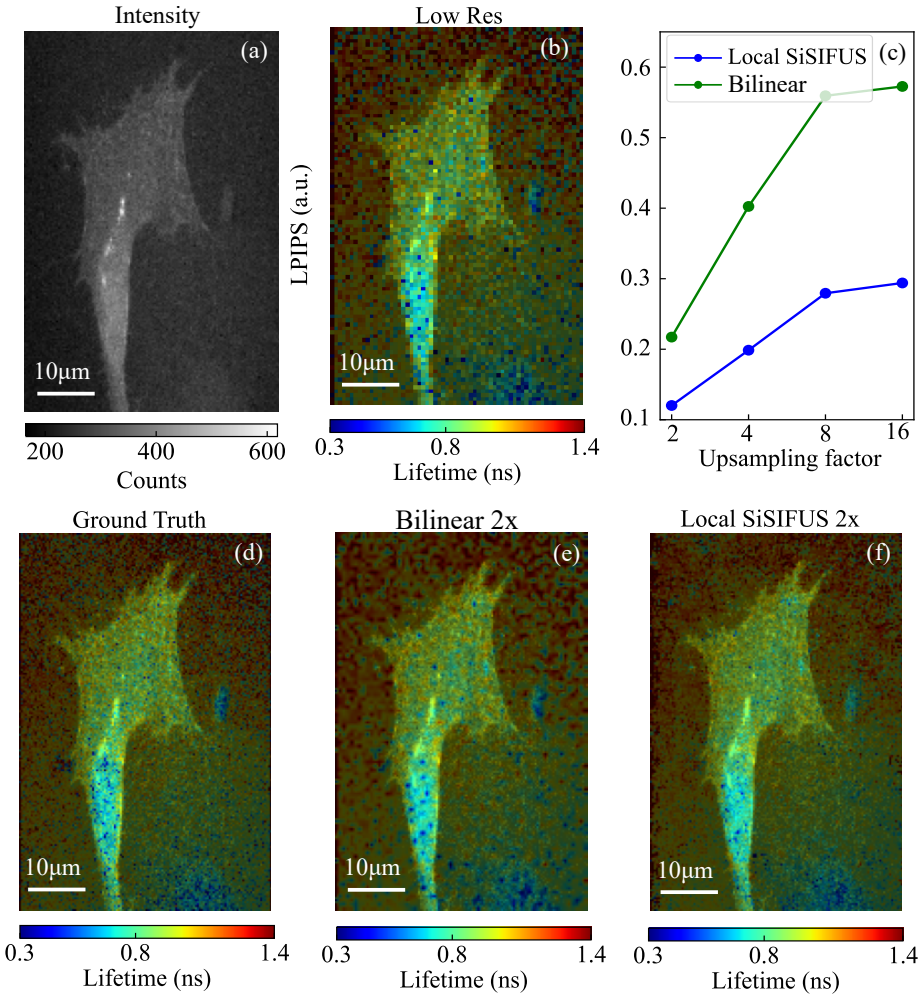

Figure 5.14: **(a)** High-resolution (192 × 128) fluorescence intensity image (SK-OV-3 Rac1-Raichu). **(b)** Low-resolution (96×64) lifetime measurement, to be upsampled. **(c)** Our perceptual similarity metric of choice (LPIPS) for 2,4,8,16× upsampling.**(d-f)** High-resolution ground truth, 2×2 bilinearly interpolated image in FLIM space, 2×2 LCP-SiSIFUS (proposed method). The LCP-based reconstruction is more perceptually similar to the ground truth than interpolation.

FLIM image is interpolated.[11] Our method outperforms interpolation at all upsampling factors.

Local SiSIFUS for 4x upsampling. We illustrate 4×4 (LCP-based) SiSIFUS on further SK-OV-3 Rac1-Raichu validation samples. 256×256 pixel images of a 301×301 μm^2 FOV were acquired using a commercial LaVision BioTec TriM Scope II system, using two-photon excitation in a confocal scanning arrangement, with a photo-multiplier tube (PMT) as the detector. The PMT uses TCSPC with 75 time-bins of 160ps duration each.

[11] Alternatively, one could perform photon-space interpolation, where we instead interpolate the LR datacube to the HR size and then fit this HR datacube. FS interpolation is faster than PS, as lifetime fitting is performed on a smaller datacube, and then only a single frame is interpolated. Done correctly, FS and PS tend to produce comparable results. However, we have noticed that people often do FS interpolation after thresholding low-intensity regions. This causes unrealistic artefacts where the positive lifetime region meets the threshold boundary. Reordering the pipeline (fit first, interpolate in FLIM space, and then threshold) can help avoid this. Similarly, if the low-intensity lifetime estimates are biased instead of just noisy, FS edge effects can appear so PS interpolation might be preferred.

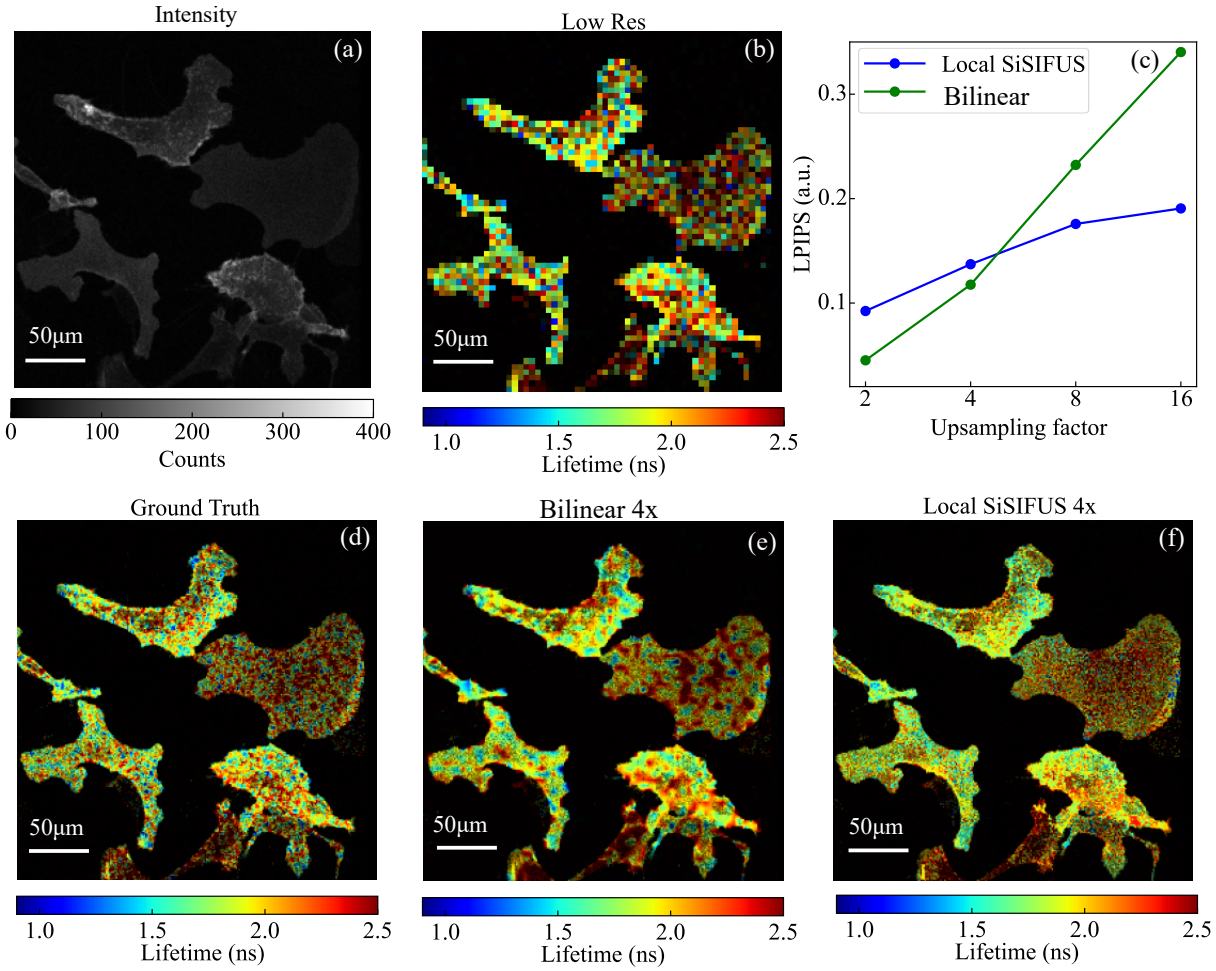

Figure 5.15: **(a)** High-resolution (256×256) intensity measurement (SK-OV-3 Rac1-Raichu). **(b)** Low resolution (64×64) FLIM input. **(c)** LPIPS for 2, 4, 8 and 16 fold upsampling. **(d-f)** Ground truth, and 4×4 interpolated vs LCP-based SiSIFUS.

Fig. 5.15(a) shows the high-resolution lifetime image, while Fig. 5.15(b) shows the low-resolution lifetime input, obtained through 4× decimation of the ground truth lifetime image. Fig. 5.15(c) evaluates LPIPS for upsampling between the ground truth (Fig. 5.15(d)), and interpolation (Fig. 5.15(e)) vs local SiSIFUS (Fig. 5.15(e)). Interpolation proves to be similar or marginally better than our proposed method for 2 and 4× super-resolution. However, SiSIFUS is better at higher upsampling factors as its upsampling quality degrades more gradually than interpolation's.

Global SiSIFUS for 8x upsampling. Global (GMP-based) SiSIFUS is validated on a Convallaria rhizome sample dyed with Acridine Orange, imaged using our previously mentioned FLIMera SPAD-sCMOS setup. By scanning twice, we obtained image resolutions of double the SPAD's pixel count, giving an x, y, t datacube of $192 \times 256 \times 326$. The CMOS was computationally co-registered to this.

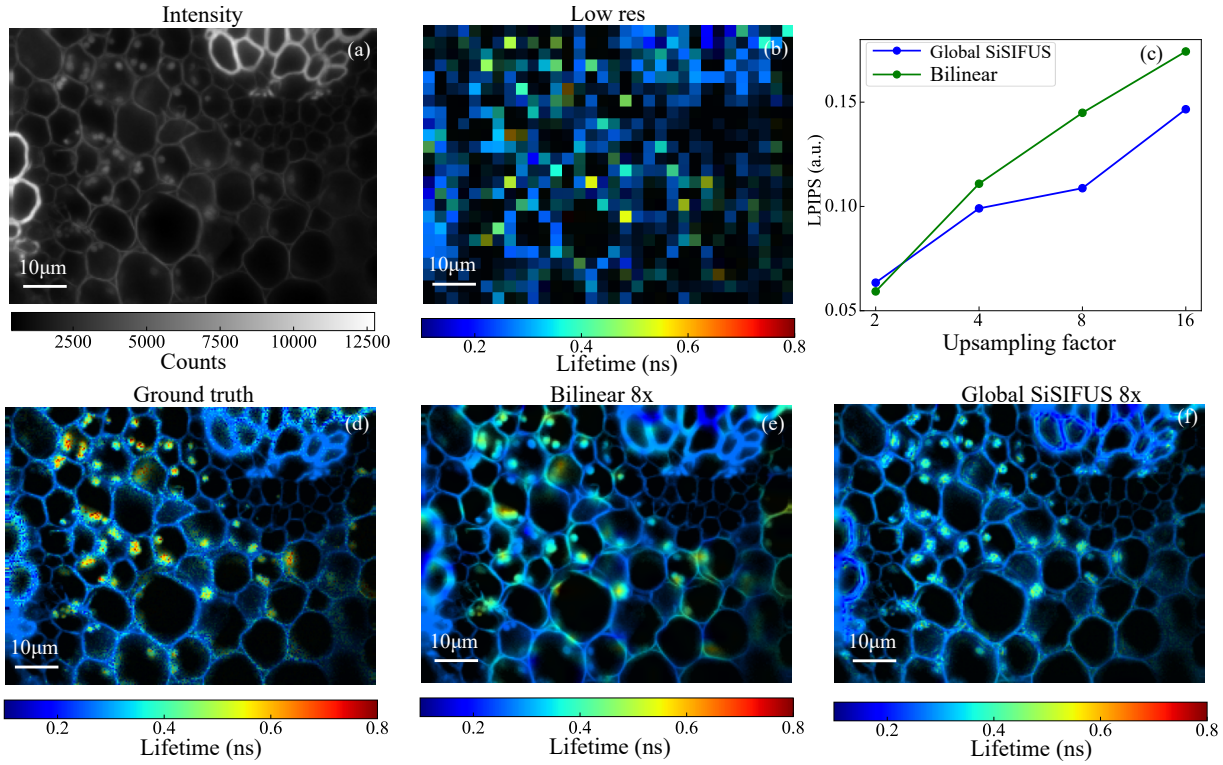

Figure 5.16: **(a)** High resolution (192×256) intensity image (Convallaria - Acridine Orange). **(b)** Low-resolution (24×32) lifetime. **(c)** Assessment of global SiSIFUS (our method) vs interpolation, using LPIPS score. Our method is similar to interpolation at low upsampling factors (2 to 4) but surpasses it at higher factors (8 and 16) **(d-f)** insets showing the ground truth and comparing interpolation vs our scheme 8×8 super-resolution.

Fig.5.16 shows our results. GMP-based SiSIFUS achieves similar perceptual quality to interpolation at 2 and 4 × upsampling, and outperforms it at 8 and 16 ×. The visual difference is stark: global SiSIFUS appears to recognise that globules have higher lifetimes than cell walls, maintaining contrast between these structures more consistently than both modes of bilinear interpolation. However, our scheme uses global statistics, removing hotspots (high-lifetime, yellow/red-coloured areas) that act as statistical outliers.

Since global SiSIFUS requires training a model from scratch each time, different training instances give different global morphological priors, depending on the model initialisation. To report representative results, we made three priors per sample and per upsampling factor (the DNN architecture was retrained thrice for all the global SiSIFUS results shown in this thesis), and we post-selected the prior with median LPIPS to the ground truth for our global SiSIFUS pipeline.

Global SiSIFUS for 16x upsampling. Fig.5.17 displays more validation results, showing a Madin-Darby canine kidney (MDCK) cell sample expressing Flipper-TR, a tension sensitive FLIM probe. The image shows a 512×512 sample grid covering a 163×163 μm^2 FOV, acquired

using the TriM Scope system mentioned earlier. Interpolation struggles at high upsampling factors (i.e. high decimation factor of the ground truth, thus large pixel pitch) because local correlations between the LR-pixels are weak. However, there is a global trend that cell membranes have high lifetimes, and vesicles have low lifetimes. Our GMP-based method can identify and learn this pattern and exploit it for better reconstructions.

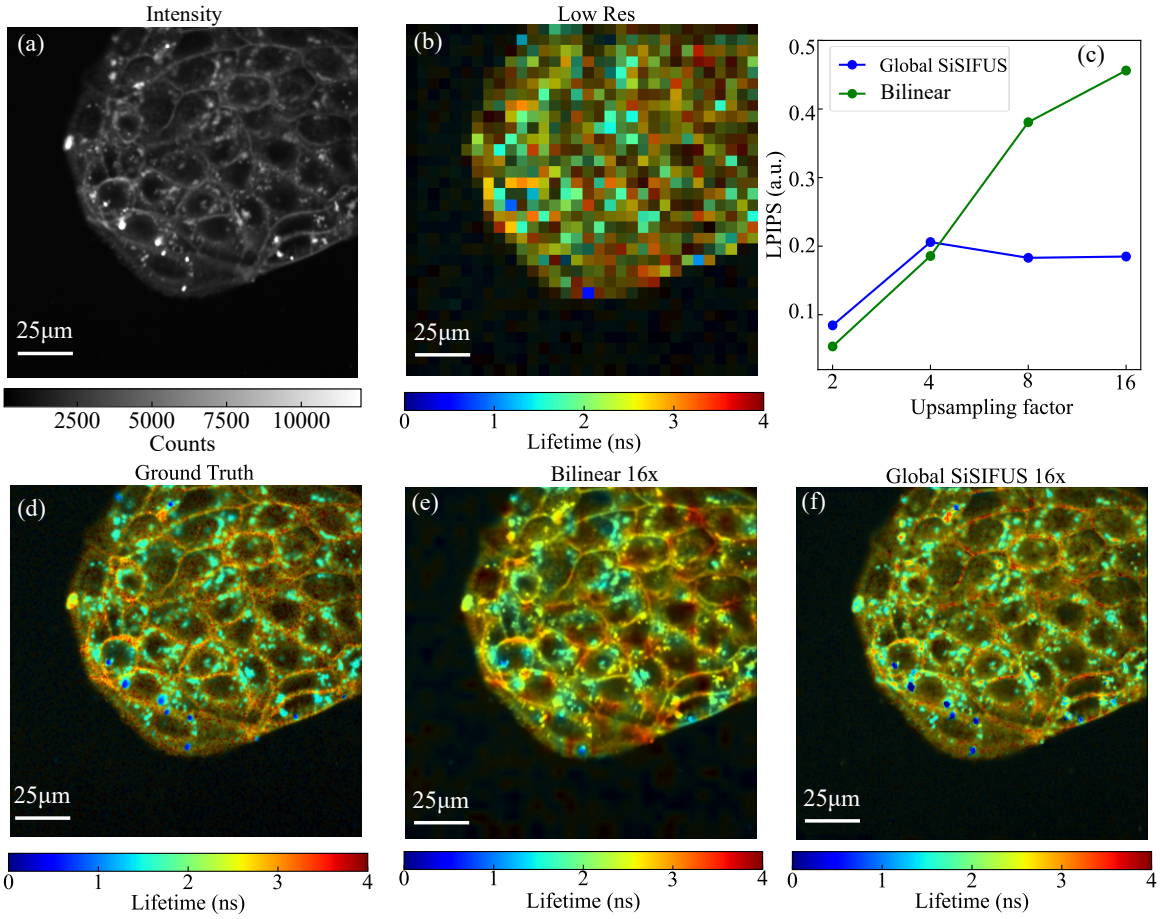

Figure 5.17: **(a)** High-resolution (512×512) fluorescence intensity sample (MDCK - Flipper TR). **(b)** Low-resolution (32×32) lifetime image. **(c)** Assessment of global SiSIFUS (our method) vs interpolation, using LPIPS score. Our method performs similarly to interpolation at low upsampling factors (2 to 4) and better than it at higher factors (8 and 16). **(d-f)** The ground truth and 16×16 super-resolved images.

SiSIFUS testing. We test our data fusion methods using the same protocols as established for the earlier validation sets. For this, a set of samples were decimated to produce low-resolution images, and upsampled again using SiSIFUS. The results shown in Fig 5.18 are of a sample of B16-F1 melanoma stained with Vinculin-TS, a focal adhesion tension probe. It is upsampled with LCP-based SiSIFUS; our method performs similarly to interpolation for low-upsampling factors, where lifetime varies smoothly between samples, but outperforms it for 8 and 16× upsampling, maintaining high-contrast image features while interpolation becomes blurred. Insets show the 16× upsampling case.

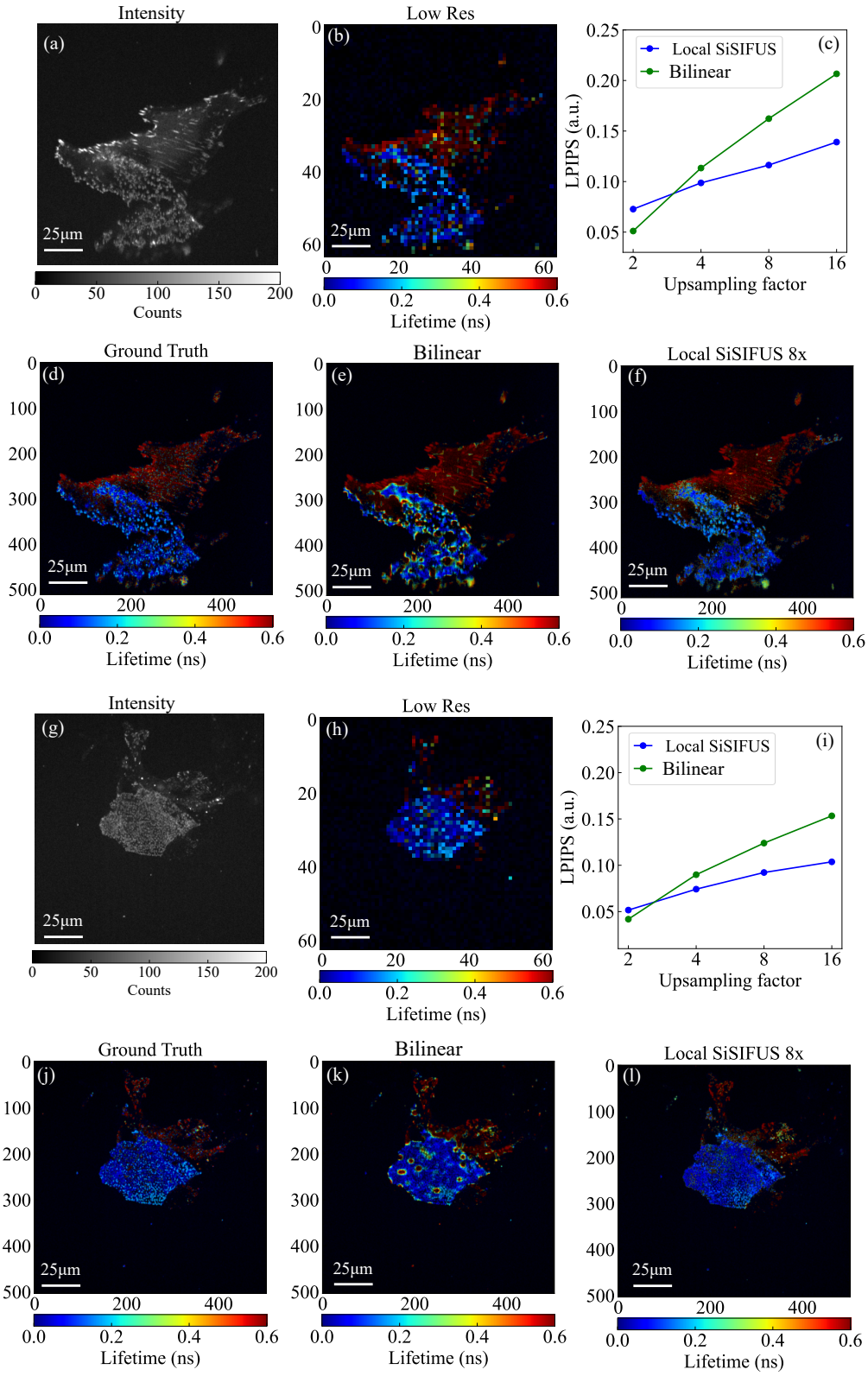

Figure 5.18: Testing local SiSIFUS.**(a,b)** 504 × 504 intensity and lifetime images (B16-F1 melanoma - Vinculin-TS). **(c)** LPIPS as a function of upsampling factors between the **(d)** ground truth against **(e)** bilinear interpolation and **(f)** local SiSIFUS. **(g-l)** Same plots for another field of view; we observe similar trends between interpolation and our scheme.

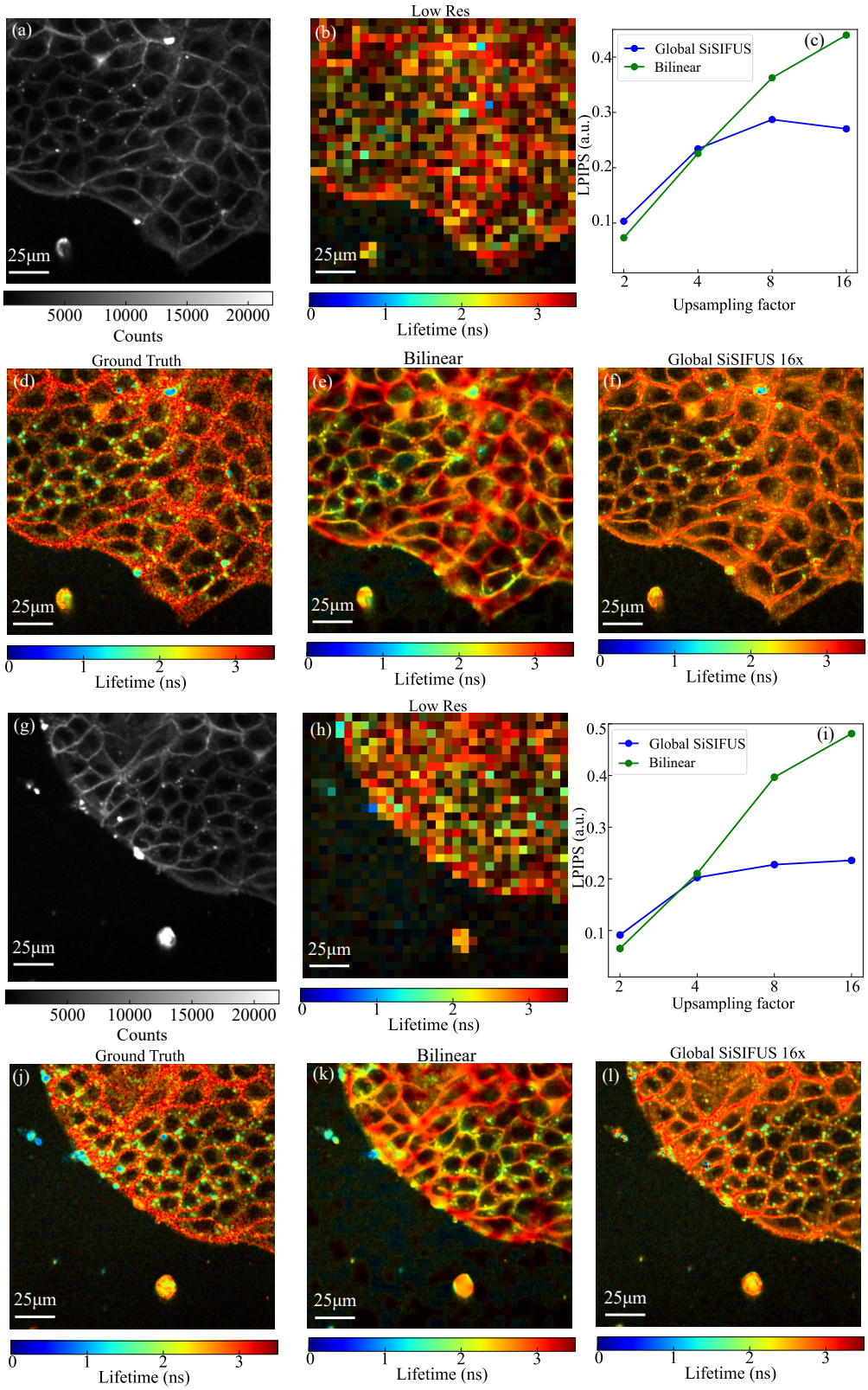

Figure 5.19: Testing global SiSIFUS. **(a,b)** High-resolution (512 × 512) intensity and low-resolution (32 × 32) fluorescence lifetime image (MDCK - Flipper-TR). **(c)** LPIPS for various upsampling factors, comparing **(d)** the ground truth to **(e)** bilinear interpolation and **(f)** local SiSIFUS. **(g-l)** A different region of interest from the same sample.

Fig 5.19 shows a sample of MDCK cells stained with Flipper-TR, super-resolved using GMP-based SiSIFUS. Again, interpolation works well until the distribution between pixels varies smoothly (2 and 4×) upsampling, reaching similar performance as our method. However, for higher upsampling factors, SiSIFUS's ability to distinguish low- and high-lifetime structures based on the intensity image lets it outperform interpolation significantly.

LOOCV study. In leave-one-out cross-validation (LOOCV), the central intensity-lifetime pair is removed from each window, and its lifetime predicted using the rest of the data in the window using the LCP protocol. We can then compare this prediction with the known lifetime value. Consequently, this simulates upsampling the image, however its quality can be assessed without knowing a high-resolution ground truth.

Leave-one-out cross-validation acts as an uncertainty upper-bound for the mean-absolute-error (MAE) of LCP. This is because of two reasons. Firstly, our LCP windows have 1 less intensity-lifetime datapoint than normally. Secondly, the distance of the central test pixel from its neighbours is greater in LOOCV than in regular LCP generation. After all, in LOOCV, 1 pixel-pitch separates the central pixel from its vertical and horizontal neighbours. In LCP, the up-sampled pixels have at least one pixel which is $\leq 1/\sqrt{2}$ pixel-pitches away. Therefore, LOOCV has weaker spatial correlation than regular LCP.

In Fig. 5.20, we show evidence that LOOCV MAE acts as an upper bound on LCP MAE, for

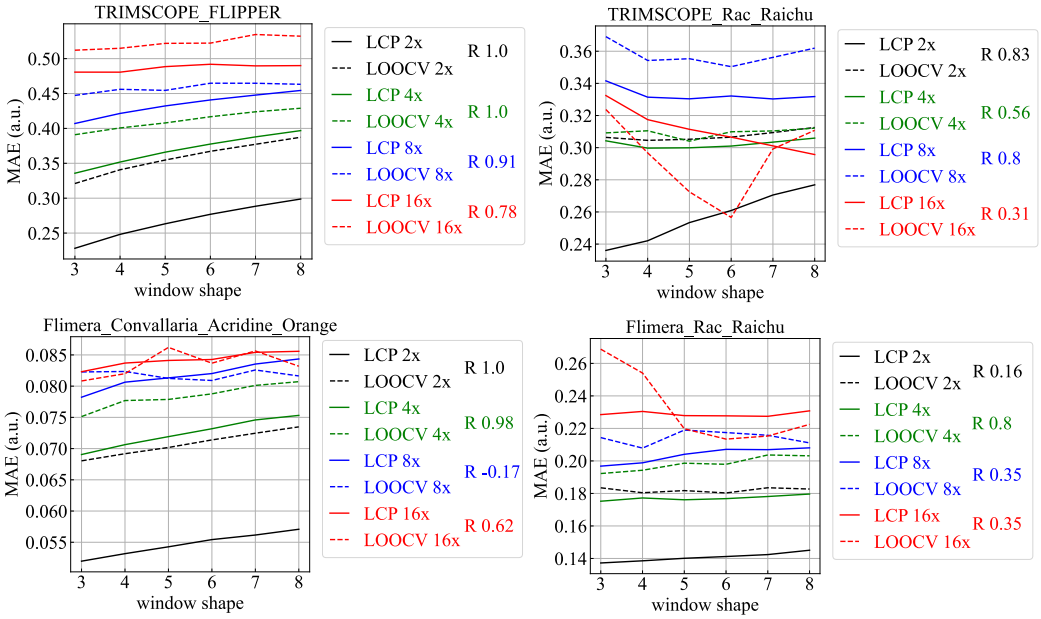

Figure 5.20: We show four samples (one per graph) used to generate local correlation priors (LCPs) at various upsampling factors (2,4,8, and 16x, denoted using different colours). For LCP window shapes in {3,4,5,6,7,8}, we compare LOOCV MAE and LCP MAE, quoting the Pearson correlation coefficient R in the legend.

low upsampling factors (2x and 4x) that LCP works well for. Further, LCP MAE and LOOCV MAE are positively correlated for the various window shapes (again, for low upsampling factors where LCP works well), hence LOOCV can be used to can be used to *a priori* decide what window size to use to minimise LCP MAE. These results evaluate the local prior, not the complete SiSIFUS pipeline, which includes inverse retrieval.

5.4 Discussion

We introduce SiSIFUS, a novel SR scheme that upsamples an LR FLIM image with the help of a second, locally or globally correlated HR intensity image. This scheme has two flavours, local correlation prior (LCP) and global morphological prior (GMP) based SiSIFUS.

LCP-SiSIFUS has parallels with interpolation, where unknown pixel values are inferred based on their distance from measurements. In our method, lifetime correlates with intensity similarity, not with spatial distance from existing samples. This allows local SiSIFUS to maintain sharp spatial boundaries, and the resulting SR image will be biased to match the pattern of the intensity image and to follow its gradients. Overall, LCP works well for super-resolving samples with strong, monotonic local intensity-lifetime correlations.

In GMP-based SiSIFUS, a DNN is trained from scratch on patches of our HR image to output corresponding LR FLIM labels. GMP-SiSIFUS shares similarities with example-based SR methods such as NE and self-similarity. Fig. 5.21 compares the NE algorithm to ours.

The prior is plugged into an inverse retrieval algorithm with a data fidelity term and a TV minimisation regulariser. We note that the TV algorithm is used to smooth out small pixel-to-pixel fluctuations in lifetime, whilst having a relatively smaller effect on sharp edges, ergo we observe a denoising effect. Together with the data fidelity term, TV acts like a diffusion equation, with the sparse LR samples acting as sources. Such diffusion enforces HR pixels nearby our LR samples to be similar to their LR neighbours, and this effect is proportional to the pixel offset of the HR target pixel from its LR neighbours. The TV regulariser is thus a denoiser and a flexible interpolator.

This diffusion competes with the learned prior in generating the final output, whose effect is independent of the distance between the HR target from its LR neighbours. Therefore, for low upsampling factors (where HR target pixels are surrounded tightly by LR samples), the diffusion effect dominates, whereas, for high upsampling factors, the learned prior dominates. This is beneficial since we expect spatial correlations between HR targets and LR samples to scale according to scale inversely with distance, thus global GMP is advantageous for larger upsampling

factors (or equivalently, greatly separated LR samples).

During this project, we repeatedly found that pixel-wise metrics did not reflect our perception of which super-resolved image matches the ground truth better than the other; SSIM was not much better. To address this, we use LPIPS instead; this metric seems generally consistent with our perceived image similarity. LPIPS is a deep-learning-based metric, but we use an open-access implementation that runs a fixed, pre-trained algorithm, making our results fully reproducible.

Lastly, we note that our method is not customised to FLIM upsampling. As the name states, it is simply a super-resolution scheme drawing on information available from a single sample, exploiting local or global mappings between two statistically correlated data modalities. Potential further applications of local SiSIFUS could be upsampling PET scans with CT scans (e.g. for tumour detection) or thermal images with intensity images (e.g. for detecting gas in a compound), where the high-resolution image can help set boundaries on the LR image of interest. Global SiSIFUS could be applied to things like upsampling infrared spectroscopy images with RGB images (e.g. for plastic sorting in a recycling plant) or upsampling synthetic aperture radar scans with satellite images (e.g. for surface mapping), where large numbers of globally repeating features in the high-resolution image modality can guide the upsampling process.

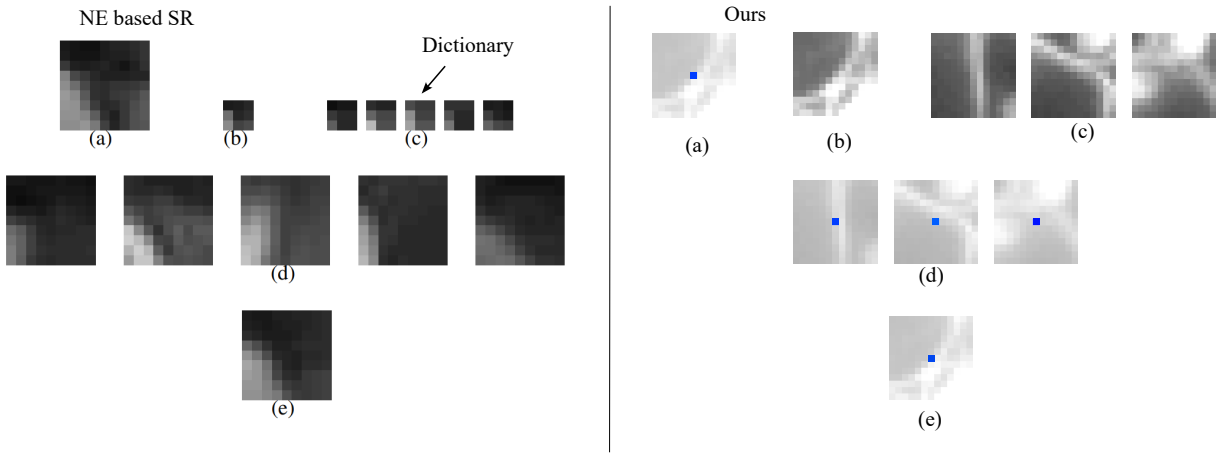

Figure 5.21: Schematic comparing example-based SR via neighbour embedding (same as Fig. 5.3) to our GMP SiSIFUS. (a) The NE target is an unknown high-resolution patch; our target is the unknown lifetime in the patch centre. (b) The NE input is an LR patch; our input is an unlabelled patch. (c) The NE algorithm explicitly searches for similar LR patches in a generic dictionary; our neural network learns implicitly associates the LR patch with similar intensity patches. (d) The corresponding NE HR patches; our corresponding lifetime values. (e) The NE output is an HR patch; our output is a lifetime label.

Chapter 6.

Encoding space in time: flash single-pixel depth imaging

Summary. To form a 3D image of a scene, we must encode spatial information about the object on the detector. There are two conventional approaches: widefield or scanning-based imaging. Widefield (camera) imaging uses a lens to image the scene onto many pixels, each pixel detecting light arriving on the lens at a given angle. If we scan instead, we illuminate only a region of the scene and detect the back-reflected light; we scan the scene by illuminating different regions one at a time. Here we show a novel, deep-learning-enabled imaging paradigm, combining flash illumination with single-pixel detection.

6.1 Introduction.

Overview

There are many approaches to 3D imaging. These range from stereoscopic imaging that relies on viewing the same scene from two different angles and estimating depth from the difference between the two images [157], through holography-based approaches that capture how objects modulate the shape of a wavefront reflecting off the surface of said objects [158], through time of flight (ToF) methods. Our work focuses on the latter; for an introduction to encoding range with ToF methods using LiDAR/RADAR, refer to Chapter 2, Sec. 2.2.

However, range is only one aspect of 3D imaging: we need to require angular information as well. Just like with FLIM, there are two common approaches for encoding angular information onto a detector. One is to use a lens, which focuses light coming from a given angle onto a fixed pixel. This works with flash illumination and widefield detection. The other is structured illumination combined with single-pixel detection. Both of these paradigms are well explored and allow images to be formed straightforwardly.

Here, we consider a new, ill-posed image formation scheme. We flood-illuminate the scene with a pulsed laser and capture back-reflected light with a lens, focusing it on a single SPAD pixel. The pulsed illumination forms a thin shell of light that propagates through the scene like the surface of an expanding balloon, reflects off objects, and is collected by the SPAD pixel. The

objective is to reconstruct the distance of objects from the camera; this is a 2D projection of the 3D scene.

Fig. 6.1(a) shows a schematic of the imaging setup. A laser emits a broad pulse that propagates in the air until it hits an object. It then gets reflected, and the SPAD picks up some of the reflected light. The SPAD picks up individual photons, tags them based on time of flight (ToF), and creates a histogram of ToFs.

Conventional schemes. In conventional LiDAR approaches, light from a given angle has a single SPAD peak. For example, a flash-illuminated scene viewed by a widefield SPAD array is shown in Fig. 6.1(c)(i-iii). The total measurement is a SPAD array datacube (x, y, t), where each (x, y) frame is a 2D snapshot of the scene at a given time. (i) Shows an early frame (x, y, t_1), containing echoes from nearby objects; in (ii-iii) we show later frames (x, y, t_2) and (x, y, t_3), containing echoes from farther objects. Each SPAD pixel has a peak light intensity in one of

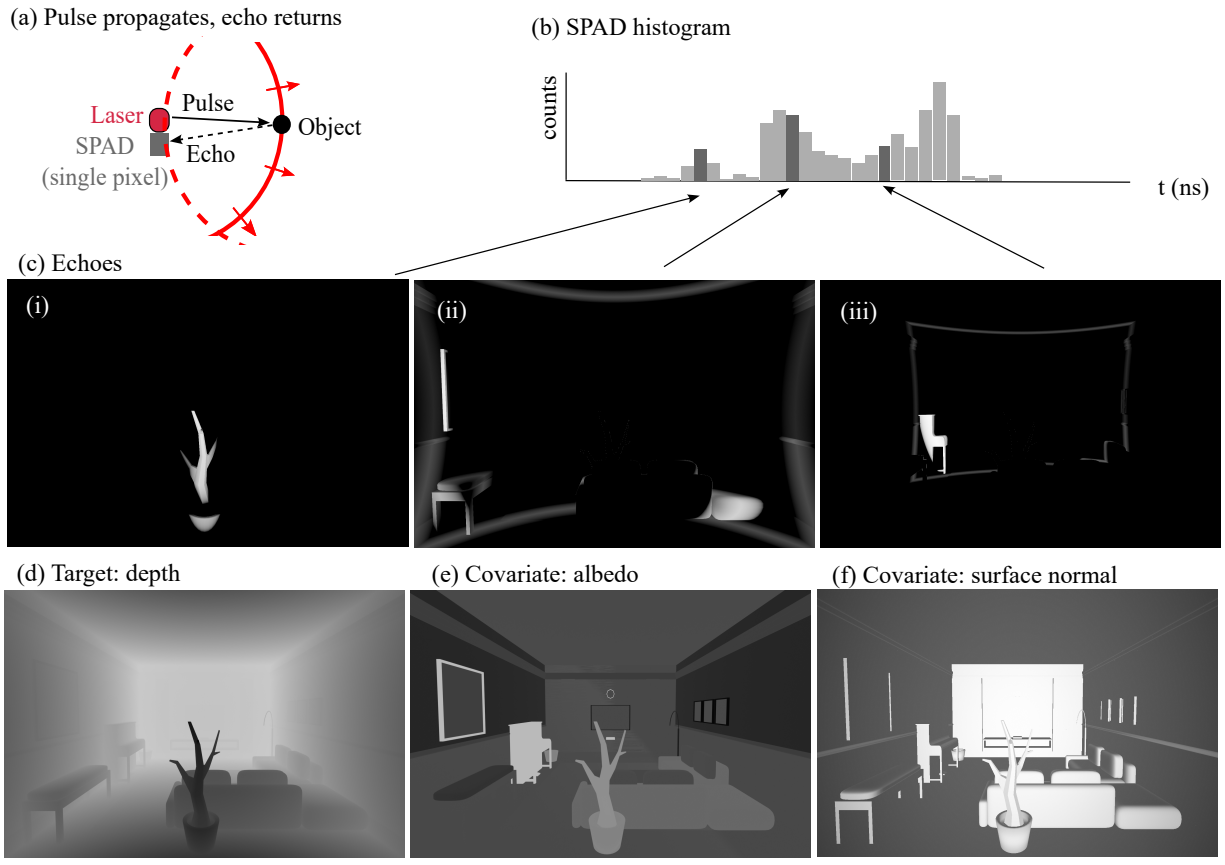

Figure 6.1: **(a)** Schematic of a pulsed imaging setup. **(b)** Example of a SPAD histogram, measuring the total (spatially integrated) light intensity in echoes from the scene. **(c)** Demonstration of the echoes that the SPAD measured. **(i)** Echoes from nearby objects, **(i-ii)** echoes from farther objects. **(d)** The target is to reconstruct the depth profile of the scene **(e-f)** Apart from the depth profile, the albedo (reflectivity) and surface normals (orientation w.r.t. to the camera) of the scene influences our measurement.

the frames, so reconstructing the 3D scene is trivial: the peak time of flight in each pixel of the SPAD array gives us the depth in that pixel directly.

Our scheme erases all angular information about the scene, so the SPAD measurement (Fig. 6.1(b)) contains many peaks. Each peak is the sum of an entire echo (Fig. 6.1(c)), but we do not know the spatial distribution of reflectors that generated this echo. For example, the potted plant in (Fig. 6.1(c)(i)) could have been towards the left of the room, or hanging from the ceiling, or another object entirely, or even two separate objects; all we know is the total light reflected with a given ToF. Reconstructing the potted plant from a few neighbouring histogram bin values is an incredibly ill-posed inverse problem, and the scene contains multiple objects beside the plant. Mathematically, all points at distance $ct_i = \sqrt{x^2 + y^2 + z^2})$ have the same ToF $2t_i$ on the SPAD, hence we cannot obtain a unique solution to the inverse problem of reconstructing the scene from a single temporal histogram.

To make matters worse, the values in the histogram bin are also dependent on the albedo (reflectivity) of objects in the scene, shown in Fig. 6.1(e). We get a higher signal from objects of higher albedo. E.g. for optical wavelengths, white objects reflect more light than black objects, causing larger histogram spikes. Fig. 6.1(f) shows the projection of the scene surface normal vectors onto the angular offset from the camera. This is another covariate that affects how much light we get back. If the surface is flat against the transceiver, it reflects more light to the camera. The more directly a surface is facing the transceiver, the higher the return signal of that surface. If reflections are diffuse (e.g. from a Lambertian surface), we still get some back-reflected light from surfaces that are not pointed directly at the transceiver, but we get less light than from specular surfaces from objects directly facing the camera. Albedo, surface normals, and surface specularity act as covariates.

Challenges, opportunities, and prior work

From a causal inference point of view, if we know the 3D properties of the scene (causes), it is easy to estimate the SPAD histogram (effect). In other words, the forward model is trivial to solve (even if it takes a long to compute); indeed, ray-tracing algorithms can track the time of flight of photons and measure how many return to a SPAD. The inverse problem of estimating the causes from the effect, in particular the depth from the SPAD measurement, is much more difficult.

The challenges of the inverse problem are:

1. Angular information is lost by summing light from all directions onto the same SPAD pixel.

2. Our single-pixel measurement is dependent on the unknown reflectivity and specularity of imaged objects.

3. The SPAD measurement also depends on the (unknown) orientation of objects in the scene w.r.t. to the transceiver.

Nonetheless, there are some indications that the SPAD measurement contains more information about the scene than is apparent at first glance. We identify the following sources of exploitable information:

1. SPAD bins are not disjoint, as objects have finite depth and so occupy multiple depth bins. Prior research has shown that different people have distinct time traces (features) in a single-pixel SPAD signal, allowing the classification of people using a single-pixel SPAD [159].

2. A stationary background environment produces a static background SPAD measurement. If we are only interested in monitoring variations in the scene, we can ignore the background and focus only on the foreground objects.

3. We note that foreground objects in the scene occlude the background. Hence, there is information about the foreground not only in their own SPAD spikes but also in the drop in background signal which they cause.

4. Light does not only travel from the laser to the scene and directly back to the SPAD. Objects also reflect light at one another, creating multipath effects. Prior research has shown that the positions of a set of points can be reconstructed up to Euclidean congruence, by exploiting secondary reflections between the points [160].[1]

This information can be used to generate priors to regularise the inverse retrieval task. There are many options here. A possibility is to use the example-based super-resolution approach (see Sec. 5.1.3) of building a dictionary of scene-histogram pairs. One could then identify potential scenes that caused the histogram by matching the measurement to training histograms in the dictionary and outputting their parent scenes as the 3D image estimate.

This approach suffers from the same degeneracy problem as super-resolution, namely, multiple scenes can create the same histogram. However, the problem is orders of magnitude worse in our case than in typical 2x, 4x or 8x super-resolution, since the dimensionality of the scene is typically orders of magnitude higher than that of the SPAD histogram. If instead, we use a

[1] Note that this algorithm is limited to simple environments. Further, reconstruction up to Euclidean congruence means we know the distances between our points and the angles between them, but not where they are.

histogram patch approach (where we match a temporal histogram patch to some ToF patch, e.g. a given object) then even if we can guess the object correctly, we do not know where it sits along the spherical shell of possible positions. We must therefore further limit the reconstruction scenario: instead of aiming to reconstruct any natural scene, we may wish to reconstruct only a very limited subset of natural scenes.

We use a neural network (NN) to learn to exploit the complex (background and object-dependent) information in a single-pixel measurement of a flash-illuminated scene. This ML approach is very similar to the example-based paradigm considered above; we alleviate the degeneracy problem by imaging a fixed background scene and training and testing on a very limited set of foreground objects - just people and fixed objects moving around in a fixed room. Unlike an IR approach, once trained, the NN can be used to predict depth images from temporal histograms straight away, without live optimisation of some cost function.

We've published two works on this topic. The first one establishes our imaging paradigm and focuses on the single-reflection case [3]. The second one improves the ML model and focuses on the effect of multipath information [4].

6.2 Single path echoes

Setup

All of our experimental setups consisted of a flash illumination source and a single-pixel bucket detector, as well as a conventional ToF imaging camera. Fig. 6.2 shows a general sketch of our imaging scheme. We illuminate the scene and bucket detect reflected light onto a single time-resolved pixel, giving a temporal histogram. Meanwhile, a conventional depth camera gives us a ground-truth ToF frame of the scene. The first step is to gather training data, namely synchronised ToF images and temporal histograms while the scene changes. On this dataset, we train a neural network to map temporal histograms to ToF frames. During deployment, the ToF camera is no longer needed, as the trained neural network can predict depth maps from the temporal histograms.

The same general setup principle is used for LiDAR, SODAR and RADAR. We will individually cover each setup in detail.

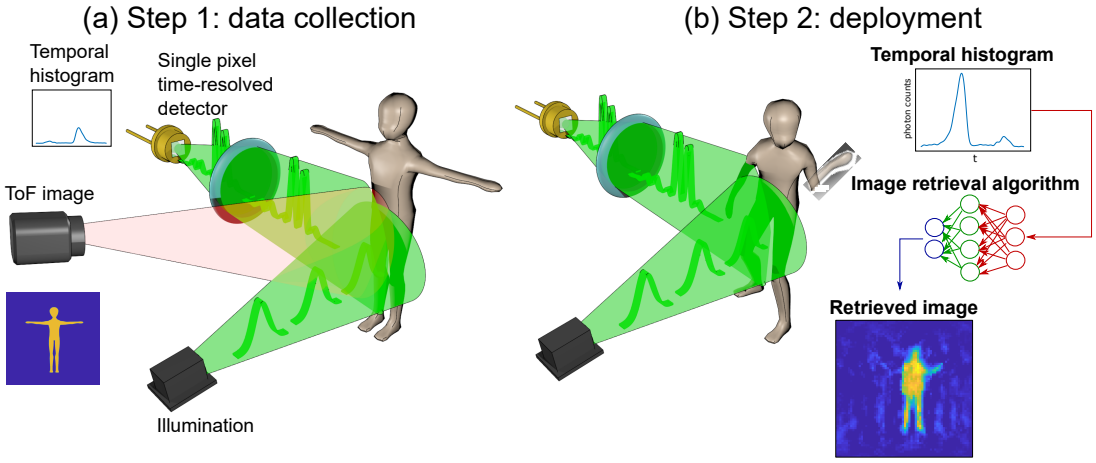

Figure 6.2: Schematic of our general imaging setup. **(a)** During data collection, we illuminate the scene with some source (e.g. a pulsed laser) and gather synchronised ToF images and single-pixel bucket detector (e.g. SPAD) temporal histograms of the same scene. We train a NN to predict the ToF images from the temporal histograms. **(b)** Once the NN is trained, we can deploy the system and predict ToF directly from the temporal histograms.

Simulation

To analyse specific properties of our approach, we first tested it in a controlled, simulated test environment. Our simulation models a $20m^3$ scene. In this scene, we place one of 10 silhouettes of humans of different sizes in different poses (shown in Fig. 6.3(a)), or its mirror image, and moved the silhouette around the scene in (x, z), where x is left-right position and z is depth. To simulate how farther objects appear smaller to the observer, we scaled the silhouettes by factor S based on their coordinates (x_i, y_i, z_i):

$$S = \frac{2}{\sqrt{x_i^2 + y_i^2 + z_i^2}}$$

The silhouettes were at constant height y to simulate moving along a surface. For the 10 silhouettes, we had 2 mirror images, 10 z-positions and 20 x-positions, giving a total of $10 \times 2 \times 10 \times 20 = 4000$ scenes; 400 per silhouette. For each voxel $r_i = (x_i, y_i, z_i)$ in the 3D scene, we calculate time of flight $t_i = (2c)^{-1}\sqrt{x_i^2 + y_i^2 + z_i^2}$, and thus project the scene onto a 720×720 ToF image. From this ToF image, we create a temporal histogram of the scene by flattening the image into a $720 \times 720 = 518400$ element array, and creating a histogram based on their time of flight values. The histograms had 8000 uniformly distributed bins between 13 and 31.1ns. Assuming uniform surfaces with uniform scattering properties and normals pointing towards the camera, and uniform illumination, the number of photons n_i reflected from a point in the scene at point r_i is inversely proportional to the point's distance to the power of 4:

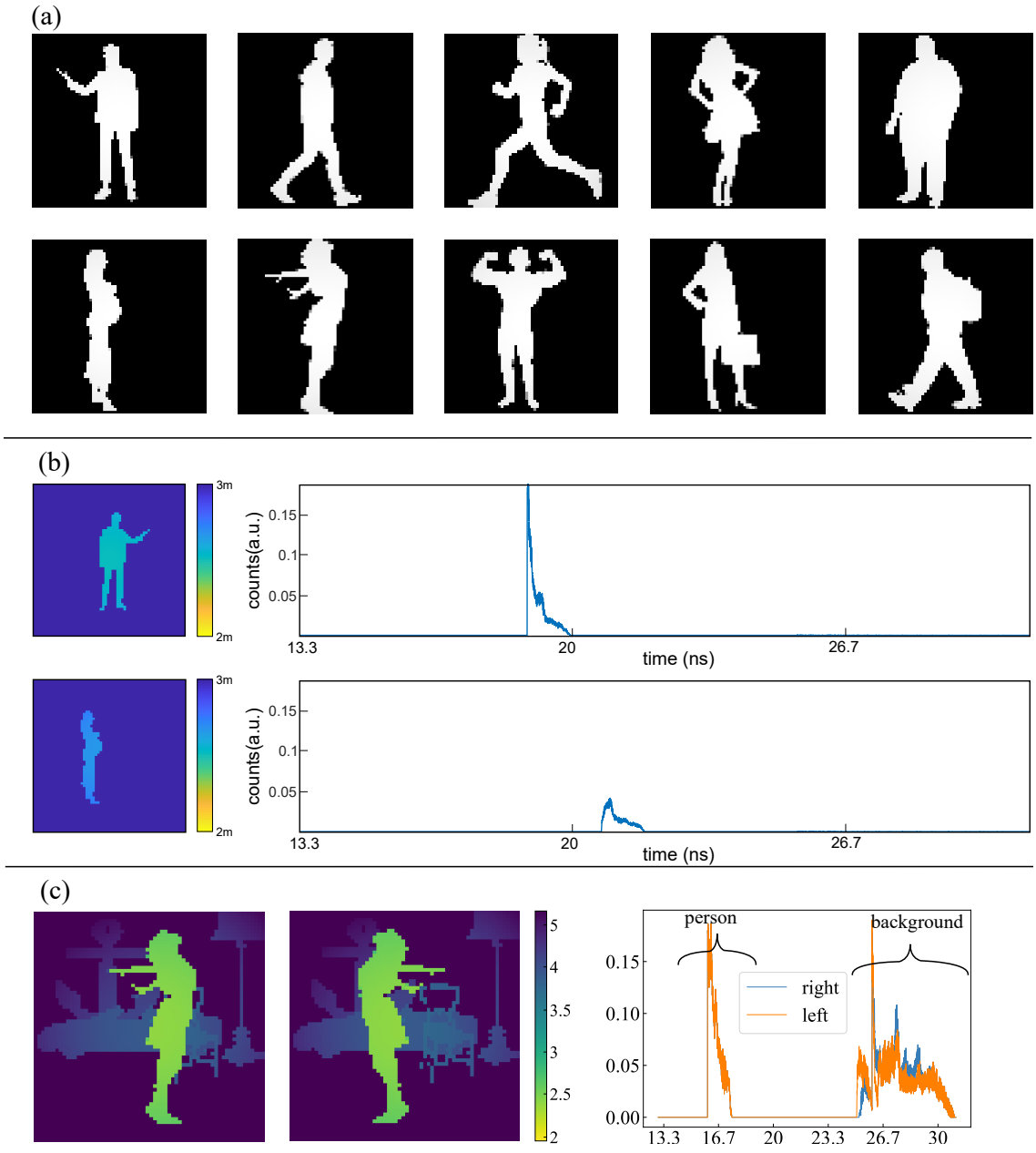

Figure 6.3: (a) The 10 silhouettes used to generate synthetic data. (b) Depth image - temporal histogram pairs. (c) Illustration of how the presence of a static background breaks spherical symmetry in the scene. A person and her mirror image are shown in a fixed room. The scenes have the same signal attributed to the person, but they have different parts of the background occluded.

$$n_i \sim \frac{P}{||r_i||^2}$$

where P_0 is the laser power. We apply this scaling to the histogram. The depth image is rescaled with 1st order spline interpolation to $64x64$. We assume the IRF of the system is only the binning operation, with bin width $2.3ps$. In other words, the basic simulation is noise free.

This process is used to generate pairs of ToF images and temporal histograms, shown in Fig. 6.3(b). The test sets were generated with the same silhouettes, just at different positions; consequently, all of our 'image reconstructions' could be interpreted as a joint (x, z) position regression, and 20-class (10 silhouettes facing either left or right) classification task. Reconstruction of the exact human form just arises from overfitting on the training set.

Background. We created a separate dataset with static background objects. These background objects are planar profiles and are simulated in the exact same way as the human silhouettes. The background consisted of a variety of objects (a chair, lamp, car and anchor), giving a complex histogram trace that the foreground objects partially occlude. An example of an image with and without the background is shown in 6.3(c); we also show the temporal histogram of the prior.

Neural network

The neural network architecture was a feedforward, fully-connected network with 3 hidden layers. It is trained in a supervised manner, end-to-end, on an $(8000, 1)$ input temporal histogram, and $64 \times 64 = 4096$ element flatten ToF image output. The hidden layers have 1024, 512 and 256 nodes. Each hidden layer has a tanh activation function. The loss function was mean-squared error.

Our choice of algorithm is motivated by the physics of the problem: any of the histogram bins can contain a signal from any ToF pixel, hence we want each output pixel to have the entire input in its receptive field[2]. Hence, fully connected networks are a sensible choice. Adding multiple hidden layers instead of just one helps the network to capture non-linearities and complex mappings from the input to the output.

Impact of training set size

In ML-based image reconstruction, a key aspect is to train the network on a large enough dataset to cover the distribution of expected testing images. Image classification algorithms, for example, are often trained on ImageNet 21k, which contains over 14 million images with corresponding class labels. Smaller algorithms are often trained on databases such as MNIST digits, which contains 60,000 images of handwritten digits (plus 10,000 for testing) or CIFAR10, which contains 50,000 training and 10,000 test images.

[2]Inspired by sensing, the term receptive field refers to the input area, or set of input nodes, that can drive a response in a node in a later layer's node. In a fully-connected neural network, every input node is connected to every subsequent node.

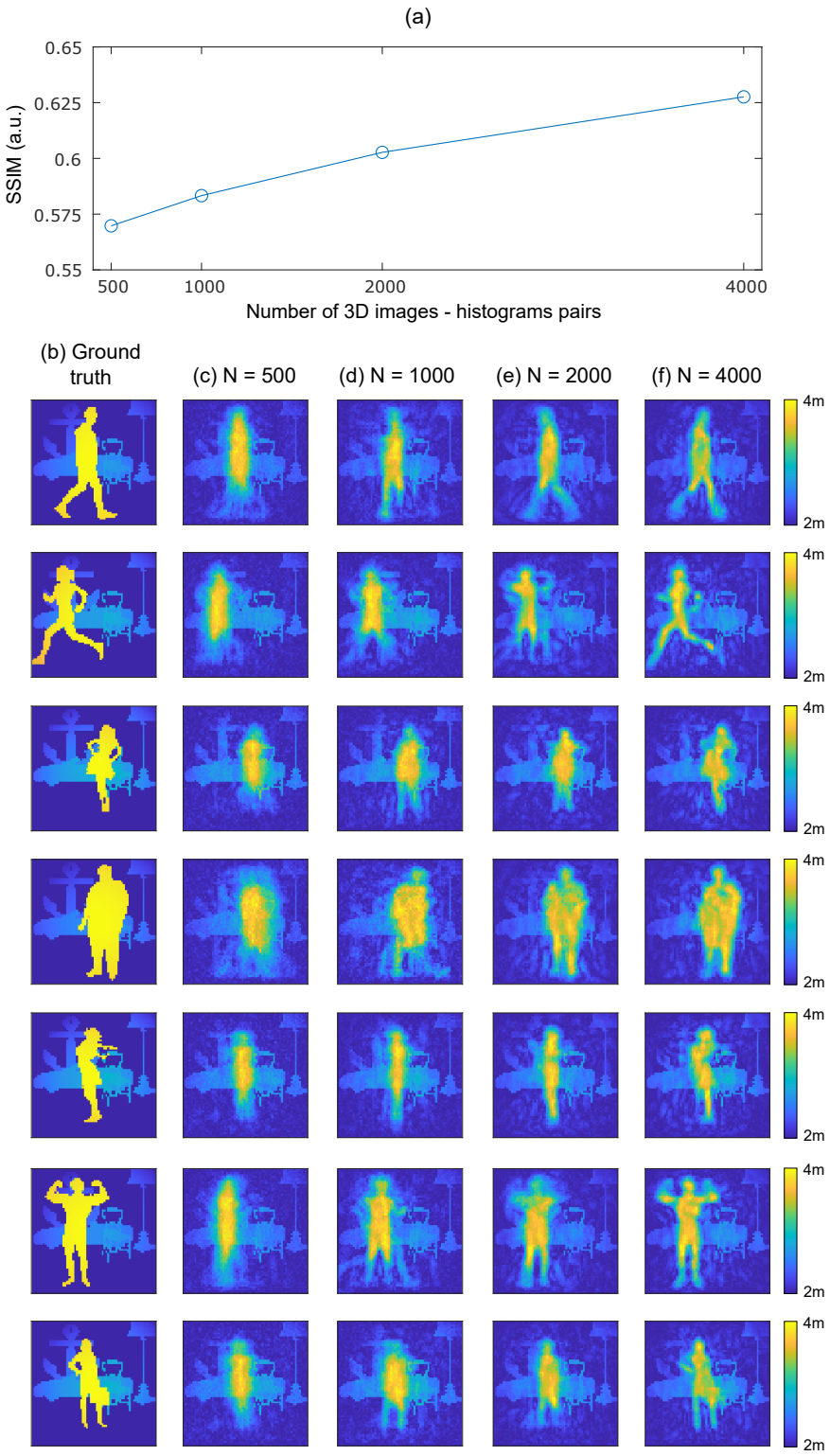

Figure 6.4: Reconstruction quality of the neural network, as a function of training set size. **(a)** We plot the SSIM between reconstructions and test images, averaged over 200 test pairs, for different sizes of the training data set. **(b)** Ground truth test images. **(c-f)** Corresponding reconstructions obtained from training data sets containing **(c)** N=500, **(d)** N=1000, **(e)** N=2000, and (f)N=4000 temporal histogram - ToF image pairs.

Our imaging paradigm would require an extensive dataset to approximate true imaging, even if constrained to a room. However, in this simple simulation, where the set of moving objects in the scene is limited and is the same for the training and test sets, our model can get away with far less training data. To see how many training examples we need exactly, we experimented with training our model on training sets of size 500, 1000, 2000, and 4000.

We assess the quality of reconstructions with structural similarity index measure (SSIM) - see Sec. 5.1.6 for details on this metric. The results are shown in Fig. 6.4. Based on these results, we used 4000 training pairs for the rest of our experiments, as reconstruction quality was deemed sufficient. Whilst these numbers are simulation-specific, we make a few observations. We observe that the smaller the training set size, the less recognisable the reconstructed silhouette. With 500 samples, the model could only reconstruct blobs, but it managed to at least position them correctly in the scene. We also observe that the model was always able to reconstruct the background, which makes sense since the background was static. As mentioned earlier, our reconstruction task is approximately a joint (x, z) position regression and 20-class classification task. Based on our results, we conclude the (x, z) regression task is the easier of the two.

Impact of background

To demonstrate the necessity of background scenes, we performed simulations with a complex, static background and without one. Fig. 6.5 shows our results. Without the background, the reconstructed shapes are less recognisable and also the model cannot decide if the silhouette faces left or right. Hence, our results show that using the background of the scenes is paramount for classifying/identifying, as well as locating foreground objects.

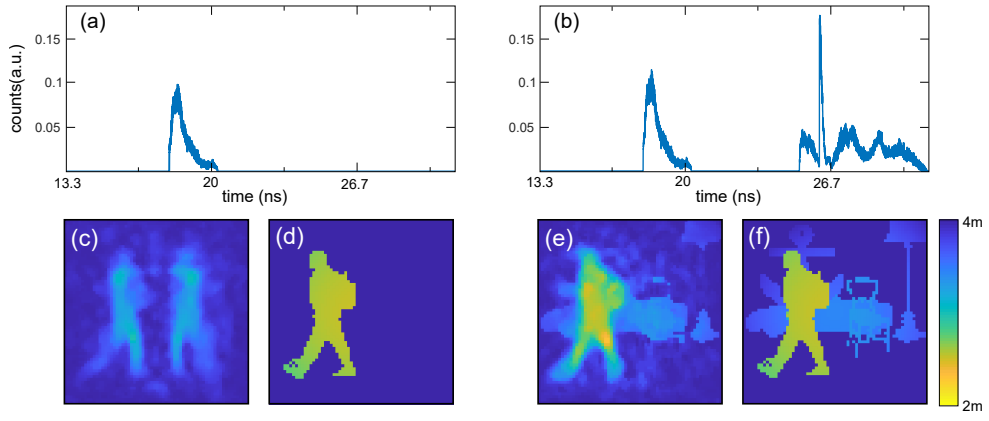

Figure 6.5: We demonstrate the impact of a static background on reconstructions. **(a)** Temporal histogram, with **(c-d)** the corresponding reconstructed ToF image and ground truth ToF image. We note that the model cannot decide the orientation of the silhouette. **(b),(e),(f)** The same data is shown in the presence of a background.

This makes sense since both orientations give the same signal; as we noted earlier, any point on the surface of a spherical shell around the transceiver will contribute to the same histogram peak, so spherical symmetries of reconstructions are equally likely. Even if we remove unlikely orientations of the object (such as having a person be upside down, or floating sideways in the air), left-right symmetries will persist. However, as we had shown in Fig. 6.3(c), the symmetries of the foreground object are broken by it occluding different sections of the background.

Impact of noise

Our simulations generate noise-free data. This is a liability since experimental systems always have some form of measurement noise, whether we consider the shot noise of the signal itself, or the thermal noise of the detection process. We must therefore investigate the effect of noise on our reconstruction ability.

For this, we simulated 3 different test noise levels and observed the effects on reconstruction quality. The model was still the one trained on noisy free data. With histograms x normalised between 0 and 1, our 3 noisy levels were:

1. $\hat{x} \sim \mathcal{P}(1000x) + \mathcal{N}(0,1) \implies (\text{PSNR} = 30\ \text{dB})$
2. $\hat{x} \sim \mathcal{P}(100x) + \mathcal{N}(0,1) \implies (\text{PSNR} = 20\ \text{dB})$
3. $\hat{x} \sim \mathcal{P}(10x) + \mathcal{N}(0,1) \implies (\text{PSNR} = 9.6\ \text{dB})$

Our results are shown in Fig. 6.6. We obtain good reconstructions for noise levels 1 and 2, and poor reconstructions for noise level 3. In the latter, we note that the quantisation error introduced by having the histogram be Poisson distributed was significant. Overall, our method seems reasonably robust to the effects of noise. In retrospect, our noise robustness might be because our measurement vector is 8000 elements long, and the histograms represent a much smaller dimensional information distribution (x, z position and left-right orientation of 10 silhouettes). Consequently, our input bins contain redundant information/the intrinsic dimension of the histograms is lower than its actual length. Similarly to exploitation of redundancy for noise-free communication over noisy channels [161], our network might be averaging over redundant bins to deal with noise. As such, this result might not hold if our measurement vector were shorter.

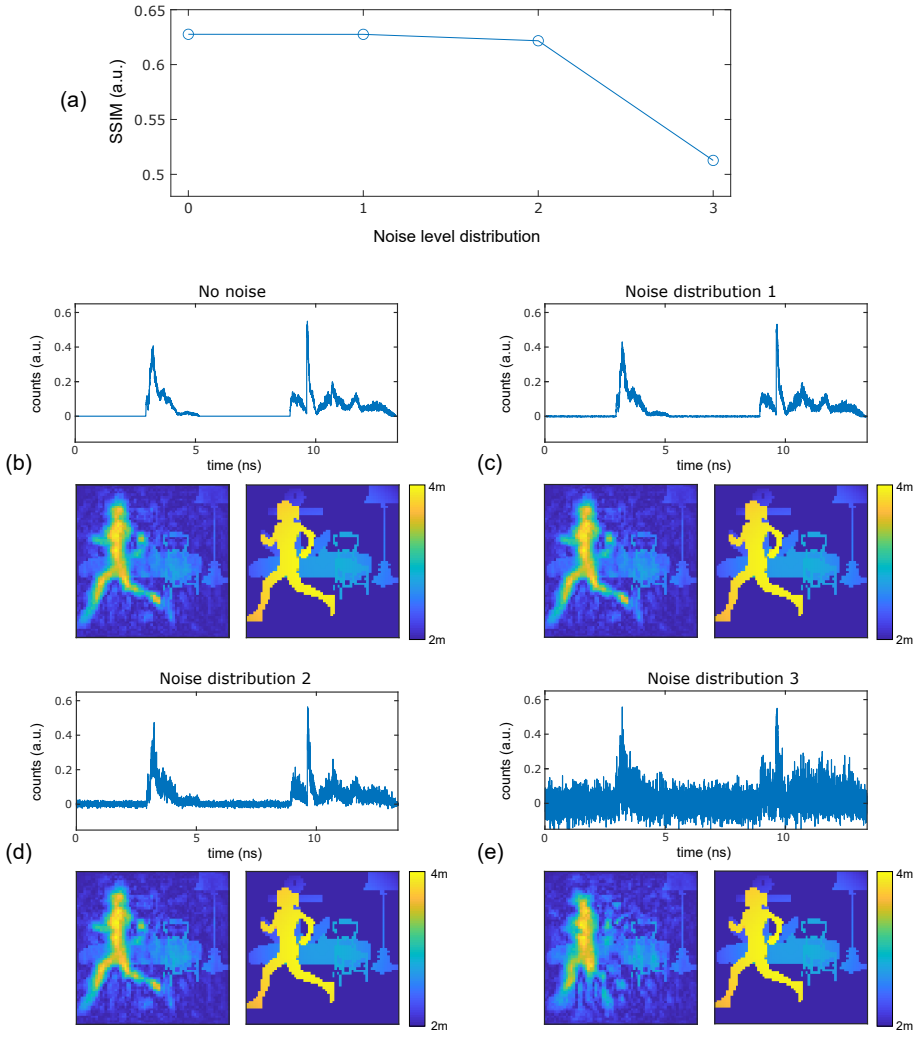

Figure 6.6: ANN reconstruction quality with noisy temporal histograms. **(a)** SSIM between test images and their corresponding reconstructions, averaged over 200 test pairs, as a function of the noise level. 'Noise level distribution 0' refers to the noise-free simulated data, while noise levels 1-3 are detailed in the text, and refer to PSNRs of 30, 20, and 9.6 dB, respectively. **(b-e)** For various noise levels, we plot examples of the same temporal histogram (top), the reconstruction from said histogram (bottom left), and the ground truth (bottom right).

Impact of IRF

We investigate how the reconstructed images are affected by the instrument response function of the simulated imaging system. This is to account for the finite IRF of any imaging system, introduced by various effects such as laser pulse width illuminating the scene, or timing jitter in the detector. Our baseline simulation is IRF-free in the sense that the only temporal degradation of the system is caused by sampling it into 2.3ps wide bins. This can also be interpreted as a best-case IRF width of 2.3ps. We can generate synthetic samples with larger IRFs by convolving this measured clean signal h with some IRF. We choose our tested IRFs to be a Gaussian G of width Δt.

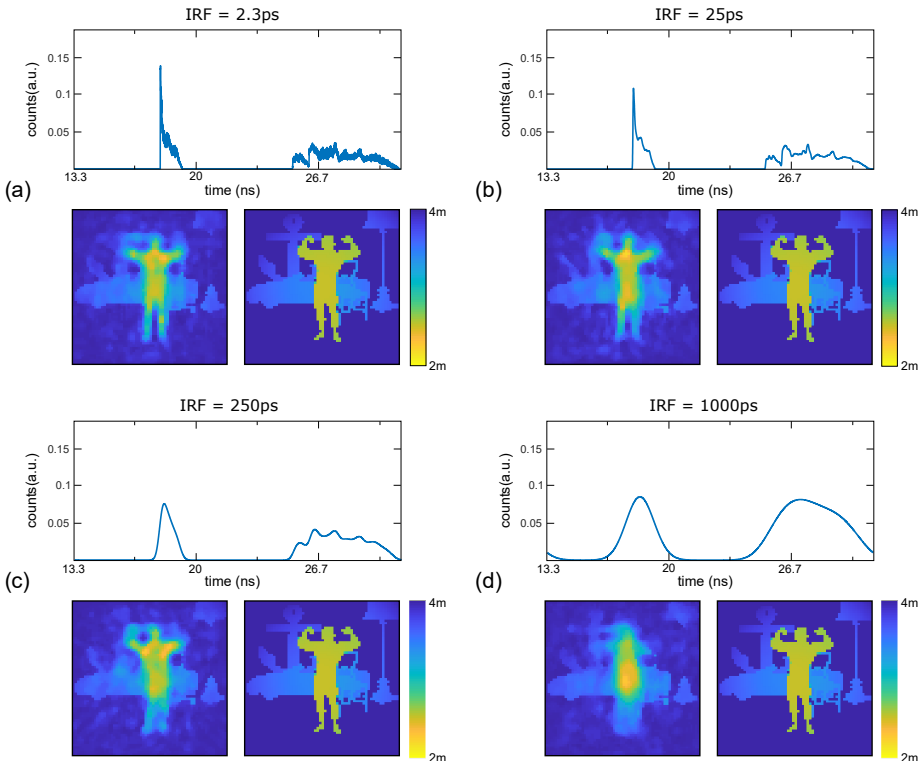

Figure 6.7: ANN reconstruction quality with varying IRF. We show temporal histogram, reconstructed depth image and ground truth depth image triplets for IRFs of width **(a)** 2.3ps (i.e. the noise-free simulation data), **(b)** 25ps **(c)** 250ps and **(d)** 1000ps.

Then, our simulation of a signal $\hat{h}$ obtained by a system with IRF Δt becomes

$$\hat{h} = h * G$$
$$\text{where } G = \exp\left(\frac{-t^2}{\Delta t^2}\right)$$

We retrain the ANN on pairs of $\hat{h}$ and ToF training images, convolved with various IRFs, and predict the corresponding test $\hat{h}$ set. We show results in 6.7. Our reconstructions worsen with IRF but are recognisable up to 250ps, which is a reasonable IRF width for a temporal detector such as a SPAD or PMT.

Impact of scene reflectivity

Next, we explored how the scene reflectivity changes our reconstruction quality. For this, we generated new synthetic data, of scenes with non-uniform foreground object albedos. The reflectivity of the silhouettes was quantified with respect to the background, denoted via $R = r_{silhouettes}/r_{background}$, where r_x is the absolute reflectivity of x.

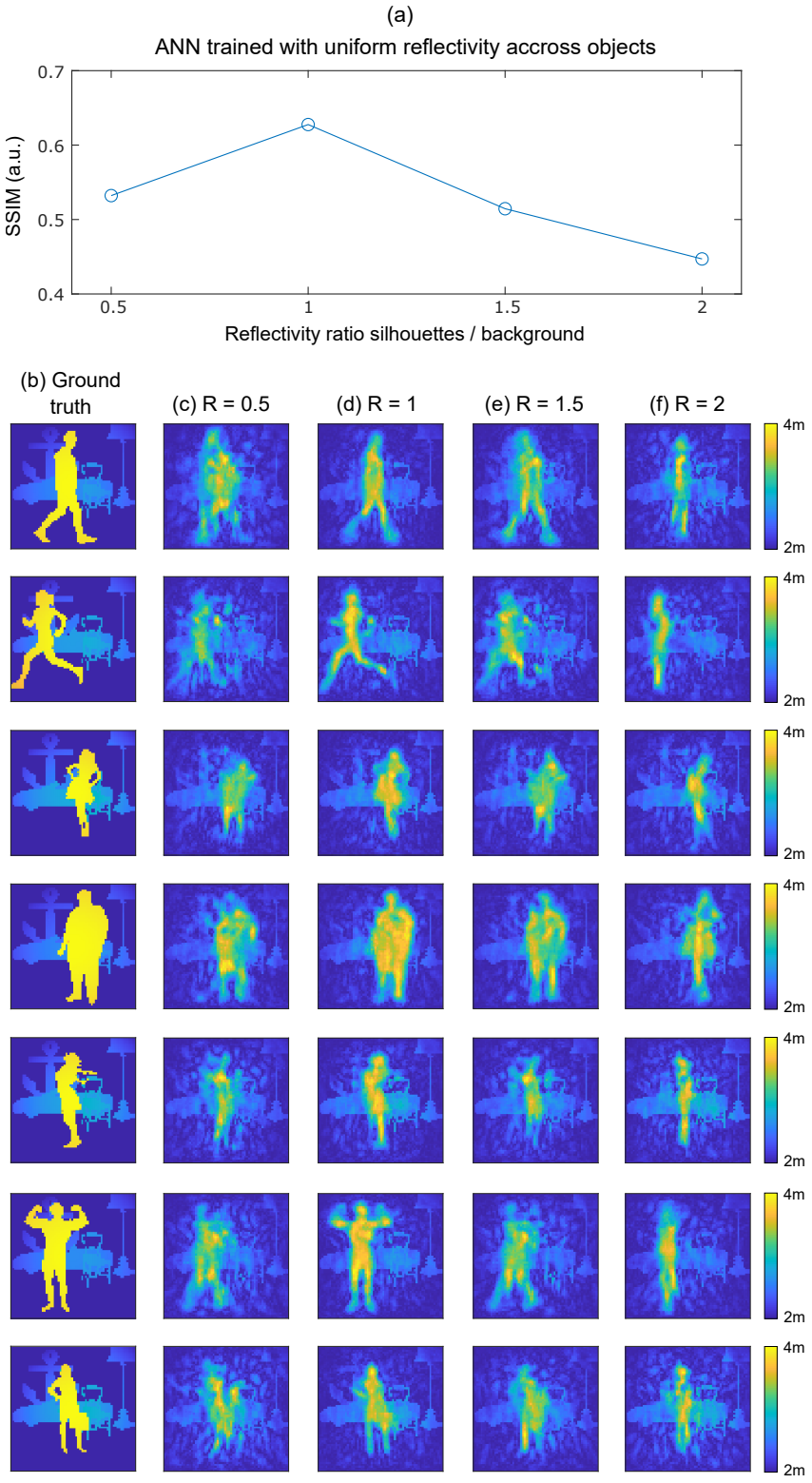

Figure 6.8: We show results trained with scenes of reflectivity R = 1, and tested on scenes with reflectivity R = 0.5, 1, 1.5, 2. In other words, the reflectivity of the silhouette is varied in the test set only. **(a)** SSIM indicates that the best reconstructions are obtained when the test set R matches the training set R, i.e. R = 1. **(b-f)** We show examples of the ground truth and reconstructed ToF images with varying R.

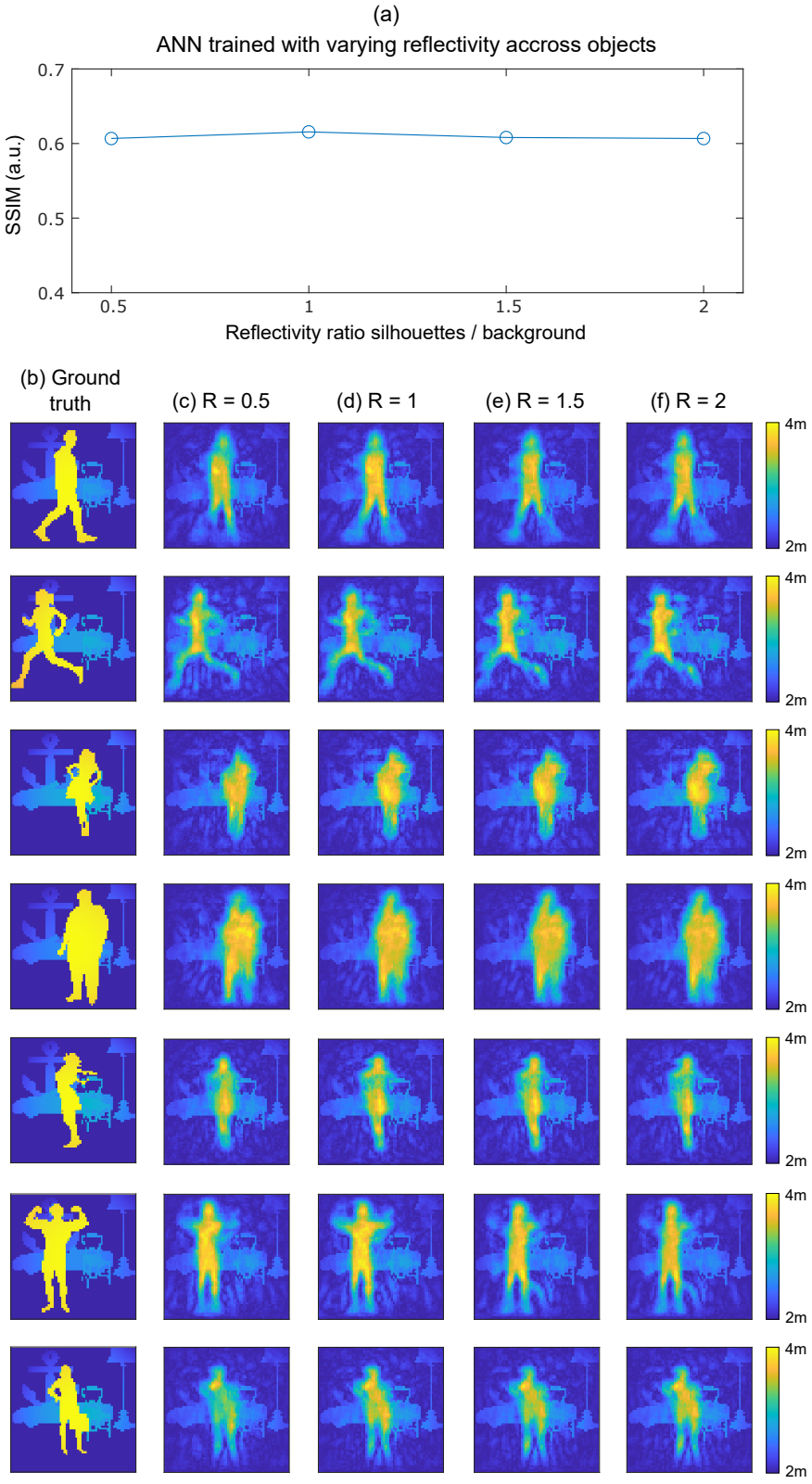

Figure 6.9: We show results trained and tested on scenes of reflectivity R = 0.5, 1, 1.5, 2. **(a-f)** Test set reconstruction quality is approximately uniform as a function of reflectivity, showing that the model generalises.

In our first test, we trained an ANN on silhouettes with R = 1 and evaluated the reconstruction quality of test histograms from scenes with R = 0.5, 1.0, 1.5, and 2.0. Results are shown in Fig. 6.8. We see that the x position of the silhouette is reconstructed correctly independently of R, but its shape is not. This makes sense since the NN can infer position from the occluded background, which does not depend on the reflectivity of the silhouette. In our second test, we varied the reflectivity of silhouettes in the training sets as $R \in 0.5, 1.0, 1.5, 2.0$, and evaluated test examples of varying reflectivity in the same ranges. This time, we obtain much more robust reconstructions; see Fig. 6.9 for results.

In conclusion, our imaging scheme requires training on objects with varying reflectivities to generalise to diverse scenes. Our results also highlight that the material of reflective surfaces is an important factor in our scheme; since all points at the same distance from the camera are summed together, the material with the highest reflectivity dominates the signal in a given temporal histogram bin.

Experiments

After numerically investigating the conceptual performance and properties of our flash-illuminated single-pixel imaging paradigm, we must confirm that it is applicable to a real-world experimental setting. For this, we created two experiments, one using LiDAR and one using RADAR.[3]

[3] I performed the numerical part of this study, so I will only cover experimental results briefly. They are covered in more detail in [3].

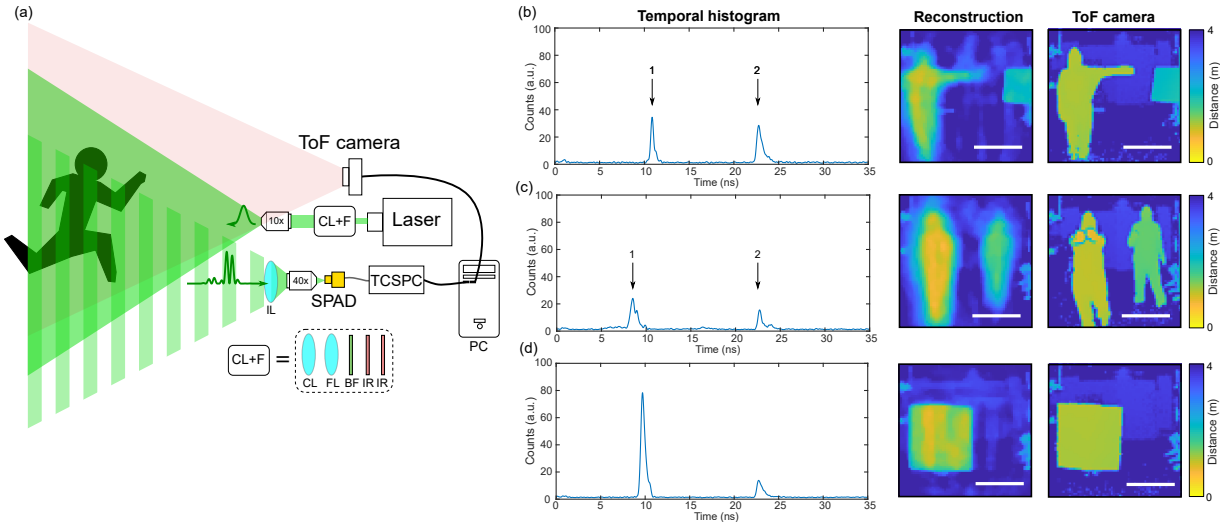

Figure 6.10: **(a)** Schematic of the LiDAR setup. **(b-d)** Results with a single figure in the foreground, as well as multiple people, and a box. Like with synthetic data, the x-y location of the reconstructed figures is consistently correct, whereas the outlines of our objects are less precise.

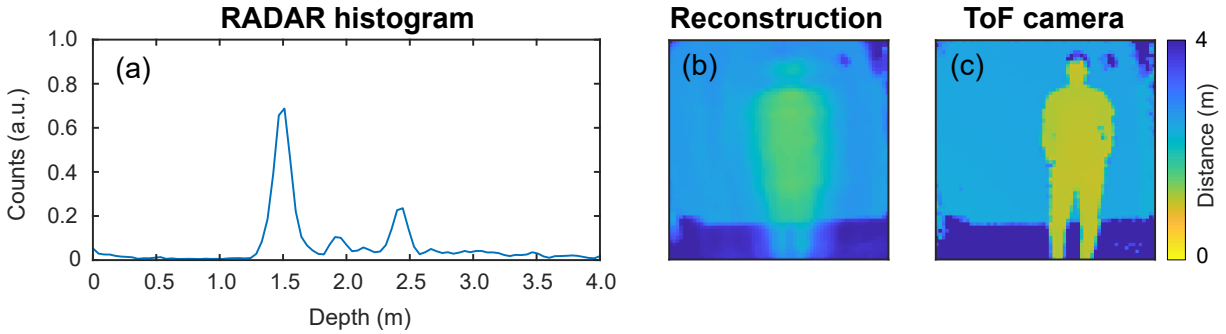

Figure 6.11: Reconstructions obtained from the time trace of a RADAR. The reconstruction quality is poorer than when using LiDAR.

LiDAR experiment

Our setup is shown in Fig. 6.10(a). The scene was flood-illuminated with a pulsed laser at 550±50 nm, with a pulse width of 75ps, using 250 mW of power expanded with a 10x microscope objective to flood-illuminate the scene, giving us an opening angle of 30°. A telescope made of 2 lenses, FL and CL, was used to fill the back aperture of the objective, whilst two infrared mirrors IR were used to filter out the laser pump. Light reflected from the scene was collected with a lens followed by a 40x objective, focusing light from a large aperture onto a single SPAD pixel operated in TCSPC. A ToF camera (flexx PMD technologies) with a maximum depth range of $[0.1-4]$ and a pixel resolution of 224×171 was synchronised with the SPAD to capture training data and ground truths for testing.

The ANN is then retrained with experimental data; Fig. 6.10(b-d) shows some results.

RADAR experiment

We performed a second experiment using an impulse RADAR transceiver operating at 7.29 GHz, using a similar setup as for LiDAR. After retraining the system with RADAR temporal histogram - ToF image pairs, we can feed new RADAR histograms into the network and estimate ToF. See Fig. 6.11 for an example reconstruction.

6.3 Multipath echoes

Our previous studies focused on direct or single-path reflections. In this section, we demonstrate that multiply reflected light, i.e. multipath echoes, can encode much richer details of the shape and position of the imaged scene than direct reflections. Multipath echoes also increase the likelihood of detecting back-reflected light from objects that are not directly facing the detector.

We provide proof of this in simulation and also find information-theoretic evidence that multipath echoes are key for encoding 3D information into a temporal measurement.

We then perform an experimental study that ascertains this claim. Our study focuses on RADAR and SODAR instead of LiDAR, because optical light reflects diffusely off of walls. Hence, each scattering event is a source of spherical waves, causing light intensity to scale as $1/r^2$ with propagation distance r between scattering events. Acoustic and mmWave radiation reflect more specularly (mirror-like) off of walls, so the rate of intensity decay per scattering event is much lower, and higher-order reflections still produce measurable signals.

Motivation

Multipath reflections between objects. Prior work has shown that multipath echoes encode the position of a set of objects relative to each other. Fig. 6.12(a) demonstrates a scenario in which two diffuse reflectors, p_1 and p_2 are illuminated by a pulsed laser, and back-reflected light is collected by a SPAD. The light follows 4 possible paths: laser-p_1-SPAD, laser-p_2-SPAD, laser-p_1-p_2-SPAD, laser-p_2-p_1-SPAD.[4] If p_1 and p_2 are at different distances, d_1 and d_2 respectively, from the laser and SPAD (which are roughly at the same spot), then the four ToF measurements corresponding to the 4 paths are: $t_1 = 2d_1/c$, $t_2 = 2d_2/c$, $t_{3a} = (d_1 + d_3 + d_2)/c$ and $t_{3b}(d_2 + d_3 + d_1)/c$, in the same order as the paths. Since the last two times-of-flight are identical, they combine to produce one intensity spike twice as large as either, giving us three times-of-flight.

Therefore, we can reconstruct the two points relative to one another, though we do not know their alignment (i.e. which one is farther than the other, are they side-by-side or one above the other, where they are within the illuminated region, etc.). We call this reconstruction correct up to Euclidean congruence, i.e. if we treat the reflective points as vertices and connect them with edges, the resulting shape is correct, but its orientation and position are unknown. In contrast, without the multipath reflection, we would only know the d_1 and d_2, not d_3. As we can see, the multipath echo constrained this simple task.

Multipath reflections combined with a fixed background. As we had shown previously (see Sec. 6.2.5), a static background contains information on the location of an object occluding this background. Let us consider the interaction of multipath echoes with such a static environment, to see if we can exploit multipath information to analytically localise an object. We show a 2D model in Fig. 6.12(b). We demonstrate that in this toy model, the multipath echoes, paired with a static background, remove the ill-posedness of our imaging task.

Let us consider an emitter and detector located at the origin of our coordinate system. The

[4] A negligible fraction of light will bounce multiple times between p_1 and p_2.

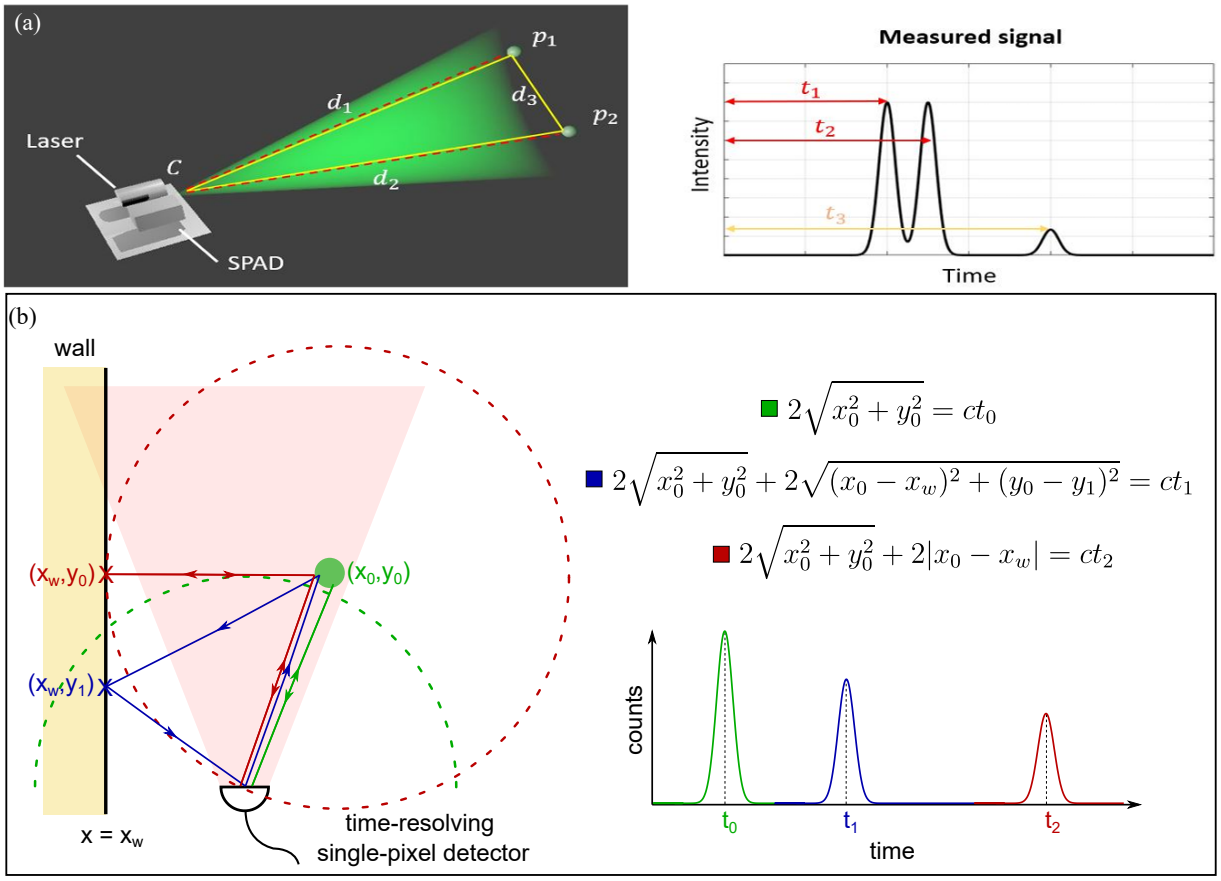

Figure 6.12: **(a)** Schematic of an imaging scheme that detects the relative position of two points with respect to one another. The multipath reflection creates a $3rd$ intensity spike, carrying information about the distance between the two points. Image adapted from [160]. (b) Illustration of a system illuminating a single point next to a fixed wall. By knowing the position of the wall, we can locate the object from the first 3 reflections. Image from [4].

emitter flash-illuminates the scene with pulsed light. We assume the detector receives any wave incident on it at any angle and retrieves ToF. Our scene consists of a mirror-like wall, described by the line $x = x_w$, and a diffuse object, modelled as a point-scatterer at position (x_0, y_0). The scatterer reflects light isotropically, i.e. in all directions. Light travels from the emitter back to the detector in 3 paths.

Firstly, it can directly reflect off the object, shown in green in Fig. 6.12(b). Using simple geometry, the ToF t_0 of this path is given by:

$$2\sqrt{x_0^2 + y_0^2} = ct_0, \tag{6.1}$$

Secondly, light can travel from the emitter to the object, then to the wall, and back to the detector (or the same path but in reverse, although in Fig. 6.12(b) the illumination is too narrow

for the reverse path). This is the 2-reflection path, shown in blue. Its time of flight t_1 obeys the following equation:

$$\sqrt{x_0^2 + y_0^2} + \sqrt{(x_w - x_0)^2 + (y_1 - y_0)^2} + \sqrt{x_w^2 + y_1^2} = ct_1, \tag{6.2}$$

Thirdly, light can travel from the emitter to the object, hit the wall, bounce back to the object, and reflect onto the detector; this is the red 3-reflection path. The corresponding ToF t_3 is:

$$2\sqrt{x_0^2 + y_0^2} + 2|x_0 - x_w| = ct_2, \tag{6.3}$$

In this instance, we see that we can substitute Eq. 6.1 into Eq. 6.3 to find $|x_0 - x_w|$. For known x_w, this gives us x_0, which is then used to find y_0, solving the system of equations. It is also possible to solve the system of equations using Eq. 6.1 and Eq. 6.2, though this is less straightforward.

In any case, we have evidence that the combination of a fixed environment and multipath information is a promising avenue for our imaging scheme. However, this toy model is very simplistic. Fig. 6.13 shows a more complex scenario of a room of mirrors. Direct reflections are shown in black, $2-4$ path in red, and $5-10$ path in cyan. It appears that incorporating higher-order echoes gives more information about the object's location within the room (e.g. we can tell that the object is standing on the floor, near the right-side wall), and high-order reflec-

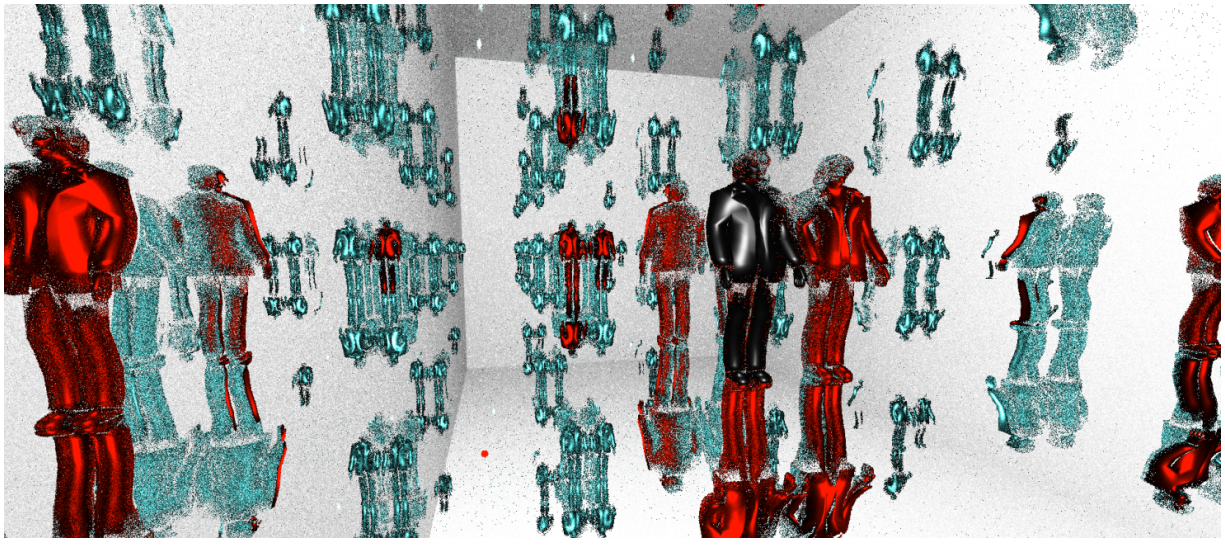

Figure 6.13: Illustration of multipath echoes off of an object (in this case, person) in a room of mirrors. The direct echoes are shown in black, echoes that scattered $2-4\times$ are shown in red, and $5-10$ path echoes in cyan.

tions show us multiple perspectives of the object, e.g we can see the subject's back. Of course, objects are typically not in a mirror-like environment, but instead in a lossy, diffuse environment. So the higher-order reflections are increasingly blurred and low intensity, rapidly approaching the noise floor of the detector. Analytical reconstruction of this ToF image from a single-pixel temporal histogram would be tedious, so we use ML instead, but multipath reflections seem like a promising way of constraining the otherwise ill-posed reconstruction problem.

Objective. Our objective is to analyse the impact of multipath reflections on the reconstruction quality of our flash-illuminated single-pixel imaging paradigm. For this, we must collect temporal histograms where we control the number of times light reflects off the scene.

A simple approach is to crop the time histogram at a certain point since the time of flight of multipath echoes is usually longer than that of direct reflections. However, light can still reflect between objects near the transceiver and return earlier than direct reflections of the background. Therefore, the cropping approach is not perfect.

A second approach is to limit the angle of the received light (i.e. to decrease the numerical aperture (NA) of the collection optics). For indoor applications, this can crop out light that reflects off objects to the floor or walls and returns to the receiver. Consequently, this approach simply reduces multipath contributions. Vice versa, if we block out central rays from large-NA systems, we can encourage the temporal histogram to collect multiply reflected data.

A third approach is to engineer an environment in which multipath echoes are reduced, or to image the object in the open with no other reflectors around. Such setups eliminate multipath contributions altogether. For RADAR, one may use an anechoic chamber; for LiDAR, a black-painted room; for acoustics, sound-absorbing materials.

None of these experimental approaches offers fine control over the number of reflections of light in the scene. So, we built a simulation instead.

Simulation

Our simulation is shown in Fig. 6.14(a). The scene consists of an object (a cuboid of size $1m \times 1m \times 5m$) moving via (x, z) translation (sliding along the floor) inside a static room (a larger, hollow cuboid of dimensions $4m \times 4m \times 7m$). The temporal histogram is generated via Monte Carlo ray tracing[5]. All surfaces were assumed to have reflectivity and specularity of 1, so they reflect any incident ray mirror-like.

[5]Monte Carlo ray tracing meaning that a ray's current state fully describes its next state.

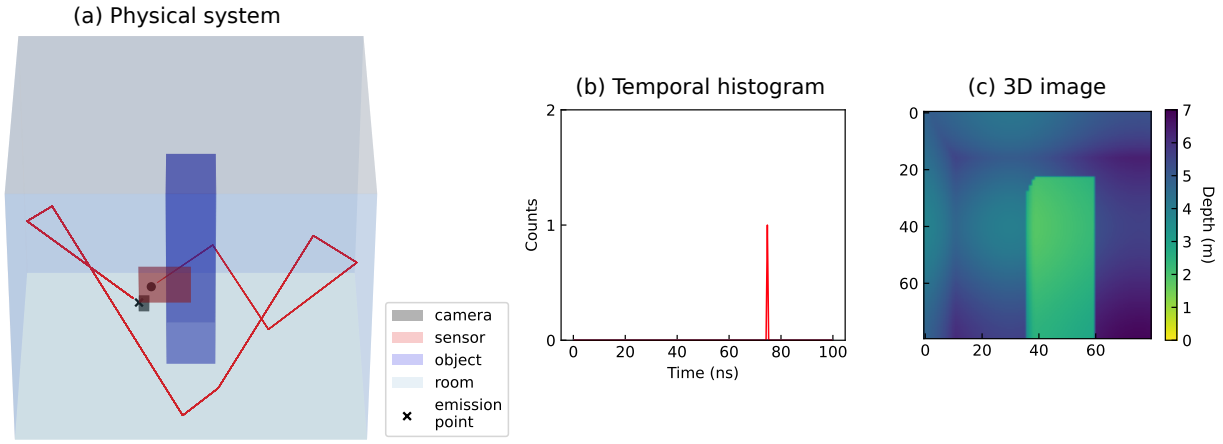

Figure 6.14: (a) Schematic of our simulation. Rays are emitted from a fixed point; one such ray is shown in red. It returns to the sensor's active area after some number of scattering events. (b) The detected ray creates a spike in the temporal histogram. (c) ToF image of the scene, made by the camera.

An emitter emits rays along a random k-vector, within azimuth and elevation angles $\theta \in [-67.5°, 67.5°]$ and $\phi \in [-67.5°, 67.5°]$. The rays travel a fixed distance in unit time, and we propagate the rays forward in discrete time instances (0.67ns). If the rays meet a wall during this instance, they change their k-vector to simulate specular scattering. We track the number of reflections that the rays undergo.

If the rays hit the sensor's active area during their travel path, their ToF is recorded based on the number of discrete time instances it took them to return to the detector. Later in post-processing, one can simulate an IRF by either convolving the arrival time with some function or adding a small, random delay to the arrival time. We chose the latter, randomly sampling from $\mathcal{N}(0.67, 0.33^2)$ ns.

We stop the simulation once a set number of time instances have passed. We chose 150 instances, corresponding to a maximum ToF of 100ns. We then repeat this process for 10000 rays and build a histogram of arrival times, binning the acquisitions into 200 bins of width 0.5ns. This is our single-pixel flash-illuminated temporal histogram. To generate the corresponding depth image, we scan across the scene from the camera position $[0.5, -1, 0.5]$, uniformly over azimuth $\theta \in [-60°, 100°]$ and elevation $\phi \in [-80°, 80°]$. The first reflection is returned at each scan position. We scan 80 points along either angle, giving an 80×80 image.

By moving the cuboid in the room along 2100 positions we generated a dataset of 2000 training and 100 testing histogram-image pairs. We made 10 such datasets, each with a set maximum number of allowed reflections.[6].

[6]In retrospect, a more time-efficient strategy would have been to save the number of reflections of each ray along with its ToF, and create the 10 datasets from one parent dataset by post-selecting rays that reflected the allowed

Machine learning

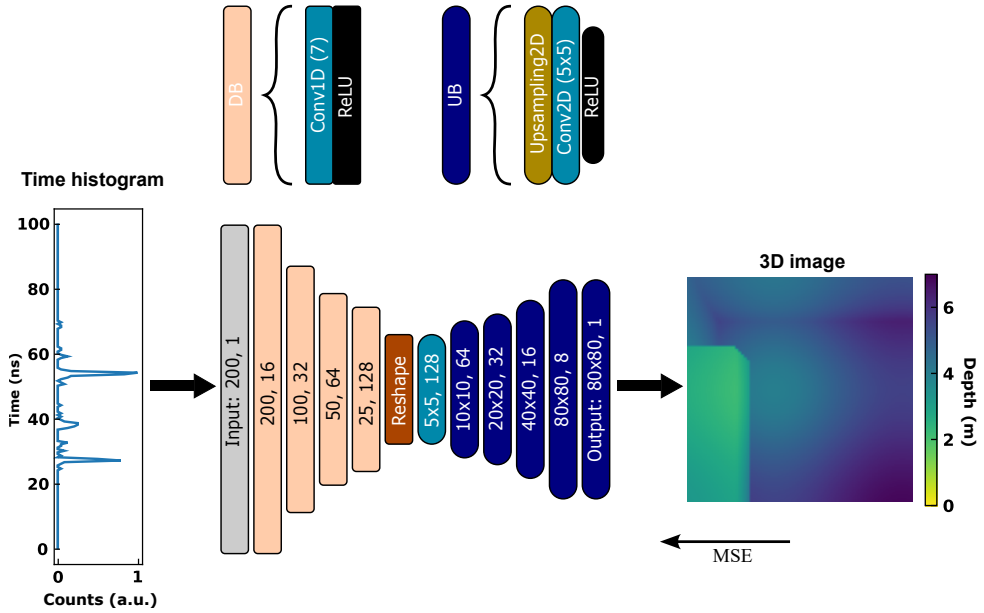

Figure 6.15: Schematic of the neural network. It is trained end-to-end in a supervised fashion on temporal histogram - ToF image pairs.

To map the histograms onto the ToF images, we used a convolutional autoencoder. It encodes the input histograms via 1D convolutions into a compressed representation and decodes this representation into depth images via stacked 2D upsampling and 2D convolutions. This encourages the network to extract features of the input that are important for the reconstruction process, and ignore noise or redundancy. These features form a space known as latent space.

The architecture is shown in Fig. 6.15. Training loss was mean squared error, and gradient descent was implemented via Adam on batches of 100 input-label pairs. We validated the number of training epochs on 200 pairs, observing that the ideal number of training epochs increases with the maximum number of scattering events. Thus, for single scattering event data, we trained for 110 epochs; for up-to-10 path data, for 350.

We used a similar architecture for experimental data, though we had to make adjustments for the varying sizes of the input histograms generated by the different single-pixel detectors, and the varying aspect ratios of the ToF cameras. For full details, please see the Supplemental Material of [4].

number of times.

Results

Reconstruction MSE. We retrained the neural network on each dataset and noted the reconstruction MSE on the corresponding test set. Results are shown in Fig. 6.16(a). Overall, MSE decreases as the number of multipath events is increased.

The local minima at 3 and 8 paths are likely caused by random chance. Most likely, they were due to the relatively small test sets, the fewness of rays used to sample each scene, and the random nature of the ray sampling. Barring these local minima, there is marked improvement in the 2-4 path regime. The rate of improvement decreases; eventually, the reconstruction quality plateaus after ~ 6 paths.

Information Theory. We analytically quantify the information gained from adding more reflections to our data using Shannon entropy and mutual information. In short: the information content of event x_i that occurs with probability $p(x_i)$ is defined as:

$$I(x_i) = \log_2(\frac{1}{p(x_i)}) = -\log_2(p(x_i))$$

This definition fulfils two criteria. Firstly, more probable events are less surprising, so their

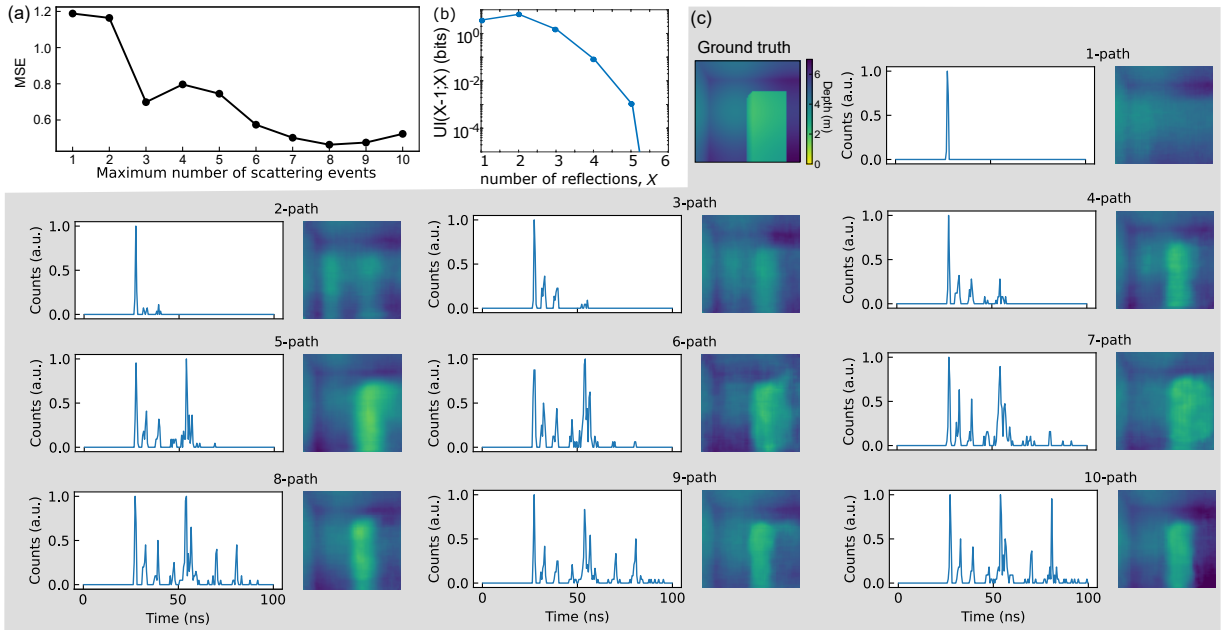

Figure 6.16: **(a)** Reconstruction mean squared error as a function of the maximum number of scattering events in the dataset. **(b)** We plot the gain in information in bits obtained between X maximum number of allowed scattering events and $X-1$. **(c)** We show example reconstructions of the same object from histograms with varying numbers of maximum scattering paths.

information content should be lower. Hence we want information content to be inversely proportional to probability.[7] Secondly, the negative logarithm ensures that events occurring with a probability of 1 have 0 information content, while events that happen with a probability approaching 0 have increasingly high information content.

Shannon entropy is the expected information associated with a variable, X, that can assume various values x_i:

$$H(X) = \sum_i p(x_i) \log_2(p(x_i))$$

For our study, we take a dataset of 2000 temporal histograms. The variable of interest is the binary shape of the histogram, i.e. we treat the histogram as a binary vector whose elements are 1 if the corresponding histogram bin has any photon counts, else 0. For a given dataset, this variable can assume at most 2000 values x_i if each histogram is unique; fewer if there are repeated binary histogram vectors in our dataset. The probability of each binary vector $p(x_i)$ is calculated from the number of times that vector occurs in the dataset, divided by 2000. Then we calculate the joint entropy for single-path histograms X and 2-path histograms Y, by assuming independence $p(x_i, y_j) = p(x_i)p(y_i)$:

$$H(X, Y) = -\sum_j \sum_i p(x_i, y_j) \log_2(p(x_i, y_j))$$

Uncorrelated information (UI), i.e. the information gained from including second reflections instead of just direct reflections, is then calculated as:

$$UI(X; Y) = H(X, Y) - H(X)$$

This is repeated for n-path and (n+1)-path datasets. Results are shown in Fig. 6.16(b). There is significant uncorrelated information gain from adding the 2^{nd} and 3^{rd} reflections, after which the gain rapidly decreases. This parallels our reconstruction results (examples graphed in 6.16(c)), which improve dramatically at first, then plateau.

We note that adding correlated information can still be valuable in the presence of noise. The noisy-channel coding theorem states that communicating a signal across a noisy medium demands redundancy in the message to ensure correct retrieval. Equivalently, in our case, if the histograms have noise, it makes sense to try to capture redundant (correlated) information since

[7] Consider a murder investigation in a forest; the detectives find a twig and a bloody teddy bear near the body and wish to decide which of these items to use as evidence and which to discard. The likelihood of finding a twig in a forest is very high, so it is unlikely to contain much information for the investigation; on the other hand, the likelihood of seeing a bloody teddy bear in the forest is low, so it is informative.

we can average over the noisy, degenerate copies to obtain the true underlying value with higher certainty.

In short, we have established that multipath reflections encode more information about the scene onto a flash-illuminated single-pixel measurement than direct reflections. We have also discovered that the first couple of scattering paths are the most informative, with the later ones being less and less important. This works in our favour: in the physical world, wave scattering is lossy, so we are unlikely to detect high-order reflections from the scene.

Next, we must demonstrate that multipath echoes are valuable not only in simulation, but also in experiments. For this, we build a RADAR and SODAR system and collect data from each.

RADAR

Our RADAR setup consists of an FMCW TI-AWR1642 transceiver module (for an introduction to FMCW ranging, see Sec. 2.2.3. The device has a range resolution of 4.4cm, a maximum unambiguous range of 9m, respectively, a vertical angular aperture of 20° and a horizontal angular aperture of 180° (-3 dB FWHM). We gathered 512 samples per chirp, giving temporal histograms of size 256. In parallel, a Basler camera acquires ToF images of dimensions 80×60.

The scene was an approximately $3 \times 4 \times 2.5m^3$ room, with a person walking around inside and the transceiver imaging from a fixed angle. We gathered training and testing datasets of 9000 and 865 images, respectively. We created estimate datasets containing direct, up-to-2, up-to-3, and full multipath data by cropping the ToF histograms at 25, 41.5, 58.1 and the full 75ns range. Results are shown in Fig. 6.17(a). As we increase the number of echoes in the temporal histograms, we get improved reconstruction MSE.

SODAR

Our SODAR setup uses a PC speaker (Logitech Z333 system, comprising two speakers and a subwoofer) as the transmitter and a PC microphone (Logitech C270 webcam microphone) as the receiver. The speakers emit pulsed sound, with a centre frequency of $5kHz$, bandwidth $1kHz$, duration of 50ms, and repetition rate of 10Hz. Return echoes are sampled by the microphone over 100ms at 192KHz sampling rate, and are Fourier filtered afterwards to post-select $4-6KHz$ echoes. Simultaneously, an Intel Realsense D435 stereovision camera gathered 64×64 depth images.

The scene consisted of a $7 \times 6 \times 2.5m^3$ room with a human subject doing exercises and the

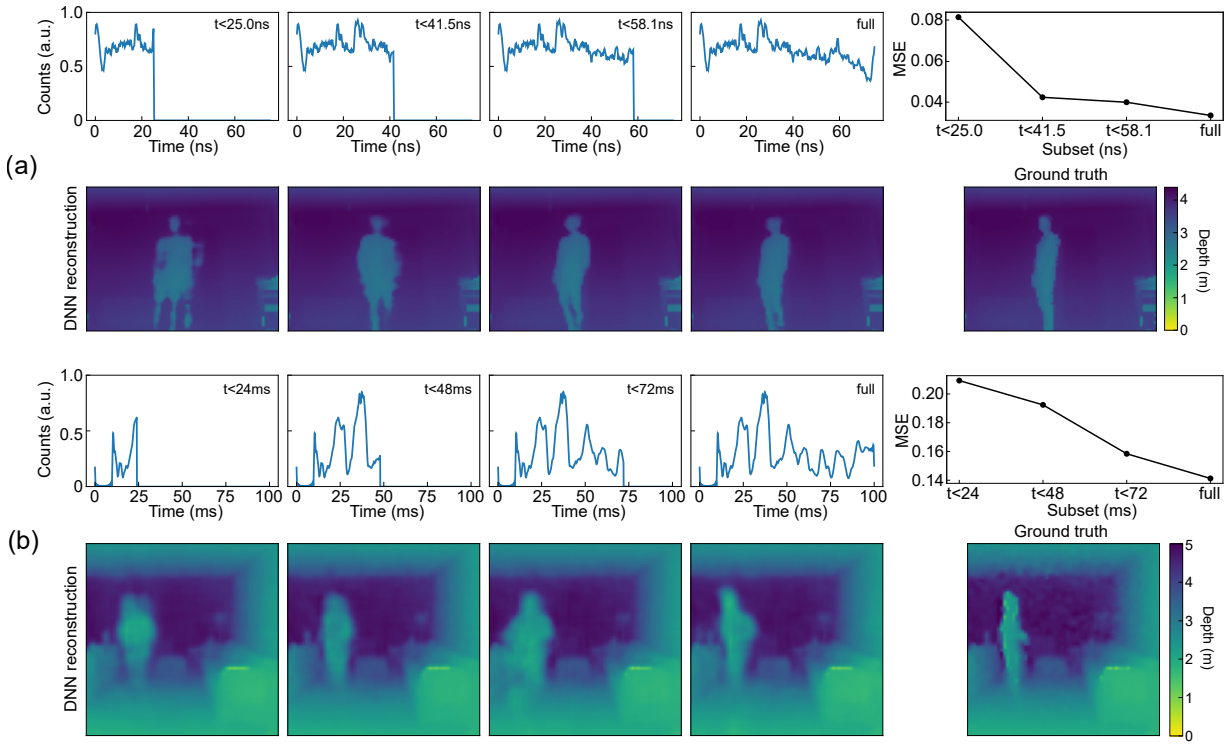

Figure 6.17: **(a)** Radar results. We show histograms and reconstructions from cropping the training and testing datasets at $t = 25, 41.5, 56.1$ and the full$75ns$, corresponding to the ground truth image shown on the right. Reconstruction MSE improves monotonically with the length of the temporal histogram. **(b)** The same results are shown for acoustic (SODAR) imaging.

acoustic/imaging system viewing from a fixed angle. 5000 training frames were acquired, and 500 testing frames. We create approximations of datasets with $1, \leq 2, \leq 3$, and full reflections by cropping the data at 24, 48, 72, and the full 100ns. We plot the results in Fig. 6.17(b). As with radar, we find that increasing the number of echoes in the dataset (by cropping later) improves reconstruction MSE.

6.4 Discussion

Conventional depth imaging uses either widefield illumination and structured detection (lens and a pixel array), or structured illumination scanning with single-pixel detection. In this chapter, we examined a new imaging paradigm using flash illumination and single-pixel detection. Thus, we lose explicit information on the spatial structure of the scene, but in return, we have the simplest hardware setup for both illumination and detection. This opens new avenues for turning cheap devices (such as a webcam audio system) into a pseudo-ToF-camera.

Our imaging paradigm is ill-posed, so we turn to machine learning to learn correlations be-

tween time traces and imaged scenes. We show that the presence of a background is vital, allowing our method to localise objects moving within the scene. We also investigate how the noise level, system IRF, and scene reflectivity affect reconstruction quality.

We find that system is resistant to random noise, though we note that this may partly be because we use simulated temporal histograms of 8000 elements. Our algorithm might exploit redundancy in this oversampled temporal data to handle independent noise; such behaviour might hold for SPAD or RADAR measurements, which typically give far fewer samples.

Regarding the system IRF, we observed decent performance up to an IRF of 250ps FWHM, which is experimentally achievable with TCSPC detectors. Our study of scene reflectivity showed that a neural network trained on objects of fixed reflectivity does not generalise well to scenes containing objects of different albedos. However, once we retrained the neural network with varying object reflectivities, it was able to generalise to diverse testing albedos. We conclude that our scheme requires a large training set to account for variability in the scene.

Our study about the effect of training set size showed it is easier to localise objects in 3D than to estimate their precise shape, at least given a fixed background. Therefore, our 3D imaging modality may be better suited to tracking tasks (where the exact shape of an object in a scene is either known or unimportant) than to full imaging.

We performed a follow-up study on the effect of multipath reflections on our imaging modality. We demonstrated that multipath echoes are key for getting good reconstructions. Whilst it is expected that more echoes give better reconstructions,[8] the surprising aspect of this study is just how marked the improvement was even in an experimental setting.

Now, let us consider the weaknesses of our imaging paradigm. Firstly, we always train and test on the same fixed scene. I performed some tests to see whether we could generalise performance to new, unseen scenes, but these studies were discouraging; a proper investigation seems implausible without gathering huge datasets. Moreover, the objects (typically human subjects) moving in the fixed scene all moved similarly in the training set and the test set, and there was a fixed number of people in the training and test scenes. To generalise beyond such scenes, a lot more data would be required.

Also, our imaging modality does not directly form images, but rather the neural network learns to associate temporal histograms with ToF images. Consequently, the fine details that are seen in the NN predictions are prone to overfitting.

[8] In our simulation, the amount of information from multipath echoes strictly increases as a function of the maximum number of allowed echoes, since the same number of rays are emitted regardless of the number of echoes, so more rays end up reaching the detector. Similarly, in our experiments, we measure the full multipath histogram and crop it down to approximate fewer reflections, consequently, the higher path data has strictly more information than the lower path data. Consequently, it is expected that multipath echoes outperform single-path ones.

Chapter 7.

Conclusions

Artificial intelligence (deep learning in particular) has revolutionised computational imaging and computer vision and continues to push the boundaries of these fields year by year. Text-to-image models such as Dall-E [162] and Stable Diffusion [163] have overhauled the world of 2D art generation. Related data formats follow suit, including 3D model generation via DreamFusion [164] or Magic3D [165], and video creation with tools like Synthesia [166]. Image-to-image models are also prevalent in style transfer, super-resolution, face-swapping, and many more applications.

However, deploying DL algorithms in the real world is challenging due to the black-box nature of machine learning and the ease of overfitting training data. Based on personal experience, real-world applications where current AI works well include depth estimation from RGB data [167] or detection of faces in camera feeds. Ones that have room for improvement include tasks like cell segmentation in microscope images [62], or super-resolution.

This thesis covers a few case studies of AI applied to computational time-resolved imaging, including data processing, modelling, uncertainty analysis and knowledge extraction. In particular, my work concentrated on:

1. Parameter estimation from noisy temporal data (chapter 3).
2. Extracting temporal information from spatial data (chapter 4) and vice versa (chapter 6).
3. Multi-modal data-fusion for super-resolution, SiSIFUS (chapter 5).

Chapter 3 focused on creating a fast but flexible fluorescence lifetime estimator using an artificial neural network. Fluorescence lifetime estimators generally fall into one of two categories: fit-free or fitting-based. Fit-free methods are computationally inexpensive, directly mapping measured fluorescence data onto lifetime. However, this makes them inflexible to uncertainty in the parameters of the decay model. Fitting-based schemes can allow such parameters to vary but at the cost of time-consuming iterative optimisation. Machine learning offers a unique opportunity of incorporating parameter flexibility into a fit-free estimator, by training a neural network on a diverse dataset of fluorescent decays. I explored the robustness of an ML-based fit-free lifetime estimator on both synthetic and experimental data, confirming that its performance is comparable to fitting-based techniques at a fraction of the computing time. These results may motivate future work on rapid yet robust fit-free parameter estimation from noisy data of various

modalities, for instance, characteristic residence time estimation from fluorescence correlation spectroscopy.

Chapter 4 demonstrated a new fluorescence lifetime imaging scheme, using an optical cavity to spread temporal information over the pixels of a time-resolved iCCD, and employing a time-integrating CMOS to capture the same FOV in parallel. The main algorithmic difficulty in this work was processing an image containing overlapping replicas of the sample. This task is comparable to blind, spatially varying point-spread-function estimation, since the optical cavity distributes intensity information from a given point onto other regions of the field of view, dependent on the unknown fluorescence lifetime at that point. We report on the forward model of this experiment, including precise calibration of the noise model. Using this, we establish the uncertainty lower-bound for our lifetime estimates using Bayesian inference, as a function of the total photon count incident on the iCCD. We also describe practices for aligning the data and demonstrate that a dilated convolutional neural network is well suited for mapping our data onto fluorescent lifetime. We validate the experiment and algorithm on data of fluorescent beads and test it on cellular samples, finding good performance.

The method is a novel implementation of single-shot fluorescence lifetime estimation. A direct follow-up study could exploit the effective parallel acquisition of multiple time gates to enable single-shot multi-exponential fluorescence lifetime estimation. Future works can also build on the optomechanical concepts in this report for different applications. They could also use similar algorithmic approaches for establishing parameter estimation uncertainty analytically,[1] or for processing overlapping images with deep learning.

Chapter 5 explores the main project of my PhD studies - super-resolution using the fusion of low-resolution time-resolved images and high-resolution time-integrated ones. This chapter starts with a review of existing single-image super-resolution (SISR) schemes, focusing on how example-based SISR works in both classical and deep learning environments. It then explores how fluorescence lifetime data differs from natural images (typically used to develop and benchmark SISR schemes) and how these differences hinder applying existing techniques to FLIM.

To overcome these challenges, we developed a novel multi-modal data-fusion-based super-resolution technique called single-sample image fusion upsampling (SiSIFUS). SiSIFUS is a method of finding local or global statistical correlations between a low-resolution image that we wish to upsample, and a high-resolution, easy-to-acquire image of a different modality. These statistical correlations come from a single measurement, thus our method works well even for individual samples with unique properties. We use learned perceptual image patch similarity (LPIPS) to benchmark our approach against interpolation, which it outperforms at high upsampling factors (8 to 16×).

[1] Note: our calculation of estimation uncertainty strongly parallels Fisher Information.

Although we only applied SiSIFUS to fluorescence data, it has the potential for super-resolving measurements from other modalities. A purely statistical method, this technique only requires that a different but correlated high-resolution image of the scene be formed in parallel. Potential target applications could include upsampling Brillouin images with HR intensity images for measuring the microscopic elasticity of samples; improving spectroscopy data with HR RGB images for sorting plastic in recycling plants; or super-resolving infrared images with traditional camera feeds for detecting atmospheric gas. By training on the sample of interest instead of a separate dataset, our scheme avoids out-of-distribution bias in the final prediction (a problem that traditional example-based methods face).

Future work on SiSIFUS could look at speeding up global morphological prior (GMP) generation - currently, a neural network is trained from scratch, which takes time. One possibility is pre-training the network on natural images offline, and then simply fine-tuning it on the sample of interest. Another weakness of global SiSIFUS is that our approach trains on HR patches of a fixed size. Thus, new schemes could be designed to make the method more robust to variations in the sizes of features in the input space. For instance, the LR FLIM and HR intensity images could be iteratively downsampled, and GMPs trained at each resolution. However, this would further increase processing time.

Chapter 6 demonstrates a novel depth imaging and ranging scheme, using flash illumination and single-pixel detection. In contrast to the previous chapters, which focused on algorithms, AI, and data processing, this work mainly presents novel physics and imaging concepts. Traditional LiDAR schemes use either structured detection (a lens and pixels) or structured illumination (scanning) to obtain spatial information. Our proposal does not explicitly capture the angle of arrival of light. Instead, it only relies on measuring the time-of-flight of received radiation. We show that this contains information about the spatial distribution of the scene, which can therefore be recovered using machine learning. In particular, we demonstrated two novel imaging insights. Firstly, we show that a fixed background allows locating an object based on the temporal trace of its shadow on the background. Secondly, we demonstrate that time-resolved multipath reflections contain significant spatial information about the scene, which can improve 3D reconstructions compared to direct reflections only.

Overall, this work introduced a seemingly counterintuitive imaging paradigm, whose properties make it unsuitable for some imaging applications, but well-suited for others. Currently, this paradigm is unfit for capturing high-contrast, high-resolution images of unconstrained scenes. Consequently, our method is unsuitable for general-purpose cameras. Instead, our paradigm is better for detecting small subsets of spatial scenes, such as a fixed room with people walking around inside. By limiting the application to such a subset instead of all potential scenes, gathering enough training data to allow a neural network to model the scene distribution becomes more

feasible. Therefore, our method can be used, for example, for security cameras. Further, since our technique does not use hardware for spatially encoding either the illumination or detection, it can use simple, compact, and low-power devices - such as the speaker/microphone of a laptop. Lastly, since our method has no physical angular resolution, and spatial information is instead simply inferred, it is privacy-conserving. Predicted images are based on the training distribution, and so are the reconstructed features of people in the scene. Further research into this paradigm could focus on different temporal detection modalities like WiFi routers.

“Always try the simplest approach first, so you have a reference point to justify any increase in complexity. In ML, this typically means using a non-ML baseline as your first model."

- François Chollet, creator of Keras

Chapter 8.

Appendix

Convolution of signal with IRF

Why is the measured signal in a FLIM experiment the convolution of the exponential decay signal with the instrument response function? This might seem counter-intuitive, if we consider that the convolution between two functions reverses the second function and scans it over the first, as depicted in Fig. 8.1. Cross-correlation and auto-correlation are shown for comparison.

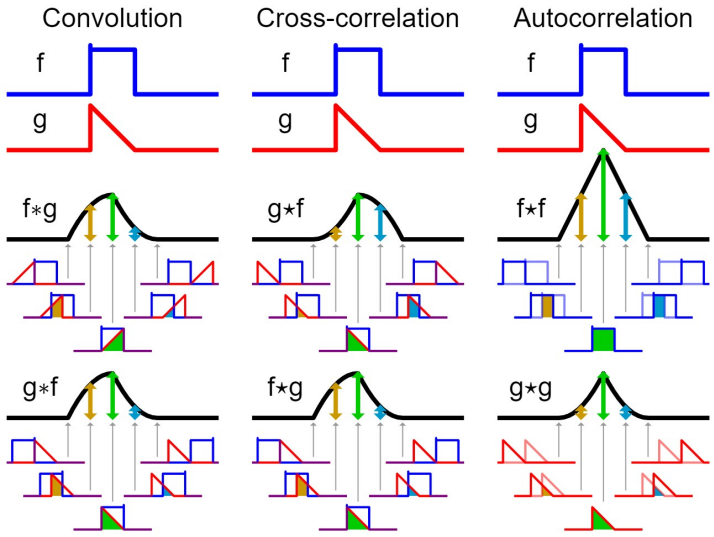

Figure 8.1: Schematic comparing convolution, cross-correlation and autocorrelation, from [168]. For the convolution of f and g, g is reversed and scanned over f, their dot product calculated at each scan point.

Why would this make sense for FLIM? To understand, let us briefly consider what convolution the IRF and convolution are. The IRF is the signal that the imaging system generates from in response to an impulse, i.e. an extremely short burst. For fluorescence, this is equal to the response we get from something that decays instantly. As for convolution, for two time-dependent functions f(t) and g(t), their convolution at time τ is:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

In other words, the convolution is the dot product of the first signal with the time-reversed second signal.

Fig. 8.2(a) illustrates an short pulse p(t) and the measured IRF(t). As Fig. 8.2(b) shows, when the system takes the input $p(t)$, we can write the measurement as $\int_{-\infty}^{\infty} p(\tau) IRF(t-\tau) d\tau = (t) * IRF$. In the limit when p(t) is an infinitely short pulse (an impulse), this by definition equals the IRF itself.

Fig. 8.2(c) uses this logic, decompose a signal $h(t)$ as the sum of many subsequent pulses of different heights. Each impulse returns its own signal:

$$h(t) = p(t) + \alpha p(t - t_0) + \beta p(t - 2t_0) + \gamma p(t - 3t_0) \tag{8.1}$$

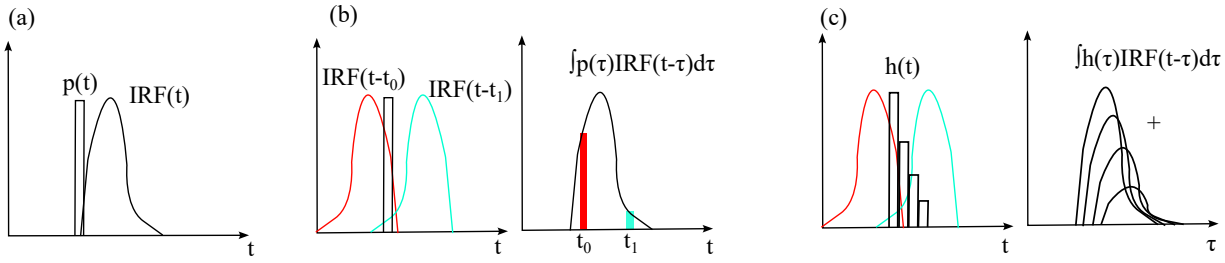

Figure 8.2: (a) A short pulse p and the system IRF. (b) Left: two reversed IRFs at different offsets τ_1 and τ_2, shown in red and cyan. Note where the red signal and the cyan intersect with p(t). Right: the convolution of p(t) with the IRF, which is the overlap between the pulse and reversed IRFs. A red and cyan point mark the height of intersections of the red and cyan curves with the pulse. This convolution is per-definition what the measurement gives: measuring a short pulse gives the IRF. (c) Left: a signal h(t) is approximated as the sum of many short pulses. Right: The signal generated by the system as a response to h(t) is therefore the sum of the signals generated from the many short pulses. This equals the convolution of h(t) with the IRF. Ergo, we measure $(h * IRF)(t)$

We can derive the signal measured by each short pulse using the same logic as in Fig. 8.2(b). Then, the signal s measured from h(t), which is the sum of these pulses, is the sum of the measurement from the individual pulses

$$\begin{aligned}
s(t) = &\int_{-\infty}^{\infty} p(\tau) IRF(t-\tau) d\tau \\
&+ \alpha \int_{-\infty}^{\infty} p(\tau - t_0) IRF(t-\tau) d\tau \\
&+ \beta \int_{-\infty}^{\infty} p(\tau - 2t_0) IRF(\tau - t) d\tau \\
&+ \gamma \int_{-\infty}^{\infty} p(\tau - 3t_0) IRF(\tau - t) d\tau \\
= &\int_{-\infty}^{\infty} \left[p(\tau) + \alpha p(\tau - t_0) + \beta p(\tau - 2t_0) + \gamma p(\tau - 3t_0) \right] IRF(\tau - t) d\tau
\end{aligned}$$

$$= \int_{-\infty}^{\infty} h(\tau) IRF(t-\tau) d\tau$$
$$= (h * IRF)(t)$$

Rapid lifetime determination derivations

Full derivation of RLD for two gates (one from 0 to x_0, the other from x_0 to ∞) in the nomenclature used in the main text:

$$a_1 = A_0 \int_0^{x_1} \exp\left(\frac{-x}{\tau}\right) dx$$
$$a_2 = A_0 \int_{x_1}^{\infty} \exp\left(\frac{-x}{\tau}\right) dx$$
$$\Rightarrow a_1 = A_0 \tau (\exp(\frac{x_1}{\tau}) - 1))$$
$$a_2 = A_0 \tau (-\exp(\frac{x_1}{\tau}))$$
$$\Rightarrow \frac{a_1}{a_2} = \frac{\exp(\frac{x_1}{\tau}) - 1}{-\exp(\frac{x_1}{\tau})}$$
$$= \frac{1}{\exp(\frac{x_1}{\tau})} - 1$$
$$\Rightarrow \frac{1}{\frac{a_1}{a_2} + 1} = \exp(\frac{x_1}{\tau})$$
$$\Rightarrow \log\left(\frac{1}{\frac{a_1}{a_2} + 1}\right) = \frac{x_1}{\tau}$$
$$\therefore \tau = -\frac{x_1}{\log(\frac{a_1}{a_2} + 1)}$$

Consider two TCSPC time-bins, $A(t_1)$ and (A_2):

$$A(t_1) = A_0 \exp\left(\frac{t_0 - t_1}{\tau}\right)$$
$$A(t_2) = A_0 \exp\left(\frac{t_0 - t_2}{\tau}\right)$$
$$\Rightarrow \log(A(t_1)) = \log(A_0) + \frac{t_0 - t_1}{\tau}$$
$$\log(A(t_2)) = \log(A_0) + \frac{t_0 - t_2}{\tau}$$
$$\Rightarrow \log(A(t_2)) - \log(A(t_1)) = \frac{t_0 - t_2}{\tau} - \frac{t_0 - t_1}{\tau}$$
$$= \frac{t_1 - t_2}{\tau}$$

$$\therefore \tau = \frac{t_1 - t_2}{\log(A(t_2)) - \log(A(t_1))}$$

Phasor derivations

Full derivation of Fourier transform of monoexponential decay signal, evaluated at frequency ω:

$$\begin{aligned}
A(t) &= A_0 \exp(\frac{-t}{\tau}) \text{ for } 0 <\leq \infty \\
\implies F(A)(\omega) &= A_0 \int_0^\infty \exp(\frac{-t}{\tau}) \exp(-j\omega t) dt \\
&= A_0 \int_0^\infty \exp(-t(\frac{1}{\tau} + j\omega)) dt \\
&= A_0 \Big[-\frac{1}{\frac{1}{\tau} + j\omega} \exp -t(\frac{1}{\tau} + j\omega) \Big]_0^\infty \\
&= 0 - A_0 \Big(-\frac{1}{\frac{1}{\tau} + j\omega} \Big) \\
&= A_0 \Big(\frac{1}{\frac{1}{\tau} + j\omega} \Big) \\
&\overset{\frac{1}{\tau} \neq j\omega}{=} A_0 \frac{\frac{1}{\tau} - j\omega}{(\frac{1}{\tau} + j\omega)(\frac{1}{\tau} - j\omega)} \\
&= A_0 \frac{\frac{1}{\tau} - j\omega}{\frac{1}{\tau^2} + \omega^2} \\
&= \underbrace{A_0 \frac{\frac{1}{\tau}}{(\frac{1}{\tau^2} + \omega^2)}}_{\text{Re}} \underbrace{- j\, A_0 \frac{\omega}{(\frac{1}{\tau^2} + \omega^2)}}_{\text{Im}}
\end{aligned}$$

Full derivation of using intensity weighted magnitude of Fourier transform for predicting fluorescence lifetime:

$$\begin{aligned}
m &= \sqrt{g^2 + s^2} \\
&= \sqrt{\frac{(\frac{1}{\tau^2})^2 + \frac{\omega^2}{\tau^2}}{(\frac{1}{\tau^2} + \omega^2)^2}} \qquad (8.2) \\
&= \sqrt{\frac{\frac{1+\omega^2\tau^2}{\tau^4}}{\frac{(1+\omega^2\tau^2)^2}{\tau^4}}} \qquad (8.3)
\end{aligned}$$

$$= \sqrt{\frac{\frac{1+\omega^2\tau^2}{\tau^4}}{\frac{(1+\omega^2\tau^2)^2}{\tau^4}}} \tag{8.4}$$

$$= \sqrt{\frac{1}{1+\omega^2\tau^2}} \tag{8.5}$$

CMM derivation

Full derivation of CMM for a mono-exponential fluorescence decay A(t) measured with neglible IRF:

$$CM \equiv \frac{\int_0^\infty tA(t)dt}{\int_0^\infty A(t)dt} \tag{8.6}$$
$$A(t) = A\exp(-t/\tau)$$

We integrate the numerator by parts $\int u \cdot dv = u \cdot v - \int du \cdot v$, where $u = t$ and $dv = \exp(-t/\tau)$, hence $du = 1$ and $v = -\tau\exp(-t/\tau)$:

$$\begin{aligned}
\int_0^\infty tA(t)dt &= A\Big[t(-\tau\exp(-t/\tau)\Big]_0^\infty - A\int_0^\infty 1 \cdot -\tau\exp(-t/\tau)dt \\
&= A\Big[t(-\tau\exp(-t/\tau)\Big]_0^\infty - A\Big[\tau^2\exp(-t/\tau)\Big]_0^\infty \\
&= A\Big[-\tau\exp(-t/\tau)(t+\tau)\Big]_0^\infty \\
&= -A\tau\lim_{t\to\infty}\frac{t+\tau}{\exp(t/\tau)} - A(-\tau^2) \\
&\overset{L'Hospital}{=} -A\tau\lim_{t\to\infty}\frac{1}{1/\tau\exp(t/\tau)} + A\tau^2 \\
&= A\tau^2
\end{aligned}$$

Evaluating the denominator of Eq. 8.6 gives $\int_0^\infty A(t)dt = A\tau$. Thus:

$$CM = \frac{A\tau^2}{A\tau} = \tau$$

Time-gate scanning

Fig. 8.3 shows that scanning a time-gate forward in time inverts the gate upon measurement. For a starting gate shape $g(t)$, the gate at measurement i is $g(t - t_i)$, where t_i is the time offset of the gate from the start. Each measurement m_{t_i} of the pulse $p(t)$ is then:

$$m_{t_i} = \int_{-\infty}^{\infty} p(t)g(t - t_i)dt \tag{8.7}$$

As we can see, this is a cross-correlation, between the gate and the pulse, hence the gate signal we measure is reversed in time compared to the true gate.

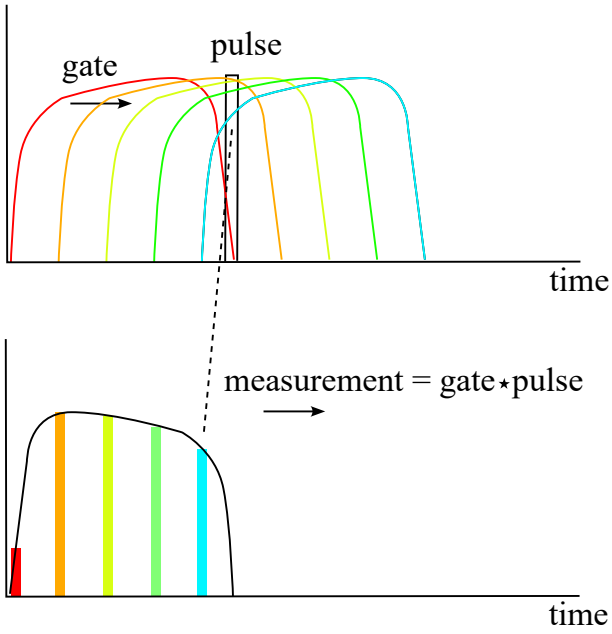

Figure 8.3: Schematic of how scanning a time-gate forward in time creates a measurement that show the gate in reverse. The colours of the gate at various positions coincide with the colours of the measurement made when scanning the pulse with that gate position. The measurement is at each point is a dot product between the pulse and the shifted gate, hence it is overall a cross-correlation

Fluorescent intensity of a mono-exponential

Consider a mono-exponential function of amplitude A_0, starting time t_0, and lifetime τ. At time t, the function value A(t) is:

$$A(t) = A_0 \exp\left(\frac{t_0 - t}{\tau}\right)$$

The fluorescent intensity observed over time is the integral under this decay curve:

$$\begin{aligned} I &= \int_{t_0}^{\infty} A_0 \exp\left(\frac{t_0 - t}{\tau}\right) dt \\ &\overset{x=t-t_0}{=} \int_{0}^{\infty} A_0 \exp\left(\frac{-x}{\tau}\right) dx \\ &= A_0 \Big[-\tau \exp\left(\frac{-x}{\tau}\right) \Big]_0^{\infty} \\ &= A_0 \tau (0 + 1) \\ &= A_0 \tau \end{aligned}$$

Cramér-Rao lower bound of fluorescence lifetime estimation

The Cramér–Rao bound is a lower bound on the estimation uncertainty of some parameter τ from a measurement T. In the example of FLIM, τ is lifetime, and T is the arrival time of a photon at our detector, assuming value t in a given measurement, $T = t$.

The Cramér–Rao bound states that the variance of an unbiased estimator $\hat{\tau}$ is at least as large as the inverse of the Fisher Information $I(\tau)$ our measurements carry about the lifetime:

$$\mathrm{var}(\hat{\tau}) \geq \frac{1}{I(\tau)} \tag{8.8}$$

Fisher Information, in turn, measures how rapidly the distribution of photon arrival times $f(T;\tau)$ changes with τ. For N independent and identically distributed measurements, Fisher Information is defined as:

$$I(\tau) = N\mathbb{E}_{\tau}\left[\left(\frac{\delta \log f(T;\tau)}{\delta \tau}\right)^2 \middle| \tau\right] = N \int_{\mathbb{R}} \left(\frac{\delta}{\delta \tau} \log f(t;\tau)\right)^2 f(t;\tau) dt \tag{8.9}$$

Assuming a perfect detector (delta IRF, quantum efficiency of 1, no noise), the probability density function of photon arrival at time t is:

$$f(t,\tau) = \frac{1}{\tau} \exp[-\frac{t}{\tau}] \tag{8.10}$$

where the factor $\frac{1}{\tau}$ is a normalisation factor, ensuring that the probability of measuring the photon decay at *some* time between $t = 0$ and $t = \infty$ is 1: $\int_0^{\infty} \exp[-\frac{t}{\tau}]dt = [-\tau \exp[-\frac{t}{\tau}]]_0^{\infty} = -\tau(0 - 1) = \tau \implies \int_0^{\infty} \frac{1}{\tau} \exp[-\frac{t}{\tau}]dt = 1$.

Substituting Eq. 8.10 into Eq. 8.9, we get:

$$I(\tau) = N \int_{\mathbb{R}} \left(\frac{\delta}{\delta\tau} \log(\frac{1}{\tau} \exp[-\frac{t}{\tau}]) \right)^2 \frac{1}{\tau} \exp[-\frac{t}{\tau}] dt \tag{8.11}$$

We first find expand the squared term:

$$\begin{aligned} \left(\frac{\delta}{\delta\tau} \log(\frac{1}{\tau} \exp[-\frac{t}{\tau}]) \right)^2 &= \left(\frac{\delta}{\delta\tau} (\log \tau^{-1} - \frac{t}{\tau}) \right)^2 = \\ &= (-\tau^{-1} + t\tau^{-2})^2 \\ &= \tau^{-2} - 2t\tau^{-3} + t^2\tau^{-4} \end{aligned}$$

Substituting into Eq. 8.11, we get:

$$\begin{aligned} I(\tau) &= N \int_{\mathbb{R}} (\tau^{-2} - 2t\tau^{-3} + t^2\tau^{-4}) \frac{1}{\tau} \exp[-\frac{t}{\tau}] dt \\ &= N \Big[\underbrace{\tau^{-3} \int_0^\infty \exp[-\frac{t}{\tau}] dt}_{\text{term 1}} - \underbrace{2\tau^{-4} \int_0^\infty t \exp[-\frac{t}{\tau}] dt}_{\text{term 2}} + \underbrace{t\tau^{-5} \int_0^\infty \exp[-\frac{t}{\tau}] dt}_{\text{term 3}} \Big] \\ \text{term 1} &= \tau^{-3} \int_0^\infty \exp[-\frac{t}{\tau}] dt = -\tau^{-2} \Big[\exp[-\frac{t}{\tau}] \Big]_0^\infty = -\tau^{-2}(0-1) = \tau^{-2} \\ \text{term 2} &= 2\tau^{-4} \int_0^\infty t \exp[-\frac{t}{\tau}] dt = ... = 2\tau^{-4}(\tau^2) = 2\tau^{-2} \\ \text{term 3} &= \tau^{-4} \int_0^\infty t^2 \exp[-\frac{t}{\tau}] dt = ... = \tau^{-5}(2\tau^{-3}) = 2\tau^{-2} \\ \therefore I(\tau) &= N(\tau^{-3} - 2\tau^{-2} + 2\tau^{-2}) = N\tau^{-2} \end{aligned}$$

Substituting into Eq. 8.8, we see that:

$$\begin{aligned} \text{var}(\hat{\tau}) &\geq N^{-1}\tau^2 \\ \sigma(\hat{\tau}) &\geq \frac{\tau}{\sqrt{N}} \end{aligned} \tag{8.12}$$

This assumes an ideal detector. In reality, detectors have IRFs, readout noise, quantisation error and other “imperfections". We simplify the cumulative effect of these imperfections via a scaling factor, the photon economy F, giving the expression used in the main text:

$$\sigma(\hat{\tau}) \geq F\frac{\tau}{\sqrt{N}}$$

Note: one can also derive this expression using the central limit theorem. Let us assume photons arrive according to $f(t,\tau) = \frac{1}{\tau}\exp[-\frac{t}{\tau}]$, as derived in Eq. 8.10. The lifetime is the expected arrival time; this expectation value is approximated as the mean arrival time of a given sample of photons. Let us say we sample N photons (detect N photons on the detector and log their arrival times).

The central limit theorem states that the mean of our arrival time samples, μ_T, for large N, will be distributed as:

$$\mu_T \sim \mathbb{N}\Big(\mathbb{E}[T], \frac{\text{var}(T)}{N}\Big) \tag{8.13}$$

The expected arrival time, $\mathbb{E}[T]$, is:

$$\mathbb{E}[T] = \int_0^\infty t\frac{1}{\tau}\exp[-\frac{t}{\tau}]dt = \tau$$

The variance, var(T), is:

$$\text{var}(T) = \mathbb{E}[T^2] - \mathbb{E}[T] = \int_0^\infty t^2\frac{1}{\tau}\exp[-\frac{t}{\tau}]dt - \tau^2 = 2\tau^2 - \tau^2 = \tau^2$$

Substituting these parameters into Eq. 8.13, and using the mean arrival time as our lifetime guess (see Chapter 2 Sec. 2.1.7 for an introduction to centre-of-mass lifetime estimation, which does exactly this) we get results matching our Fisher Information derivation in Eq. 8.12 for large N:

$$\hat{\tau} \equiv \mu_T \sim \mathbb{N}\Big(\tau, \frac{\tau^2}{N}\Big)$$

Cavity FLIM inverse retrieval

Our forward model uses equations (4.1) and (4.2) to calculate the measurement registered on the sensors. We use the function **P** to concisely represent the iCCD forward model, which maps lifetime map τ and amplitude values A to a measurement registered on the iCCD sensor as:

$$\hat{s} = \mathbf{P}(\tau, A) + n \tag{8.14}$$

where n is measurement noise. Integrating (4.2) for t from t_0 to ∞ yields $q_{i,j} = kA_{i,j}\tau_{i,j}$. This can be rearranged as $A_{i,j} = \frac{q_{i,j}}{k\tau_{i,j}}$. Hence, we use the CMOS measurement to evaluate the amplitude of decay $A_{i,j}$, which can then be to the IR algorithm. For this, the factor k in (4.2) has been calibrated experimentally, via a one-time calibration for the system; k represents the net difference between the expected intensities on the iCCD and CMOS, (before the intensifier). With A rewritten as a function of known intensity and unknown lifetime τ, τ becomes the only unknown variables of function **P**:

$$\begin{gathered}\hat{\tau} = \arg\min_{\tau} C(\tau), \text{ where} \\ C(\tau) = \|\mathbf{P}(\tau) - \hat{s}\|_2 + \alpha\|\tau\|_2^2 \\ \text{subject to } \tau \geq 0\end{gathered} \tag{8.15}$$

We initiate the optimization with a random guess for the lifetime map τ. If the lifetime map at some iteration n is represented by $\tau^{(n)}$, then the solution at the next iteration progresses via gradient descent:

$$\tau^{(n+1)} = \tau^{(n)} - \beta[\nabla_\tau C(\tau)]_{\tau=\tau^{(n)}} \tag{8.16}$$

where β is the step size, determined in each iteration by backtracking line search [169]. $\nabla_\tau C(\tau)$ is the gradient of the cost function with respect to τ:

$$\nabla_\tau C(\tau) = \frac{(\mathbf{P}(\tau) - \hat{s})}{\|\mathbf{P}(\tau) - \hat{s}\|_2}\frac{\partial \mathbf{P}(\tau)}{\partial \tau} + 2\alpha\sum_{i,j}\tau_{i,j} \tag{8.17}$$

We evaluate the above partial derivative by considering the discrete version of (4.1), giving:

$$\begin{aligned}\left.\frac{\partial \mathbf{P}(\tau)}{\partial \tau}\right|_{i,j} = &- f_1\sum_t G(t)q_{i,j}e^{\frac{t_0-t}{\tau_{i,j}}}\left[\frac{t_0-t}{\tau_{i,j}^3} + \frac{1}{\tau_{i,j}^2}\right] \\ &- f_2\sum_t G(t)q_{i,j-y}e^{\frac{t_0+t_c-t}{\tau_{i,j-y}}}\left[\frac{t_0+t_c-t}{\tau_{i,j-y}^3} + \frac{1}{\tau_{i,j-y}^2}\right] \\ &- \ldots \\ &- f_6\sum_t G(t)q_{i,j-5y}e^{\frac{t_0+5t_c-t}{\tau_{i,j-5y}}}\left[\frac{t_0+5t_c-t}{\tau_{i,j-5y}^3} + \frac{1}{\tau_{i,j-5y}^2}\right]\end{aligned} \tag{8.18}$$

Overlapping, iCCD only - bead results

Fig. 8.4 (c-d) and (e-f) show reconstruction results with and without the CMOS, respectively. The smaller beads are expected to show a $2ns$ lifetime while the larger ones are expected to have a lifetime of $4ns$. The CNN which is given the CMOS and the iCCD produces a much more

faithful lifetime reconstruction than the CNN which only sees the iCCD image.

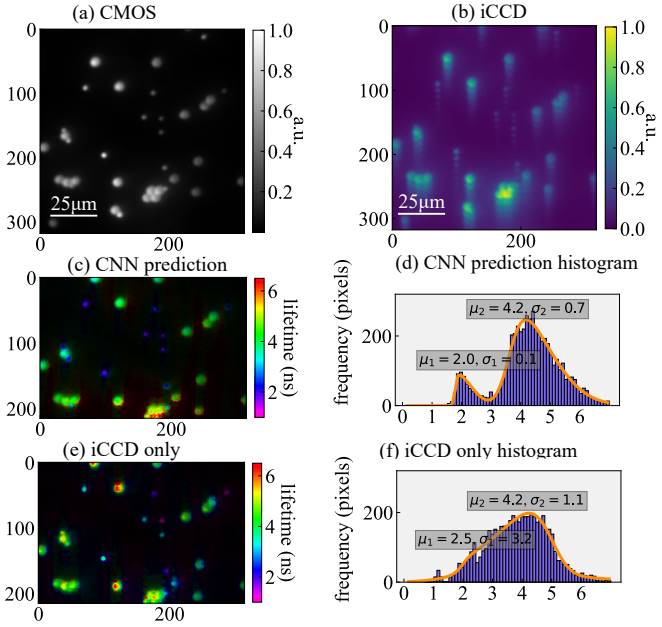

Figure 8.4: Lifetime reconstructions of (a-b) our bead sample from Fig. 3 of the main text using a CNN, with and without the CMOS camera. (c-d) CMOS and the iCCD data fusion reconstruction. (e-f) iCCD-only reconstuction.

Bead lifetime validation with FLIMera

Fig. 8.5 shows our validation samples, acquired with a FLIMera. The decay measurements were fitted using maximum likelihood evaluation based IRF deconvolution. Fig. 8.5 (a) shows the $2\mu m$ samples, while Fig. 8.5 (b) shows the $4\mu m$ samples. For both cases, we show (i) a global lifetime fit, (ii) a histogram of individual pixel fits (iii) the fluorescence intensity of the sample (iv) the sample's fluorescence lifetime, fitted pixelwise.

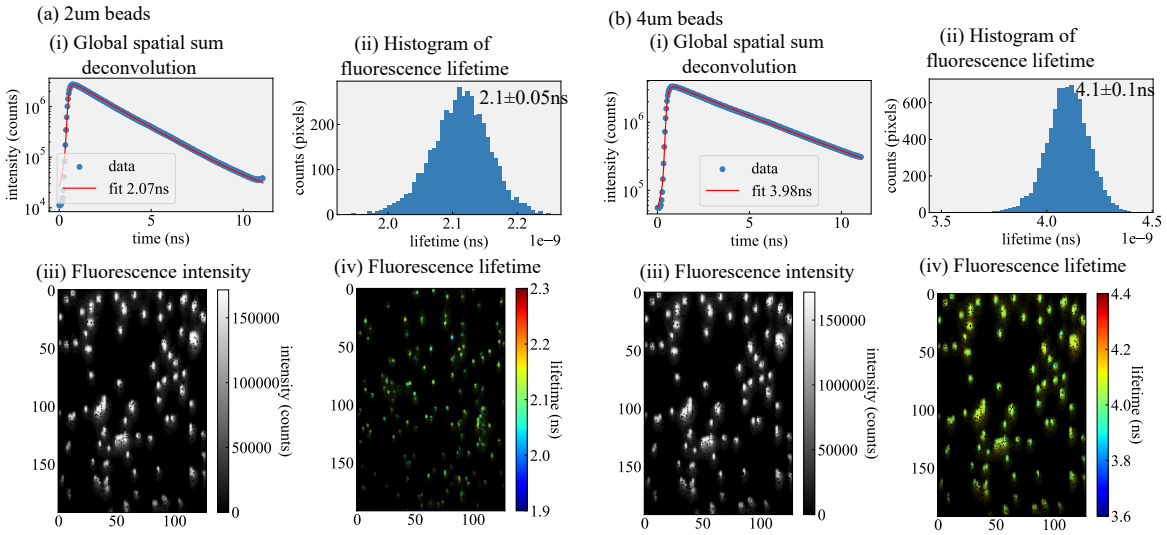

Figure 8.5: **(a)** Data from the $2\mu m$ bead sample. **(b)** Shows the same data for the $4\mu m$ beads.

Cavity FLIM training and validation curves

Fig. 8.6 shows training and validation loss as a function of epoch number for our dilated CNN, in units of $ns^2\alpha$, where alpha is the arbitrary intensity scaling unit. The model generalised very well, aided by our conscious architecture choices and large, diverse training set.

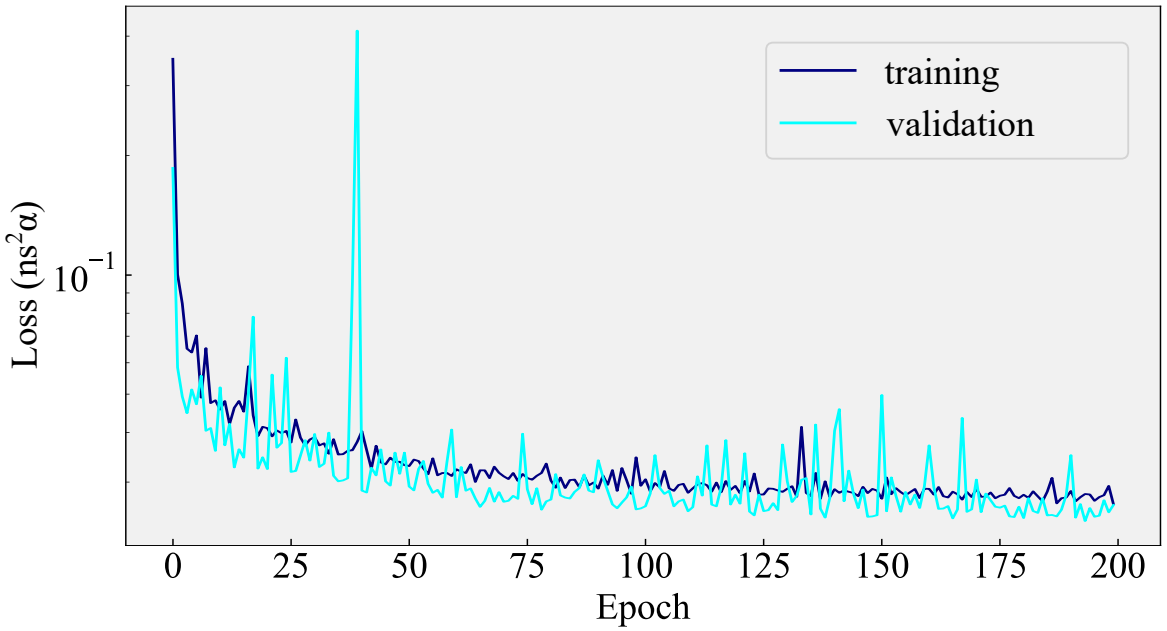

Figure 8.6: Training and validation set loss curves.

Bibliography

[1] V. Zickus, M.-L. Wu, K. Morimoto, V. Kapitany, A. Fatima, A. Turpin, R. Insall, J. Whitelaw, L. Machesky, C. Bruschini, *et al.*, "Fluorescence lifetime imaging with a megapixel spad camera and neural network lifetime estimation," *Scientific Reports*, vol. 10, no. 1, pp. 1–10, 2020.

[2] V. Kapitany, V. Zickus, A. Fatima, G. Carles, and D. Faccio, "Single-shot time-folded fluorescence lifetime imaging," *Proceedings of the National Academy of Sciences*, vol. 120, no. 16, p. e2214617120, 2023.

[3] A. Turpin, G. Musarra, V. Kapitany, F. Tonolini, A. Lyons, I. Starshynov, F. Villa, E. Conca, F. Fioranelli, R. Murray-Smith, *et al.*, "Spatial images from temporal data," *Optica*, vol. 7, no. 8, pp. 900–905, 2020.

[4] A. Turpin, V. Kapitany, J. Radford, D. Rovelli, K. Mitchell, A. Lyons, I. Starshynov, and D. Faccio, "3d imaging from multipath temporal echoes," *Physical Review Letters*, vol. 126, no. 17, p. 174301, 2021.

[5] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[6] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, 2017.

[7] P. Kirkland, V. Kapitany, A. Lyons, J. Soraghan, A. Turpin, D. Faccio, and G. Di Caterina, "Imaging from temporal data via spiking convolutional neural networks," in *Emerging Imaging and Sensing Technologies for Security and Defence V; and Advanced Manufacturing Technologies for Micro-and Nanosystems in Security and Defence III*, vol. 11540, pp. 66–85, SPIE, 2020.

[8] K. Mitchell, K. Kassem, C. Kaul, V. Kapitany, P. Binner, A. Ramsay, R. Murray-Smith, and D. Faccio, "mmsense: Detecting concealed weapons with a miniature radar sensor," *arXiv preprint arXiv:2302.14625*, 2023.

[9] A. Colom, E. Derivery, S. Soleimanpour, C. Tomba, M. D. Molin, N. Sakai, M. González-Gaitán, S. Matile, and A. Roux, "A fluorescent membrane tension probe," *Nature chemistry*, vol. 10, no. 11, pp. 1118–1125, 2018.

[10] C. Grashoff, B. D. Hoffman, M. D. Brenner, R. Zhou, M. Parsons, M. T. Yang, M. A. McLean, S. G. Sligar, C. S. Chen, T. Ha, *et al.*, "Measuring mechanical tension across vinculin reveals regulation of focal adhesion dynamics," *Nature*, vol. 466, no. 7303, pp. 263–266, 2010.

[11] S. H. Lee and R. Dominguez, "Regulation of actin cytoskeleton dynamics in cells," *Molecules and cells*, vol. 29, no. 4, pp. 311–325, 2010.

[12] F. H. van der Linden, E. K. Mahlandt, J. J. Arts, J. Beumer, J. Puschhof, S. de Man, A. O. Chertkova, B. Ponsioen, H. Clevers, J. D. van Buul, *et al.*, "A turquoise fluorescence lifetime-based biosensor for quantitative imaging of intracellular calcium," *Nature Communications*, vol. 12, no. 1, pp. 1–13, 2021.

[13] R. Yasuda, "Imaging spatiotemporal dynamics of neuronal signaling using fluorescence resonance energy transfer and fluorescence lifetime imaging microscopy," *Current opinion in neurobiology*, vol. 16, no. 5, pp. 551–561, 2006.

[14] D. Surzhikova, M. Gerasimova, V. Tretyakova, A. Plotnikov, and E. Slyusareva, "Emission properties of fluorescein in strongly acidic solutions," *Journal of Photochemistry and Photobiology A: Chemistry*, vol. 413, p. 113233, 2021.

[15] H. Zhong, T. Duan, H. Lan, M. Zhou, and F. Gao, "Review of low-cost photoacoustic sensing and imaging based on laser diode and light-emitting diode," *Sensors*, vol. 18, no. 7, p. 2264, 2018.

[16] D. L. Andrews, "A unified theory of radiative and radiationless molecular energy transfer," *Chemical Physics*, vol. 135, no. 2, pp. 195–201, 1989.

[17] B. Valeur and M. N. Berberan-Santos, *Molecular fluorescence: principles and applications*. John Wiley & Sons, 2012.

[18] W. Becker, L. M. Hirvonen, J. Milnes, T. Conneely, O. Jagutzki, H. Netz, S. Smietana, and K. Suhling, "A wide-field tcspc flim system based on an mcp pmt with a delay-line anode," *Review of Scientific Instruments*, vol. 87, no. 9, p. 093710, 2016.

[19] M. Gersbach, R. Trimananda, Y. Maruyama, M. Fishburn, D. Stoppa, J. Richardson, R. Walker, R. Henderson, and E. Charbon, "High frame-rate tcspc-flim using a novel spad-based image sensor," in *Detectors and Imaging Devices: Infrared, Focal Plane, Single Photon*, vol. 7780, pp. 357–369, SPIE, 2010.

[20] E. Charbon, "Single-photon imaging in complementary metal oxide semiconductor processes," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 372, no. 2012, p. 20130100, 2014.

[21] R. Datta, T. M. Heaster, J. T. Sharick, A. A. Gillette, and M. C. Skala, "Fluorescence lifetime imaging microscopy: fundamentals and advances in instrumentation, analysis, and applications," *Journal of biomedical optics*, vol. 25, no. 7, p. 071203, 2020.

[22] L. Wei, W. Yan, and D. Ho, "Recent advances in fluorescence lifetime analytical microsystems: Contact optics and cmos time-resolved electronics," *Sensors*, vol. 17, no. 12, p. 2800, 2017.

[23] L. I. BV, "Frequency-domain flim: From nanoseconds to milliseconds." https://www.lambertinstruments.com/applications-1/2014/12/5/frequency-domain-flim-from-nanoseconds-to-milliseconds.

[24] Y. Sun, J. Phipps, D. S. Elson, H. Stoy, S. Tinling, J. Meier, B. Poirier, F. S. Chuang, D. G. Farwell, and L. Marcu, "Fluorescence lifetime imaging microscopy: in vivo application to diagnosis of oral carcinoma," *Optics letters*, vol. 34, no. 13, pp. 2081–2083, 2009.

[25] J. R. Lakowicz, H. Szmacinski, K. Nowaczyk, and M. L. Johnson, "Fluorescence lifetime imaging of free and protein-bound nadh.," *Proceedings of the National Academy of Sciences*, vol. 89, no. 4, pp. 1271–1275, 1992.

[26] K. Morimoto, A. Ardelean, M.-L. Wu, A. C. Ulku, I. M. Antolovic, C. Bruschini, and E. Charbon, "Megapixel time-gated spad image sensor for 2d and 3d imaging applications," *Optica*, vol. 7, no. 4, pp. 346–354, 2020.

[27] C. Callenberg, A. Lyons, D. d. Brok, A. Fatima, A. Turpin, V. Zickus, L. Machesky, J. Whitelaw, D. Faccio, and M. Hullin, "Super-resolution time-resolved imaging using computational sensor fusion," *Scientific reports*, vol. 11, no. 1, pp. 1–8, 2021.

[28] M. Raspe, K. M. Kedziora, B. Van Den Broek, Q. Zhao, S. De Jong, J. Herz, M. Mastop, J. Goedhart, T. W. Gadella, I. T. Young, *et al.*, "siflim: single-image frequency-domain flim provides fast and photon-efficient lifetime data," *nature methods*, vol. 13, no. 6, pp. 501–504, 2016.

[29] O. E. Olarte, J. Andilla, E. J. Gualda, and P. Loza-Alvarez, "Light-sheet microscopy: a tutorial," *Advances in Optics and Photonics*, vol. 10, no. 1, pp. 111–179, 2018.

[30] D. Axelrod, N. L. Thompson, and T. P. Burghardt, "Total internal reflection fluorescent microscopy," *Journal of microscopy*, vol. 129, no. 1, pp. 19–28, 1983.

[31] J. R. Lakowicz, "Plasmonics in biology and plasmon-controlled fluorescence," *Plasmonics*, vol. 1, no. 1, pp. 5–33, 2006.

[32] BeckerHickl, "Sted flim." https://www.becker-hickl.com/applications/sted-flim/.

[33] S. R. P. Pavani, M. A. Thompson, J. S. Biteen, S. J. Lord, N. Liu, R. J. Twieg, R. Piestun, and W. E. Moerner, "Three-dimensional, single-molecule fluorescence imaging beyond the diffraction limit by using a double-helix point spread function," *Proceedings of the National Academy of Sciences*, vol. 106, no. 9, pp. 2995–2999, 2009.

[34] M. Pascucci, S. Ganesan, A. Tripathi, O. Katz, V. Emiliani, and M. Guillon, "Compressive three-dimensional super-resolution microscopy with speckle-saturated fluorescence excitation," *Nature communications*, vol. 10, no. 1, pp. 1–8, 2019.

[35] Y. Li, L. Liu, D. Xiao, H. Li, N. Sapermsap, J. Tian, Y. Chen, and D. D.-U. Li, "Lifetime determination algorithms for time-domain fluorescence lifetime imaging: A review," *Fluorescence Imaging-Recent Advances and Applications*, 2022.

[36] D. Mandic, "Advanced signal processing the method of least squares," 2015.

[37] A. S. Dabir, C. Trivedi, Y. Ryu, P. Pande, and J. A. Jo, "Fully automated deconvolution method for on-line analysis of time-resolved fluorescence spectroscopy data based on an iterative laguerre expansion technique," *Journal of Biomedical Optics*, vol. 14, no. 2, p. 024030, 2009.

[38] K. K. Sharman, A. Periasamy, H. Ashworth, and J. Demas, "Error analysis of the rapid lifetime determination method for double-exponential decays and new windowing schemes," *Analytical chemistry*, vol. 71, no. 5, pp. 947–952, 1999.

[39] A. Clayton, "Frequency-domain fluorescence lifetime imaging microscopy (fd-flim)," in *Practical Manual for Fluorescence Microscopy Techniques*, pp. 3–4, PicoQuant GmbH Berlin, 2016.

[40] R. Roriz, J. Cabral, and T. Gomes, "Automotive lidar technology: A survey," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[41] P. Blondel and B. J. Murton, *Handbook of seafloor sonar imagery*, vol. 7. Wiley Chichester, 1997.

[42] R. N. Rankin, A. Fenster, D. Downey, P. Munk, M. Levin, and A. Vellet, "Three-dimensional sonographic reconstruction: techniques and diagnostic applications.," *AJR. American journal of roentgenology*, vol. 161, no. 4, pp. 695–702, 1993.

[43] A. Zyweck and R. E. Bogner, "Radar target classification of commercial aircraft," *IEEE Transactions on Aerospace and Electronic systems*, vol. 32, no. 2, pp. 598–606, 1996.

[44] M. L. Stone and G. P. Banner, "Radars for the detection and tracking of ballistic missiles, satellites, and planets," *Lincoln laboratory journal*, vol. 12, no. 2, pp. 217–244, 2000.

[45] N. Lazaros, G. C. Sirakoulis, and A. Gasteratos, "Review of stereo vision algorithms: from software to hardware," *International Journal of Optomechatronics*, vol. 2, no. 4, pp. 435–462, 2008.

[46] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," *arXiv preprint arXiv:1812.11941*, 2018.

[47] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12179–12188, 2021.

[48] B. Behroozpour, P. A. Sandborn, M. C. Wu, and B. E. Boser, "Lidar system architectures and circuits," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 135–142, 2017.

[49] S. Royo and M. Ballesta-Garcia, "An overview of lidar imaging systems for autonomous vehicles," *Applied sciences*, vol. 9, no. 19, p. 4093, 2019.

[50] H. Sarbolandi, M. Plack, and A. Kolb, "Pulse based time-of-flight range sensing," *Sensors*, vol. 18, no. 6, p. 1679, 2018.

[51] J. P. Godbaz, M. J. Cree, A. A. Dorrington, and A. D. Payne, "A fast maximum likelihood method for improving amcw lidar precision using waveform shape," in *SENSORS, 2009 IEEE*, pp. 735–738, IEEE, 2009.

[52] S. Kawahito, I. A. Halin, T. Ushinaga, T. Sawada, M. Homma, and Y. Maeda, "A cmos time-of-flight range image sensor with gates-on-field-oxide structure," *IEEE Sensors Journal*, vol. 7, no. 12, pp. 1578–1586, 2007.

[53] S.-H. Lee, W.-H. Kwon, Y.-S. Lim, and Y.-H. Park, "Highly precise amcw time-of-flight scanning sensor based on parallel-phase demodulation," *Measurement*, vol. 203, p. 111860, 2022.

[54] C. Niclass, A. Rochas, P.-A. Besse, and E. Charbon, "Design and characterization of a cmos 3-d image sensor based on single photon avalanche diodes," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 9, pp. 1847–1854, 2005.

[55] B. Behroozpour, P. A. Sandborn, N. Quack, T.-J. Seok, Y. Matsui, M. C. Wu, and B. E. Boser, "Electronic-photonic integrated circuit for 3d microimaging," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 161–172, 2016.

[56] M. Jacob, J. C. Ye, L. Ying, and M. Doneva, "Computational mri: Compressive sensing and beyond [from the guest editors]," *IEEE signal processing magazine*, vol. 37, no. 1, pp. 21–23, 2020.

[57] P. Mikhail, M. G. D. Le, and G. Mair, "Computational image analysis of nonenhanced computed tomography for acute ischaemic stroke: a systematic review," *Journal of Stroke and Cerebrovascular Diseases*, vol. 29, no. 5, p. 104715, 2020.

[58] A. Kumar, "Computer-vision-based fabric defect detection: A survey," *IEEE transactions on industrial electronics*, vol. 55, no. 1, pp. 348–363, 2008.

[59] A. Kumar and G. K. Pang, "Defect detection in textured materials using gabor filters," *IEEE Transactions on industry applications*, vol. 38, no. 2, pp. 425–440, 2002.

[60] J. Xu, H. Ma, and Y. Liu, "Stochastic optical reconstruction microscopy (storm)," *Current protocols in cytometry*, vol. 81, no. 1, pp. 12–46, 2017.

[61] R. Henriques, C. Griffiths, E. Hesper Rego, and M. M. Mhlanga, "Palm and storm: unlocking live-cell super-resolution," *Biopolymers*, vol. 95, no. 5, pp. 322–331, 2011.

[62] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: a generalist algorithm for cellular segmentation," *Nature methods*, vol. 18, no. 1, pp. 100–106, 2021.

[63] M. Khosravy, N. Gupta, N. Patel, and C. A. Duque, "Recovery in compressive sensing: a review," *Compressive sensing in healthcare*, pp. 25–42, 2020.

[64] M. Rani, S. B. Dhok, and R. B. Deshmukh, "A systematic review of compressive sensing: Concepts, implementations and applications," *IEEE Access*, vol. 6, pp. 4875–4894, 2018.

[65] S. B. Kotsiantis, I. Zaharakis, P. Pintelas, *et al.*, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, no. 1, pp. 3–24, 2007.

[66] M. E. Celebi and K. Aydin, *Unsupervised learning algorithms*, vol. 9. Springer, 2016.

[67] A. Burkov, *The hundred-page machine learning book*, vol. 1. Andriy Burkov Quebec City, QC, Canada, 2019.

[68] D. MacKay, "Information theory, pattern recognition and neural networks," in *Proceedings of the 1st International Conference on Evolutionary Computation*, Cambridge University Press Cambridge, UK, 2003.

[69] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[70] J. T. Smith, R. Yao, N. Sinsuebphon, A. Rudkouskaya, J. Mazurkiewicz, M. Barroso, P. Yan, and X. Intes, "Ultra-fast fit-free analysis of complex fluorescence lifetime imaging via deep learning," *bioRxiv*, p. 523928, 2019.

[71] A. K. Pediredla, A. C. Sankaranarayanan, M. Buttafava, A. Tosi, and A. Veeraraghavan, "Signal processing based pile-up compensation for gated single-photon avalanche diodes," *arXiv preprint arXiv:1806.07437*, 2018.

[72] A. Esposito, "How many photons are needed for fret imaging?," *Biomedical optics express*, vol. 11, no. 2, pp. 1186–1202, 2020.

[73] A. Ulku, A. Ardelean, M. Antolovic, S. Weiss, E. Charbon, C. Bruschini, and X. Michalet, "Wide-field time-gated spad imager for phasor-based flim applications," *Methods and applications in fluorescence*, vol. 8, no. 2, p. 024002, 2020.

[74] N. Miyoshi, K. Hara, I. Yokoyama, G. Tomita, and M. Fukuda, "Fluorescence lifetime of acridine orange in sodium dodecyl sulfate premicellar solutions," *Photochemistry and photobiology*, vol. 47, no. 5, pp. 685–688, 1988.

[75] A. J. Lam, F. St-Pierre, Y. Gong, J. D. Marshall, P. J. Cranfill, M. A. Baird, M. R. McKeown, J. Wiedenmann, M. W. Davidson, M. J. Schnitzer, *et al.*, "Improving fret dynamic range with bright green and red fluorescent proteins," *Nature methods*, vol. 9, no. 10, pp. 1005–1012, 2012.

[76] K. J. Martin, E. J. McGhee, J. P. Schwarz, M. Drysdale, S. M. Brachmann, V. Stucke, O. J. Sansom, and K. I. Anderson, "Accepting from the best donor; analysis of long-lifetime donor fluorescent protein pairings to optimise dynamic flim-based fret experiments," *PloS one*, vol. 13, no. 1, p. e0183585, 2018.

[77] ImageJ, "Bigstitcher."

[78] M. Vitali, F. Picazo, Y. Prokazov, A. Duci, E. Turbin, C. Götze, J. Llopis, R. Hartig, A. J. Visser, and W. Zuschratter, "Wide-field multi-parameter flim: long-term minimal invasive observation of proteins in living cells," *PLoS One*, vol. 6, no. 2, p. e15820, 2011.

[79] Z. Wang, Y. Zheng, D. Zhao, Z. Zhao, L. Liu, A. Pliss, F. Zhu, J. Liu, J. Qu, and P. Luan, "Applications of fluorescence lifetime imaging in clinical medicine," *J. Innov. Opt. Health Sci.*, vol. 11, p. 1830001, July 2017.

[80] J. Jonkman and C. M. Brown, "Any way you slice it—a comparison of confocal microscopy techniques," *Journal of biomolecular techniques: JBT*, vol. 26, no. 2, p. 54, 2015.

[81] K. Suhling, L. M. Hirvonen, W. Becker, S. Smietana, H. Netz, J. Milnes, T. Conneely, A. Le Marois, O. Jagutzki, F. Festy, *et al.*, "Wide-field time-correlated single photon counting-based fluorescence lifetime imaging microscopy," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 942, p. 162365, 2019.

[82] L. M. Hirvonen and K. Suhling, "Wide-field TCSPC: Methods and applications," *Meas. Sci. Technol.*, vol. 28, p. 012003, Dec. 2016.

[83] K. Suhling, L. M. Hirvonen, W. Becker, S. Smietana, H. Netz, J. Milnes, T. Conneely, A. L. Marois, O. Jagutzki, F. Festy, Z. Petrášek, and A. Beeby, "Wide-field time-correlated single photon counting-based fluorescence lifetime imaging microscopy," *Nucl. Instrum. Methods Phys. Res.*, vol. 942, p. 162365, Oct. 2019.

[84] R. Krishnan, H. Saitoh, H. Terada, V. Centonze, and B. Herman, "Development of a multiphoton fluorescence lifetime imaging microscopy system using a streak camera," *Review of Scientific Instruments*, vol. 74, no. 5, pp. 2714–2721, 2003.

[85] A. Campillo and S. Shapiro, "Picosecond streak camera fluorometry-a review," *IEEE Journal of Quantum Electronics*, vol. 19, no. 4, pp. 585–603, 1983.

[86] G. H. Kassier, K. Haupt, N. Erasmus, E. Rohwer, H. Von Bergmann, H. Schwoerer, S. M. Coelho, and F. D. Auret, "A compact streak camera for 150 fs time resolved measurement of bright pulses in ultrafast electron diffraction," *Review of Scientific Instruments*, vol. 81, no. 10, p. 105103, 2010.

[87] J. Itatani, F. Quéré, G. L. Yudin, M. Y. Ivanov, F. Krausz, and P. B. Corkum, "Attosecond streak camera," *Physical review letters*, vol. 88, no. 17, p. 173903, 2002.

[88] M. Fujiwara and W. Cieslik, "Fluorescence lifetime imaging microscopy: Two-dimensional distribution measurement of fluorescence lifetime," in *Methods in enzymology*, vol. 414, pp. 633–642, Elsevier, 2006.

[89] wikipedia commons, "Cathode-ray tube (crt)."

[90] L. Gao, J. Liang, C. Li, and L. V. Wang, "Single-shot compressed ultrafast photography at one hundred billion frames per second," *Nature*, vol. 516, no. 7529, pp. 74–77, 2014.

[91] Y. Ma, Y. Lee, C. Best-Popescu, and L. Gao, "High-speed compressed-sensing fluorescence lifetime imaging microscopy of live cells," *Proceedings of the National Academy of Sciences*, vol. 118, no. 3, p. e2004176118, 2021.

[92] A. J. Bowman, B. B. Klopfer, T. Juffmann, and M. A. Kasevich, "Electro-optic imaging enables efficient wide-field fluorescence lifetime microscopy," *Nat. Commun.*, vol. 10, pp. 1–8, Oct. 2019.

[93] D. M. Jameson and J. A. Ross, "Fluorescence polarization/anisotropy in diagnostics and imaging," *Chemical reviews*, vol. 110, no. 5, pp. 2685–2708, 2010.

[94] Teledyne, "Iccd and emiccd cameras: The basics," 2021.

[95] Andor, "Comparing emccd, iccd and ccd cameras." https://andor.oxinst.com/learning/view/article/ccd,-emccd-and-iccd-comparisons, N/A.

[96] Andor, "istar 334t ccd." https://andor.oxinst.com/products/intensified-camera-series/istar-334t, 2018.

[97] R. M. Ballew and J. Demas, "An error analysis of the rapid lifetime determination method for the evaluation of single exponential decays," *Analytical Chemistry*, vol. 61, no. 1, pp. 30–33, 1989.

[98] J. P. Houston, M. A. Naivar, P. Jenkins, and J. P. Freyer, "Capture of fluorescence decay times by flow cytometry," *Current protocols in cytometry*, vol. 59, no. 1, pp. 1–25, 2012.

[99] M. Patting, P. Reisch, M. Sackrow, R. Dowler, M. Koenig, and M. Wahl, "Fluorescence decay data analysis correcting for detector pulse pile-up at very high count rates," *Optical engineering*, vol. 57, no. 3, p. 031305, 2018.

[100] S. Orthaus-Mueller, B. Kraemer, R. Dowler, A. Devaux, A. Tannert, T. Roehlicke, M. Wahl, H.-J. Rahn, and R. Erdmann, "rapidflim: the new and innovative method for ultra fast flim imaging," *PicoQuant application note*, pp. 1–8, 2016.

[101] Matplotlib, "Interpolations for imshow."

[102] Q. Sun, J. Zhang, X. Dun, B. Ghanem, Y. Peng, and W. Heidrich, "End-to-end learned, optically coded super-resolution spad camera," *ACM Trans. Graph.*, vol. 39, mar 2020.

[103] J. Sun, Z. Xu, and H.-Y. Shum, "Image super-resolution using gradient profile prior," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[104] F. Shi, J. Cheng, L. Wang, P.-T. Yap, and D. Shen, "Low-rank total variation for image super-resolution," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 155–162, Springer, 2013.

[105] L. Fan, R. Meng, Q. Guo, M. Shi, and C. Zhang, "Image denoising by low-rank approximation with estimation of noise energy distribution in svd domain," *IET Image Processing*, vol. 13, no. 4, pp. 680–691, 2019.

[106] B. Deka and M. K. R. Baruah, "Single-image super-resolution using compressive sensing," *algorithms*, vol. 6, p. 7, 2013.

[107] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[108] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[109] Q. Sun, X. Dun, Y. Peng, and W. Heidrich, "Depth and transient imaging with compressive spad array cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 273–282, 2018.

[110] F. Soldevila, A. Lenz, A. Ghezzi, A. Farina, C. D'Andrea, and E. Tajahuerce, "Gigavoxel multidimensional fluorescence imaging combining single-pixel detection and data fusion," *Optics Letters*, vol. 46, no. 17, pp. 4312–4315, 2021.

[111] N. Antipa, G. Kuo, R. Heckel, B. Mildenhall, E. Bostan, R. Ng, and L. Waller, "Diffusercam: lensless single-exposure 3d imaging," *Optica*, vol. 5, no. 1, pp. 1–9, 2018.

[112] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[113] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, pp. I–I, IEEE, 2004.

[114] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[115] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[116] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proceedings of the IEEE international conference on computer vision*, pp. 1920–1927, 2013.

[117] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *2009 IEEE 12th international conference on computer vision*, pp. 349–356, IEEE, 2009.

[118] Z. Wang, J. Chen, and S. C. Hoi, "Deep learning for image super-resolution: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3365–3387, 2020.

[119] H. Chen, X. He, L. Qing, Y. Wu, C. Ren, R. E. Sheriff, and C. Zhu, "Real-world single image super-resolution: A brief review," *Information Fusion*, vol. 79, pp. 124–145, 2022.

[120] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[121] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks: Corr," 2015.

[122] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 286–301, 2018.

[123] X. Hu, M. A. Naiel, A. Wong, M. Lamm, and P. Fieguth, "Runet: A robust unet architecture for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[124] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[125] A. Shocher, N. Cohen, and M. Irani, ""zero-shot" super-resolution using deep internal learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3118–3126, 2018.

[126] C. Ren, X. He, and T. Q. Nguyen, "Single image super-resolution via adaptive high-dimensional non-local total variation and adaptive geometric feature," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 90–106, 2016.

[127] W.-Z. Shao, Q. Ge, L.-Q. Wang, Y.-Z. Lin, H.-S. Deng, and H.-B. Li, "Nonparametric blind super-resolution using adaptive heavy-tailed priors," *Journal of Mathematical Imaging and Vision*, vol. 61, pp. 885–917, 2019.

[128] S. Bell-Kligler, A. Shocher, and M. Irani, "Blind super-resolution kernel estimation using an internal-gan," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[129] A. Bulat, J. Yang, and G. Tzimiropoulos, "To learn image super-resolution, use a gan to learn how to do image degradation first," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 185–200, 2018.

[130] S. Bond-Taylor, A. Leach, Y. Long, and C. G. Willcocks, "Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[131] R. Dahl, M. Norouzi, and J. Shlens, "Pixel recursive super resolution," in *Proceedings of the IEEE international conference on computer vision*, pp. 5439–5448, 2017.

[132] Z.-S. Liu, W.-C. Siu, and Y.-L. Chan, "Photo-realistic image super-resolution via variational autoencoders," *IEEE Transactions on Circuits and Systems for video Technology*, vol. 31, no. 4, pp. 1351–1365, 2020.

[133] A. Lugmayr, M. Danelljan, L. Van Gool, and R. Timofte, "Srflow: Learning the super-resolution space with normalizing flow," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pp. 715–732, Springer, 2020.

[134] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *International conference on machine learning*, pp. 1747–1756, PMLR, 2016.

[135] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[136] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *International conference on machine learning*, pp. 1530–1538, PMLR, 2015.

[137] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[138] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult?," in *2002 IEEE International conference on acoustics, speech, and signal processing*, vol. 4, pp. IV–3313, IEEE, 2002.

[139] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[140] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2, pp. 1398–1402, Ieee, 2003.

[141] C. Li and A. C. Bovik, "Content-weighted video quality assessment using a three-component image model," *Journal of Electronic Imaging*, vol. 19, no. 1, pp. 011003–011003, 2010.

[142] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

[143] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 5, pp. 2567–2581, 2020.

[144] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Comparison of full-reference image quality models for optimization of image processing systems," *International Journal of Computer Vision*, vol. 129, pp. 1258–1281, 2021.

[145] H. Plank, G. Holweg, T. Herndl, and N. Druml, "High performance time-of-flight and color sensor fusion with image-guided depth super resolution," in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1213–1218, IEEE, 2016.

[146] M. Campos-Taberner, A. Romero-Soriano, C. Gatta, G. Camps-Valls, A. Lagrange, B. Le Saux, A. Beaupere, A. Boulch, A. Chan-Hon-Tong, S. Herbin, *et al.*, "Processing of extremely high-resolution lidar and rgb data: outcome of the 2015 ieee grss data fusion contest–part a: 2-d contest," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 12, pp. 5547–5559, 2016.

[147] A.-V. Vo, L. Truong-Hong, D. F. Laefer, D. Tiede, S. d'Oleire Oltmanns, A. Baraldi, M. Shimoni, G. Moser, and D. Tuia, "Processing of extremely high resolution lidar and rgb data: Outcome of the 2015 ieee grss data fusion contest—part b: 3-d contest," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 12, pp. 5560–5575, 2016.

[148] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.

[149] A. C. Quiros, N. Coudray, A. Yeaton, X. Yang, L. Chiriboga, A. Karimkhan, N. Narula, H. Pass, A. L. Moreira, J. L. Quesne, *et al.*, "Self-supervised learning unveils morphological clusters behind lung cancer types and prognosis," *arXiv preprint arXiv:2205.01931*, 2022.

[150] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical imaging and vision*, vol. 20, no. 1, pp. 89–97, 2004.

[151] D. R. J. Verboogen, N. G. Mancha, M. Ter Beest, and G. van den Bogaart, "Fluorescence lifetime imaging microscopy reveals rerouting of snare trafficking driving dendritic cell activation," *Elife*, vol. 6, p. e23525, 2017.

[152] M. Stöckl and A. Herrmann, "Detection of lipid domains in model and cell membranes by fluorescence lifetime imaging microscopy," *Biochimica et Biophysica Acta (BBA)-Biomembranes*, vol. 1798, no. 7, pp. 1444–1456, 2010.

[153] A. Pierzyńska-Mach, P. A. Janowski, and J. W. Dobrucki, "Evaluation of acridine orange, lysotracker red, and quinacrine as fluorescent probes for long-term tracking of acidic vesicles," *Cytometry Part A*, vol. 85, no. 8, pp. 729–737, 2014.

[154] A. K. Estandarte, S. Botchway, C. Lynch, M. Yusuf, and I. Robinson, "The use of dapi fluorescence lifetime imaging for investigating chromatin condensation in human chromosomes," *Scientific reports*, vol. 6, no. 1, pp. 1–12, 2016.

[155] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[156] R. K. Henderson, N. Johnston, F. Mattioli Della Rocca, H. Chen, D. Day-Uei Li, G. Hungerford, R. Hirsch, D. Mcloskey, P. Yip, and D. J. S. Birch, "A 192 × 128 time correlated spad image sensor in 40-nm cmos technology," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 7, pp. 1907–1916, 2019.

[157] S. T. Barnard and M. A. Fischler, "Computational stereo," *ACM Computing Surveys (CSUR)*, vol. 14, no. 4, pp. 553–572, 1982.

[158] Y. Frauel, T. J. Naughton, O. Matoba, E. Tajahuerce, and B. Javidi, "Three-dimensional imaging and processing using computational holographic imaging," *Proceedings of the IEEE*, vol. 94, no. 3, pp. 636–653, 2006.

[159] P. Caramazza, A. Boccolini, D. Buschek, M. Hullin, C. F. Higham, R. Henderson, R. Murray-Smith, and D. Faccio, "Neural network identification of people hidden from view with a single-pixel, single-photon detector," *Scientific reports*, vol. 8, no. 1, p. 11945, 2018.

[160] J. H. Nam and A. Velten, "Super-resolution remote imaging using time encoded remote apertures," *Applied Sciences*, vol. 10, no. 18, p. 6458, 2020.

[161] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[162] "Dall-e 2," 2023.

[163] E. Mostaque, "Stable Diffusion Public Release — Stability AI," 3 2023.

[164] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022.

[165] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3d: High-resolution text-to-3d content creation," *arXiv preprint arXiv:2211.10440*, 2022.

[166] Synthesia, "Synthesia | 1 AI Video Generation Platform."

[167] Y. Ming, X. Meng, C. Fan, and H. Yu, "Deep learning for monocular depth estimation: A review," *Neurocomputing*, vol. 438, pp. 14–33, 2021.

[168] Cmglee, "Comparison_convolution_correlation," 2016.

[169] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.