RECONSTRUCTION OF HIGH-SPEED EVENT-BASED VIDEO USING PLUG AND PLAY

A Thesis

Submitted to the Faculty

of

Purdue University

by

Trevor D. Moore

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Electrical and Computer Engineering

December 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. Charles Bouman, Chair

School of Electrical and Computer Engineering

Dr. Stanley Chan

School of Electrical and Computer Engineering

Dr. Mary Comer

School of Electrical and Computer Engineering

Approved by:

Dr. Pedro Irazoqui

Head of the School Graduate Program

ACKNOWLEDGMENTS

Special thanks to Professor Bouman and the Purdue Military Research Initiative for providing me this opportunity to study at Purdue University and for their continued support.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Moore, Trevor D. MSECE, Purdue University, December 2018. Reconstruction of High-Speed Event-Based Video using Plug and Play. Major Professor: Charles A. Bouman.

Event-Based cameras, also known as neuromophic cameras or dynamic vision sensors, are an imaging modality that attempt to mimic human eyes by asynchronously measuring contrast over time. If the contrast changes sufficiently then a 1-bit event is output, indicating whether the contrast has gone up or down. This stream of events is sparse, and its asynchronous nature allows the pixels to have a high dynamic range and high temporal resolution. However, these events do not encode the intensity of the scene, resulting in an inverse problem to estimate intensity images from the event stream. Hybrid event-based cameras, such as the DAVIS camera, provide a reference intensity image that can be leveraged when estimating the intensity at each pixel during an event. Normally, inverse problems are solved by formulating a forward and prior model and minimizing the associated cost, however, for this problem, the Plug and Play (P&P) algorithm is used to solve the inverse problem. In this case, P&P replaces the prior model subproblem with a denoiser, making the algorithm modular, easier to implement. We propose an idealized forward model that assumes the contrast steps measured by the DAVIS camera are uniform in size to simplify the problem. We show that the algorithm can swiftly reconstruct the scene intensity at a user-specified frame rate, depending on the chosen denoiser's computational complexity and the selected frame rate.

1. INTRODUCTION

High-speed cameras are a marvel of engineering able to record dynamic scenes at thousands of frames per second and high-definition resolution. However, they are not without drawbacks; high-speed cameras generate large amounts of data, require well-illuminated scenes, and they are expensive. Neuromorphic, or event-based, sensors are a new imaging modality that offer many advantages over traditional high-speed cameras: they generate less data, have a larger dynamic range, and they cost less. However, this system does have a drawback. Event-based cameras do not record scene intensity at high speed; instead, each pixel simply outputs whether the intensity it measures has gone up or down over time, which is referred to as an event. Thus, this imaging modality only measures the change in the scene, which gives it its advantages, but the drawback is that an intensity image cannot be directly estimated from the events, thereby precluding standard methods to display and analyze the events.

To motivate this work further, let us make a more direct comparison. First, it must be stated that it is difficult to directly compare standard high-speed cameras with an event-based camera, due to their different data outputs. Event-based cameras do not have a frame rate and the number of events is dependent upon the activity in the scene. However, we must find some method to compare them in order to evaluate why we should consider using an event-based camera at all. The DAVIS camera has a single pixel bandwidth of 3kHz, while the interpixel bandwidth is around 1MHz. This means that if the scene changes at a rate greater than 3kHz then individual pixels will not be able to measure the change, but the pixel array can measure the change. An example of this is if an LED was in the field of view of the camera and filled just one pixel, if it was blinking at 6kHz the single pixel would miss some events, but if the LED was blinking at 6kHz and moving across the field of view fast enough then the array of pixels would measure each blink. In an attempt to make the comparison as

fair as possible, we will then consider a high-speed camera filming at the same rate as the maximum rate of a single pixel in the DAVIS camera. A high-speed camera with a resolution of 180x240 pixels, which is the same as the DAVIS cameras resolution, filming at 3kHz and 8-bit grayscale would create 123.6MB/s, uncompressed. At 10 seconds of filming, that becomes 1.21GB of video that must then be compressed and saved. The DAVIS camera outputs both synchronously sampled intensity frames and asynchronous events. If we assume the DAVIS is measuring intensity frames with the Active Pixel Sensor(APS) at 20Hz then that creates 8.24MB of uncompressed data for the same 10 second period. But we must also include the events, which is a bit more challenging. The specifications of the camera state the maximum number of events output by the DAVIS camera is 12 million events per second, however, it is rarely the case that that many events occur. From the data sets used in this work, the average events per second was about 1.5 million. If we consider the spatiotemporal volume created by the 180x240x3000 pixels, this represents about 1% of the volume. Each event is an 8-byte word which indicates the pixel location, event time-stamp, and event polarity. This corresponds to 11.4MB/s or 114MB over 10 seconds. Combined with the data from the intensity images the DAVIS creates 122.2MB, which is about 10% of the amount of data from a standard high-speed camera. It also has the added benefit of not requiring compression, since the DVS acts as an analog compression device that only outputs data when the scene changes sufficiently.

This means the DAVIS camera could film 10 times as long and create the same amount of data as the high-speed camera, which is crucial when it is difficult to predict when the thing you want to film will happen. In addition, when reconstructing the high speed video, the entire volume does not need to be reconstructed, it can be windowed down to the point of interest. Furthermore, high-speed cameras have half the dynamic range of event-based cameras, meaning high dynamic range scenes, which would be saturated by high-speed cameras, can still be imaged and reconstructed using the DAVIS camera. The DAVIS camera is also a simple CMOS sensor which

does not require much power (180mA at 5VDC), which means it could be shrunk to the size of a cellphone camera sensor and have a comparable cost.

All in all, this means the DAVIS camera has more robust operating conditions, creates a fraction of the data, can be miniaturized, and costs a fraction of high-speed cameras. The only trade-off is the computation required to reconstruct the video from events. High-speed cameras are already used in many applications across various industries, but their limitations prevent wider adoption. With the advances offered by event-based cameras, it is possible that high-speed cameras could be in the pocket of everyone with a cellphone and could be used in industries and applications currently unimagined.

This work endeavors to take the event stream and a reference intensity frame, measured by a DAVIS camera [1], and reconstruct high-speed video using the Plug & Play algorithm. The algorithm allows the user to choose a denoiser algorithm to replace the prior model of the MAP estimate, saving the user time and effort as the denoisers can be changed at will, rather than requiring an explicit prior model derivation [2]. This allows the user an easier method to solve inverse problems, while giving them more flexibility to trade image quality and computational complexity.

2. RELATED WORKS

There has been a handful of attempts within the past few years to develop algorithms to estimate intensity images from event data to allow more conventional methods to display and process such data. Kim jointly estimated the motion of the event-camera and the intensity image to create an intensity map of the space for use in self-localization and mapping [3], while Reinbacher formed the events into a 3D spatio-temporal manifold and regularized the manifold to estimate the scene intensity purely from events [4]. However, without knowing the initial intensity of the scene it is challenging to estimate sharp edges and texture from noisy events, as seen in the figures presented by Reinbacher. On the other hand, Brandli realized the issue that stems from the absence of initial intensities and attempted to estimate the event intensities from the intensity frames of the DAVIS camera [5]. However, these intensity estimates are noisy and the pixels do not see uniform change in intensity, which results in streaking effects and ghost images in the reconstructed images. Shedligeri took a different approach and used a neural network to warp one intensity image to the next by estimating the depth and pose of the camera using the events and intensity images then performed the warping operation. They were able to arbitrarily increase the frame rate by sampling the warped image periodically as it progresses from one intensity image to the next [6]. However, in high dynamic range environments, warping is not sufficient as the intensity image will be clipped at its maximal value while the event-camera detects edges in regions that are flat according to the active pixel sensor (APS) portion of the DAVIS camera.

3. THE EVENT CAMERA

3.1 DAVIS Model

To model the operation of the DAVIS camera we must first define some variables corresponding to the two sensors in each pixel, the Dynamic Vision Sensor (DVS) and the Active Pixel Sensor (APS).

To start, let us define the unknown image that the DAVIS camera will sense as $\tilde{x}$. Let $\tilde{x}_{s,n}$ be the pixel at location $s \in S$ in the sensor array and integer discrete time $n \in \mathbb{Z}$, using some fixed sampling rate. Furthermore, let $N = |S|$ be the number of pixels in S. As previously stated, an event camera asynchronously measures contrast over time and only indicates whether the intensity has gone up or down beyond the threshold intensity stored in the sensor. For this reason, the DVS portion of the camera measures $\tilde{x}_{s,n}$ and then converts it to the logarithmic domain, as temporal contrast can be measured by subtraction rather than multiplication. To denote this, let us define $x_{s,n} = \log(\tilde{x}_{s,n})$. When measuring temporal contrast the DVS must store the log intensity after an event so that it may be compared to the current log intensity, which we will denote as x_{s,n_i+r}, where $n_i + r$ is the time after the refractory period, denoted r, of the i-th event occurred. If the absolute difference of $x_{s,n}$ and x_{s,n_i} is greater than the contrast sensitivity of the DVS, then an event occurs; let us denote this operation as $G(\Delta)$. Let us define $G(\Delta)$ as

$$G(\Delta) = \begin{cases} 1, & \Delta \geq t \\ -1, & \Delta \leq -t \\ \emptyset, & \text{otherwise} \end{cases}$$

Where $\Delta = x_{s,n} - x_{s,n_i}$ and t is the contrast sensitivity of the camera. The output of $G(\Delta)$ is the polarity of the event. The event stream is a list of events that are output

sequentially from the camera as a 3-tuple, indicating the pixel position in the array, denoted as s_i, the discrete time of the event, denoted n_i, and the polarity, denoted p_i, where i is a natural number stating the event's location in the list and $p_i \in \{-1, 1\}$. This can be more succinctly written as $\{E_i\} = \{s_i, n_i, p_i\}$.

To define the APS portion of the sensor let $\tilde{y}$ be the image captured by the APS. $\tilde{y}$ is measured at every T samples, where T is an integer. We use the APS to impose the constraint that $\tilde{x}_{s,mT} = \tilde{y}_{s,m}$, $\forall m$. Also, we assume that there is no lens blur or additive noise from the sensor.These definitions are visualized in 3.1.

Using the events alone results in lower quality images, as scene texture that is not high contrast is lost, and a mechanical shutter to trigger all the pixels would be required, or sufficient time would need to pass in order to have an estimate of the initial scene intensity, which is impractical and leads to poorer performance. Instead, by combining an APS with the DVS in one pixel, as demonstrated by the DAVIS camera, it is possible to have an initial intensity frame, which we will leverage to estimate subsequent intensity frames from the event stream using Plug & Play. But first we must formulate the forward model, which is laid out in section 4.2.

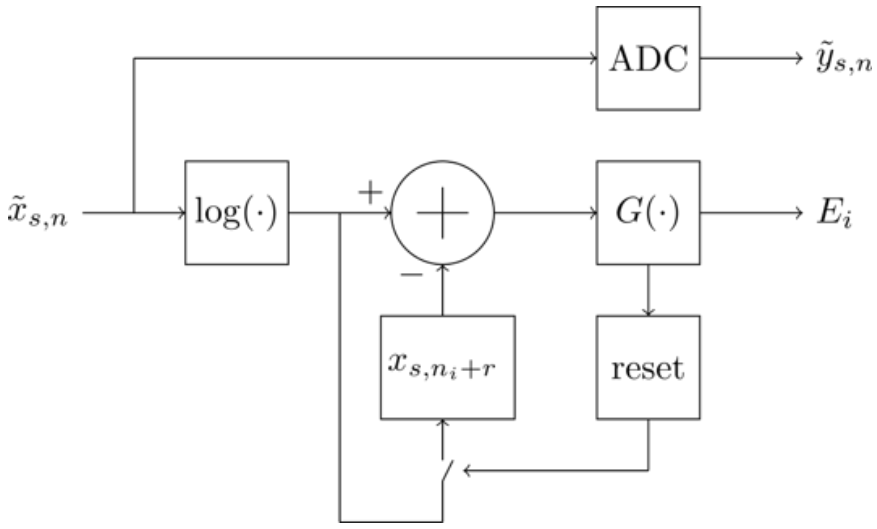


Fig. 3.1.: Model of the DAVIS Camera

3.2 Additional Camera Specifications

To further expound the specifications of the DAVIS camera: the APS array outputs intensity frames synchronously (about 25 frames per second), the frames' resolution is 240x180, the frames are 8-bit grayscale images, and it has a dynamic range of about 55dB. The pixels are rather large at 18.5μm $\times$ 18.5μm. The DVS is asynchronous, lending itself to its high dynamic range of about 130dB and its high temporal resolution of about $1\mu s$ between pixels and $333.3\mu s$ within each pixel. The minimum contrast sensitivity is 11%, which means edges of contrast lower than that will not be captured. Each event is output into a list as an 8-byte word. The DAVIS camera draws about 0.9W when in operations, or 180mA at 5VDC, and has a small form factor of H 56mm x W 62mm x D 28mm.

This is great news because the APS array is essentially sampling at a low temporal resolution to save on storage, and the DVS array fills in the gaps, outputting how the scene changed between samples with minimal data. However, for the DAVIS camera to be useful as a high speed camera, the intensities corresponding to the events must be estimated. Chapter 2 introduced some of the methods used to attempt to estimate intensities from events, but there is more that can be done to solve the inverse problem, namely Plug and Play.

4. INTENSITY IMAGE ESTIMATION FROM EVENTS

Inverse problems are a generic class of problems where measurements are taken and used to estimate the input to a physical phenomenon or system. These phenomena and systems have equations that describe them. These problems are ill-posed and impossible to solve without imposing some assumptions and constraints on the solution. In the case of image processing, inverse problems are those which attempt to estimate the "true," or unknown, image from imperfect measurements, such as reconstructing 3D volumes measured by a MRI machine, removing noise and blur from a picture, or image super-resolution similar to the zoom and enhance trope seen in crime movies and television. Posed in a probabilistic framework, if we denote the measured data as y and the unknown data as x, we can try to maximize the probability of the unknown image given the measurements. This is known as the Maximum A Posteriori (MAP) Estimate and can be written

$$\hat{x} = \arg\max_x p(x|y).$$

This equation is not very informative in itself, instead we will use Bayes' Theorem, which results in

$$\hat{x} = \arg\max_x \frac{p(y|x)p(x)}{p(y)}.$$

In order to simplify things we take the negative log of this expression. This turns multiplication into addition and allows us to use standard optimization algorithms to minimize the expression. Furthermore, $p(y)$ is not a function of x, thus it is only a constant term that ensures the cumulative distribution function is valid, which means we can drop it from the optimization problem. This results in

$$\hat{x} = \arg\min_x \left\{ -\log p(y|x) - \log p(x) \right\},$$

$p(y|x)$ is referred to as the forward model and $p(x)$ is referred to as the prior model. In plain English, the forward model explains how likely it is that the measured data resulted from the estimate of the unknown image. The prior model is formulated in such a way that assumptions about the characteristics of the unknown image are enforced by giving them a higher probability. An example of such, is that we assume real images do not look like white noise, they have smooth regions, patterned textures, and clear edges. Therefore, when minimizing this expression the optimal solution is one which balances fitting the data with conforming to our assumptions of how images look. For this application we will use the measurements from the DAVIS camera and solve the inverse problem to temporally super-resolve the data from about 25 frames per second to an arbitrarily high frame rate up to 1 million frames per second, theoretically.

In practice, the intricacies and wide variety of real images makes it challenging to formulate a good prior model that encompasses all conceivable images. This motivated the development of the Plug and Play algorithm. We will use Plug and Play to reconstruct high-speed video from the DAVIS frames and events by formulating a forward model and picking an off-the-shelf denoiser.

4.1 Plug and Play

The Alternating Direction Method of Multipliers (ADMM) is a variant of the augmented Lagrangian method, also known as the method of multipliers, which is an algorithm for constrained optimization. The form of the equation is

$$\min_{x,y} f(x) + g(y), \quad \text{subject to} \quad x = y,$$

$f(\cdot)$ and $g(\cdot)$ are proximal maps of the forward and prior models, respectively. The ADMM algorithm has only 3 steps that are iterated until the convergence criteria is met

Algorithm 1: ADMM

Initialize: $\sigma^2 > 0, v \leftarrow \text{something}, x \leftarrow v, u \leftarrow 0$

while not converged **do**

$x \leftarrow \arg\min_x f(x) + \frac{\sigma^2}{2}||x - v + u||^2$

$v \leftarrow \arg\min_v g(v) + \frac{\sigma^2}{2}||v - x - u||^2$

$u \leftarrow u + x - v$

end while

The Plug and Play algorithm simply replaces the proximal mapping of the prior model with an off-the-shelf denoiser

Algorithm 2: Plug & Play

Input: Denoiser, λ

Initialize: $\sigma^2 > 0, \rho \leftarrow 1, \sigma_H \leftarrow \sqrt{\lambda/\rho}, v \leftarrow \text{something}, x \leftarrow v, u \leftarrow 0$

while not converged **do**

$x \leftarrow \arg\min_x f(x) + \frac{\sigma^2}{2}||x - v + u||^2$

$v \leftarrow \text{Denoiser}_{\sigma_H}(x + u)$

$u \leftarrow u + x - v$

end while

With this algorithm we have the framework we need to estimate the event intensities to reconstruct the video in high speed, however, to actually implement the reconstruction algorithm, the forward model must still be formulated and a denoiser must be chosen.

4.2 The Forward Model

In order to formulate the forward model we must make a few more definitions in addition to those declared in section 3.1. Let us first reiterate the salient definitions then add a few more. Let us define the unknown image that the DAVIS camera will sense as $\tilde{x}$. Let $\tilde{x}_{s,n}$ be the pixel at location $s \in S$ and integer discrete time $n \in \mathbb{Z}$, using some fixed sampling rate. Let $\tilde{y}$ be the image captured by the active pixel sensor (APS) portion of the DAVIS camera, and let $\tilde{y}_{s,n}$ be the intensity measured at pixel $s \in S$ at time $n = mT$. From the list of events let us take p_i for all i and let us sum the events for each pixel $s \in S$ in the discrete time interval $[T_{n-1}, T_n]$ and define this integer as $p_{s,n}$. Due to the loss of information during the refractory period and the unknown contrast step size corresponding to each event, we have defined a simplified model, which we will refer to as the uniform quantization forward model. This model assumes that each event denotes a uniform step in contrast, thus, each event alters the quantization level of the image. Let us define the boundary between quantization levels as $\tilde{q}$. At each pixel $\tilde{q}_{s,n} = \alpha^{p_{s,n}} \tilde{q}_{s,n-1}$. We have constrained the problem so that at time $n = mT$, $\tilde{x}_{s,n} = \tilde{y}_{s,n}$. For $n \neq mT$

$$\frac{\tilde{q}_{s,n}}{\alpha} \leq \tilde{x}_{s,n} < \tilde{q}_{s,n}$$

However, because contrast is linear in the logarithmic domain, we will convert the linear domain variables into the logarithmic domain to improve computation and notational simplicity. Let us define the log domain of the unknown image as $x_{s,n} = \log(\tilde{x}_{s,n})$, the log intensity image as $y_{s,n} = \log \tilde{y}_{s,n}$ and the log quantization boundaries $q_{s,n} = \tilde{q}_{s,n}$. This means that at each pixel $q_{s,n} = q_{s,n-1} + cp_{s,n}$ where $c = \log \alpha$ and for $n \neq mT$

$$q_{s,n} - c \leq x_{s,n} < q_{s,n}$$

With these definitions in place, we may define the proximal map of the forward model

$$F_{q,y}(v - u) = \arg\min_x \left\{ f(x, q, y) + \frac{1}{2\sigma^2} ||x - v + u||^2 \right\},$$

$f(x,q,y) = 0$ when x is consistent with q and y measured by the DAVIS, and $f(x,q,y) = \infty$ if it is inconsistent. Next we define the set

$$Q(q,y) = \min_x \{x \in \mathbb{R} : f(x,q,y) < \infty\}.$$

It follows that the proximal map has the form

$$x = F_{q,y}(v-u) = \min_{x \in Q(q,y)} ||x - v + u||^2.$$

This can be stated as $Q(q,y)$ being the set of all possible images, $v-u$, that are consistent with the DAVIS measurements, q and y, and x is the closest point in $Q(q,y)$ to $v-u$. x can be computed at every pixel for every time $n = mT$ as $x_{s,n} = y_{s,n}$. x can be computed at every pixel for every time $n \neq mT$ as

$$\begin{aligned} x^*_{s,n} &\leftarrow v_{s,n} - u_{s,n} \\ x_{s,n} &\leftarrow \text{clip}(x^*_{s,n}, [q_{s,n} - c, q_{s,n}]). \end{aligned}$$

However, this does not work well because the events do not actually take uniform step sizes. The reason for this can be visualized in figure 4.2. Despite there being 2 up events and 3 down events the intensity has actually gone up from the intensity at the first event. The loss of information during the refractory period and the nonuniform quantization step sizes at each event are poorly modeled by clipping the pixels within their quantization boundaries.

We compensate for this by adding a quadratic penalty term to the values outside of the quantization boundaries, which can be written as

$$x^*_{s,n} \leftarrow v_{s,n} - u_{s,n}$$

$$x_{s,n} \leftarrow \begin{cases} \frac{x^*_{s,n} + \sigma^2 q_{s,n}}{1+\sigma^2}, & x^*_{s,n} > q_{s,n} \\ x^*_{s,n}, & q_{s,n} - c \leq x^*_{s,n} \leq q_{s,n} \\ \frac{x^*_{s,n} + \sigma^2 (q_{s,n} - c)}{1+\sigma^2}, & x^*_{s,n} < q_{s,n} - c \end{cases}$$

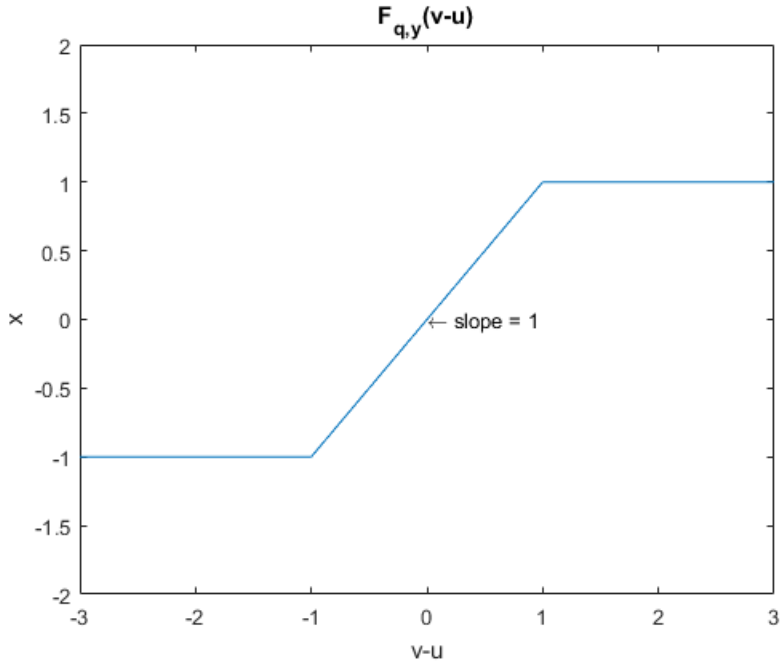


Fig. 4.1.: Proximal Mapping of the Forward Model

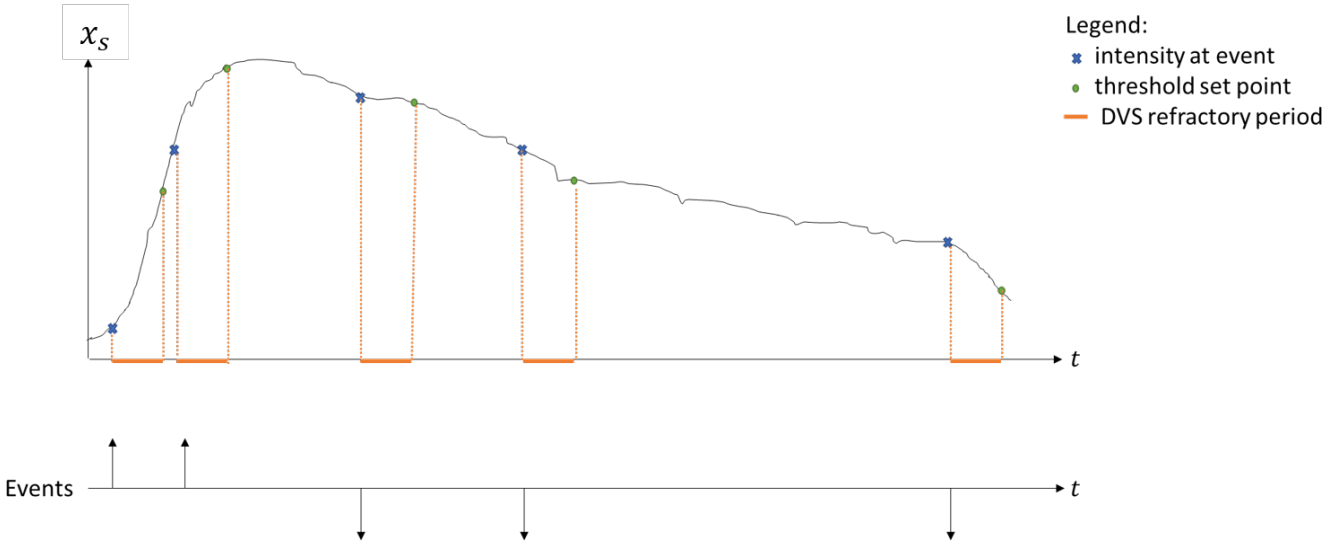


Fig. 4.2.: Example of Intensity and Events from One Pixel

Essentially this is a "soft clip," as seen in 4.3, where the $x^*_{s,n}$ pixels that fall in the range of the current and previous quantization levels keep their value, and values

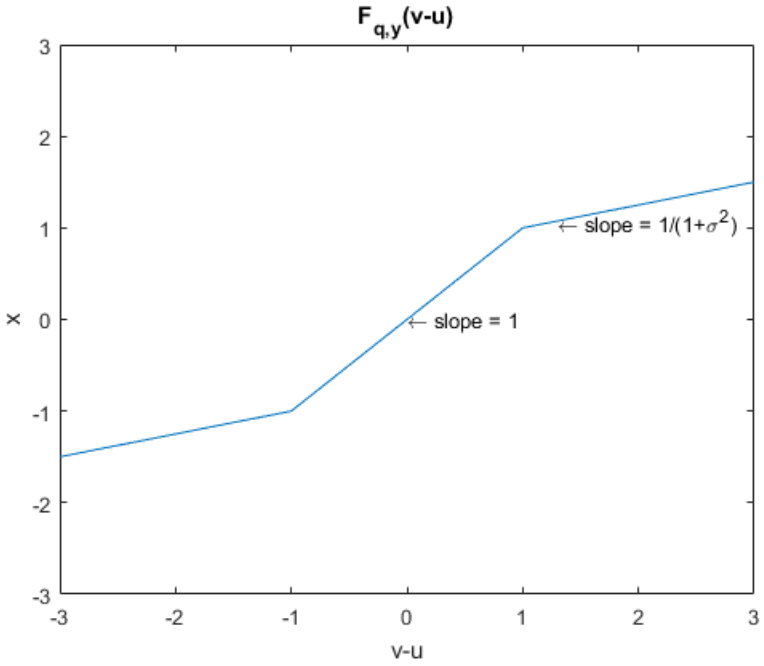


Fig. 4.3.: Proximal Mapping of the Forward Model

that fall outside the range are attenuated by a weighted average with the quantization boundary. This give the algorithm more freedom to compensate for the limitations of the uniform quantization model and results in a better reconstruction.

4.3 Denoisers

Denoising is a fundamental topic in image processing. Digital sensors suffer from many forms of noise that corrupt the image, making fine details hard to see and results in images that are unappealing to the common viewer. Thus, there has been a cornucopia of denoising algorithms and advances in hardware to mitigate the deliterious effects of noise on images. This bodes well for Plug & Play because there are many options available and gives the engineer a trade space they can optimize for their desired application. There are a few types of denoisers that can be divided into groups by how they perform the denoising operation. There is some overlap between the groups in the following statements and the denoisers mentioned are not

an exhaustive list, but the denoisers are grouped based upon the main intent of the author. The first denoiser taught to young electrical engineering students is the Linear Time-Invariant (LTI)/Linear Space-Invariant (LSI) filter. In the case of the LSI filter, a mask is convolved with the image to smooth it and reduce the high spatial frequency noise. However, these filters cannot discriminate between noise and edges, resulting in images that are blurry, which is visually unappealing.

The next type of denoiser is the statistical filter. This includes a wide array of filters with different performance levels such as minimum mean square error (MMSE) filters like the Wiener filter, the median and weighted median filters, which, as the name suggests, finds the median of the pixels within the window. The non-local means filter is a more computationally expensive algorithm, which finds the weighted average for each pixel by weighing all other pixels in the image by their similarity to the pixel being processed [7]. Arguably the best off-the-shelf denoiser today is Block Matching and 3D Filtering (BM3D). As the name suggests the algorithm splits the image into blocks and matches them if the cost function is below the threshold. Once the image has been divided into 3D cubes of the blocks the cubes are then filtered by a Wiener filter and then recombined into the "true" image by a weighted average of the local estimates [8].

Edge-preserving filters are yet another type of denoiser. Anisotropic diffusion poses the image denoising problem in the form of a partial differential equation that is a general case of the heat equation. It is solved by iteratively applying an approximation of the diffusion equation to the image and the level of smoothing is controlled by the diffusion coefficient [9]. The bilateral filter performs a weighted average of neighboring pixels by their Euclidian distance from the pixel being updated and by the neighboring pixels' similarity in intensity [10]. By considering both the distance and intensity the edges of the image are preserved. The guided filter formulates the problem as the optimization of the coefficients of an affine transform of the noisy image [11]. Because the solution is in the form of an affine transform the gradients of the noisy and denoised images are approximately the same, up to a scaling coeffi-

cient. The recursive filter iteratively performs a weighted average between the noisy image and the last update of the denoised image [12]. By manipulating the recursion equation it can be seen that the weighting coefficient is essentially the same weight computed by the bilateral filter.

In recent years with the growing success and interest in machine learning, specifically deep learning, engineers have attempted and succeeded in developing state of the art denoisers from deep neural networks (DNN) [13]. The concept is fairly simple, give the computer thousands to millions of clean and degraded image pairs, and have the computer learn what noise "looks like" in the image. Then when a new noisy image is given to the DNN, it uses what it has learned about noise to quickly estimate the noise-free image. This has only become possible in the past few years with the development of powerful graphical processing units(GPUs) for HD gaming and 3D model rendering. Despite the state-of-the-art performance of deep learning, there are no free lunches. The drawback of this type of denoiser is the training involved. The plethora of image must be collected and labelled, then the training occurs, which depending on the architecture of the DNN, the amount of training data, and how many GPUs are used, can take hours to weeks of training.

Clearly there are many options when it comes to denoising an image. This greatly benefits the Plug and Play algorithm, as the denoiser can be chosen specifically for certain types of noise found in the forward model, for restored image quality, or for computational ease. With the ability to drop any denoiser code into Plug and Play, the user now has a trade space to balance quality and performance, especially when considering implementation on embedded systems or real-time processing. In this application we will use a subset of the denoisers that are sensitive to the noise level and that have code which is readily available online to be downloaded and plugged in without any modifications to the code.

5. IMPLEMENTATION

The algorithm was implemented in MATLAB r2017a/b and all of the denoisers were either from the image processing toolbox or toolboxes downloaded from the publisher of the denoiser. The data sets that were used to implement and test the algorithm are from the Event-Camera Dataset and Simulator [14], which will later be discussed at greater length.

The data sets consist of a series of images from the APS, and a text file corresponding to the list of events captured by the DVS. This data was downloaded and converted into a MAT-file to be easily loaded and manipulated in MATLAB.

This algorithm gives a lot of leeway to the user to trade performance with computation, however, it comes at the additional cost of needing to hand-tune the hyperparameters. Without tuning the contrast step size, scaling the convergence constraint of the forward model, and setting the regularization level, the results can be visually unappealing and noisy, or over-regularized to the point of being nearly one gray level.

5.1 Computational Complexity

The computational complexity of the image can be estimated by inspecting algorithm 3. There is a for loop that estimates each new frame with a while loop nested inside of it to perform the Plug & Play optimization. Within the while loop there can be a computationally simple or complex denoiser. All of the denoisers used are $O(N)$ [12, 15], however denoisers such as NLM [7] and BM3D [8] have large constant factors while light-weight filters such as the Wiener filter [16] and the guided filter [11] have small constant factors. Combining this complexity with the two loops for frames and optimization results in a complexity of $O(mkN)$, where m is the number of frames being estimated, and k is the number of iterations until the image

Algorithm 3: DAVIS Reconstruction

Input: $y, p, m, T, \lambda, \sigma^2, c$, Denoiser

Initialize: $\epsilon > 0, \rho \leftarrow 1, \sigma_H \leftarrow \sqrt{\lambda/\rho}, \gamma \leftarrow 2.4$

for all $s \in S$ **do**

$q_{s,mT} \leftarrow y_{s,m}$

$x_{s,mT} \leftarrow y_{s,m}$

end for

for new frames $n = mT + 1 : (m+1)T$ **do**

for all $s \in S$ **do**

$v_{s,n} \leftarrow y_{s,n-1}$

end for

$u \leftarrow 0$

repeat

$x^* \leftarrow v - u$

for all $s \in S$ **do**

$$x_{s,n} \leftarrow \begin{cases} \frac{x^*_{s,n} + \sigma^2 q_{s,n}}{1+\sigma^2}, & x^*_{s,n} > q_{s,n} \\ x^*_{s,n}, & q_{s,n} - c \le x^*_{s,n} \le q_{s,n} \\ \frac{x^*_{s,n} + \sigma^2 (q_{s,n} - c)}{1+\sigma^2}, & x^*_{s,n} < q_{s,n} - c \end{cases}$$

end for

$v \leftarrow \text{Denoiser}_{\sigma_H}(x + u)$

$u \leftarrow u + (x - v)$

until $||x - v|| < \epsilon$ or for 20 iterations

end for

$\tilde{x} \leftarrow 255 \left(\frac{\exp(x) - 1}{e - 1} \right)^{\frac{1}{\gamma}}$

converges. However, to reduce complexity, the DAVIS reconstruction algorithm caps the optimization steps at 20 iterations, thereby making the complexity $O(mN)$. Furthermore, for reasonable reconstructions, the number of pixels should outnumber the

number of frames to be reconstructed, resulting in a linear time algorithm, $O(N)$. Performance time is then based on the corresponding constant factors of the denoiser and the number of frames.

5.2 Data Sets and Ground Truthing

As previously mentioned, the data sets used are from the Event-Camera Dataset and Simulator [14] repository. The data sets were recorded and ground-truthed for the purposes of pose estimation, visual odometry, and SLAM, as that is a much larger area of research in event-based sensing compared to image reconstruction. There are currently no image reconstruction focused event-based data sets that have high-speed video ground truth recordings. To further complicate matters, DVS's have about twice the dynamic range of APS cameras, therefore multiple high-speed cameras would need to share an aperture and record simultaneously at different contrast levels to capture the entire dynamic range of the DVS. This becomes an issue in high-contrast environments, for example, when the sun is behind buildings in the background. APS pixels sensing the background region will be saturated, but the DVS can still sense the edges of the the buildings. This is demonstrated by the "outdoors_running" data set.

Without a high-speed video ground truth, the next option is to consider the APS frames from the DAVIS as a makeshift ground truth. To clarify, the proposed algorithm will take $\tilde{y}_{s,n-1}, \forall s \in S$ as the input, where $n-1 = mT$, estimate the desired number of frames up to time $(m+1)T$ and then perform a PSNR calculation between $\tilde{y}_{s,n}$ and the final frame estimate $\tilde{x}_{s,n}, \forall s \in S$. This is the metric we will use to assess the reconstruction quality of the proposed algorithm. However, as previously stated, $\tilde{y}$ can be saturated in high-contrast environments and reconstructions will be penalized for showing structures that the APS cannot see. Furthermore, the APS can only record at about 25 frames per second, and fast movement of objects in the scene or rapid camera movement will cause motion-blur, which is unacceptable for the ground

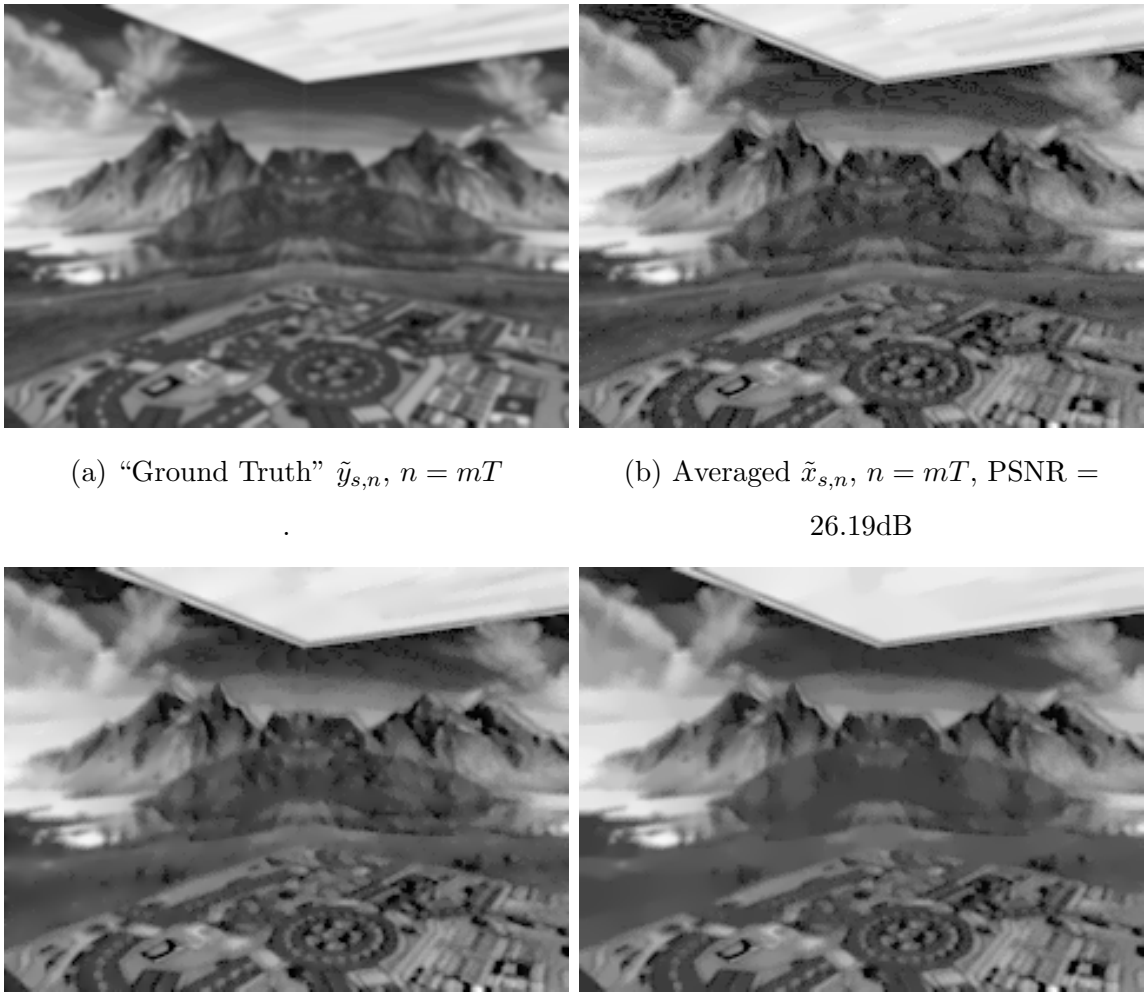
truth. This is clearly a problem for evaluating the performance of the algorithm, but it does not make it impossible.

Therefore, for the purpose of quantifying the performance of the proposed DAVIS reconstruction algorithm, the data set, and frames to be reconstructed within the data set, must be carefully chosen to avoid the aforementioned issues. One option is to use a simulation of a scene and the camera with no noise, thus rendering $\tilde{y}$ the same as $\tilde{x}$. Another option is to use a simplistic data set that does not present issues to ground-truthing. The "shapes_translation" data set was selected to benchmark the proposed algorithm for this reason, as there are no dynamic range mismatch issues, as well as there being minimal to no motion-blur, depending on the frames being reconstructed. The proposed algorithm can handle any scenery, motion, and dynamic range, the simplistic data set used is purely due to a lack of high-speed video ground truth data. To reinforce this assertion we also performed a reconstruction of the "outdoors_running" data set to qualitatively assess the performance of the reconstruction algorithm.

5.3 Results

The first assessment of the algorithm used a synthetic data set titled "simulation_3walls." As the name implies, the data set is a simulation of two walls with a mountain terrain texture and a floor with a cartoon city area rug. The contrast step size, noise level, and the scaling term that forces ADMM to converge were all hand-tuned for maximum PSNR. The temporal resolution of the video was super-resolved by 5 times the original sample rate to test the algorithm's ability to upsample the video. We estimated $\tilde{x}_{s,n}$ where n = frame70 and compared it to the corresponding APS frame $\tilde{y}_{s,n}$. From figure 5.1, we see that all of the algorithms resulted in a PSNR in the 25 to 26dB range, with the best performance coming from BM3D at 26.2dB, while the much simpler average of $q_{s,(m+1)T-1}$ and $q_{s,(m+1)T}$ had a PSNR of 26.18dB.

They all suffered from the same streaking artifact, which is caused by the uniform quantization level assumption.



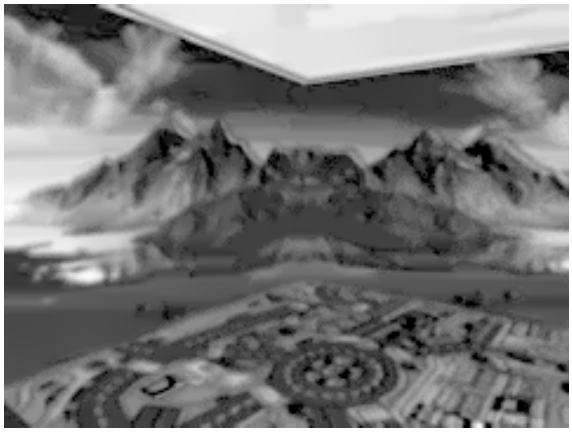
(a) "Ground Truth" $\tilde{y}_{s,n}$, $n = mT$.

(b) Averaged $\tilde{x}_{s,n}$, $n = mT$, PSNR = 26.19dB

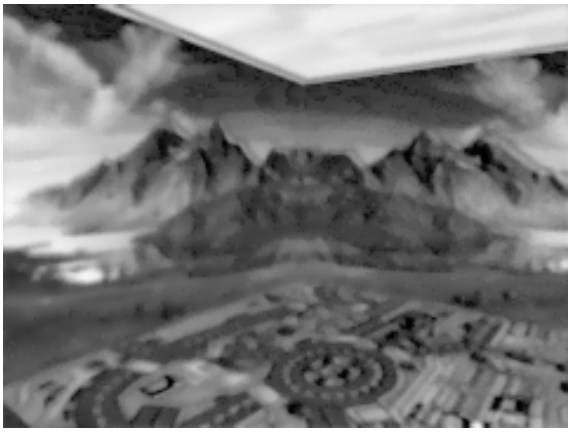
(c) Wiener Filter $\tilde{x}_{s,n}$, $n = mT$, PSNR = 25.99dB, λ=2e-6

(d) Guided Filter $\tilde{x}_{s,n}$, $n = mT$, PSNR = 25.99dB, λ=2e-6

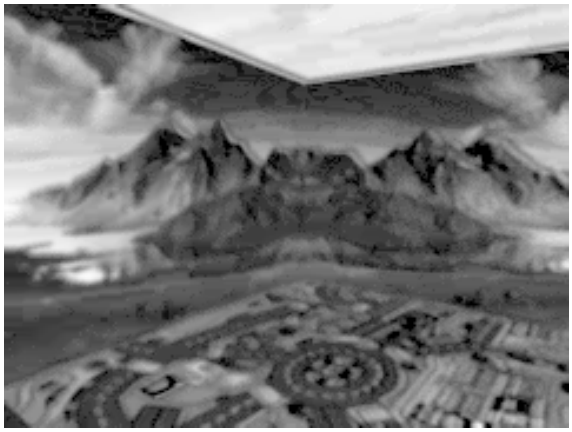
Fig. 5.1.: Reconstruction of Synthetic Data Set "simulation_3walls"



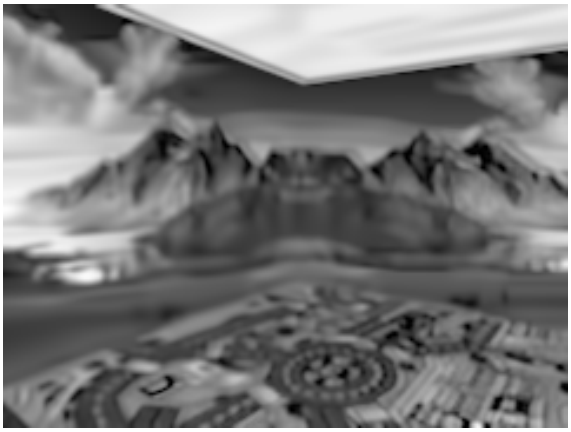
(e) Recursive Filter $\tilde{x}_{s,n}$, $n = mT$, PSNR = 25.85dB, λ=1.7e-7



(f) TV Deconvolution $\tilde{x}_{s,n}$, $n = mT$, PSNR = 26.1dB, λ=2.1e-5



(g) NLM Filter $\tilde{x}_{s,n}$, $n = mT$, PSNR = 25.83dB, λ=5e-8

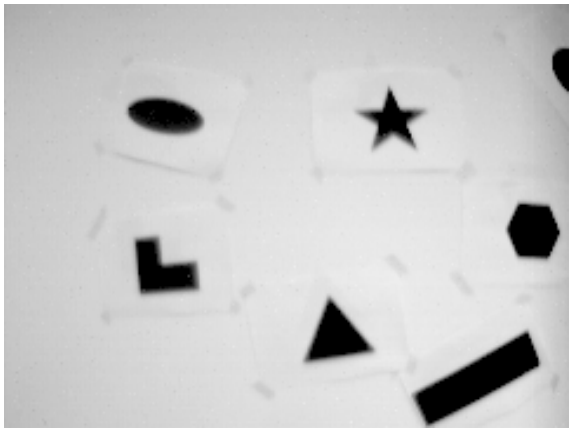


(h) BM3D Filter $\tilde{x}_{s,n}$, $n = mT$, PSNR = 26.2dB, λ=1.1e-7

Fig. 5.1.: Reconstruction of Synthetic Data Set "simulation_3walls"

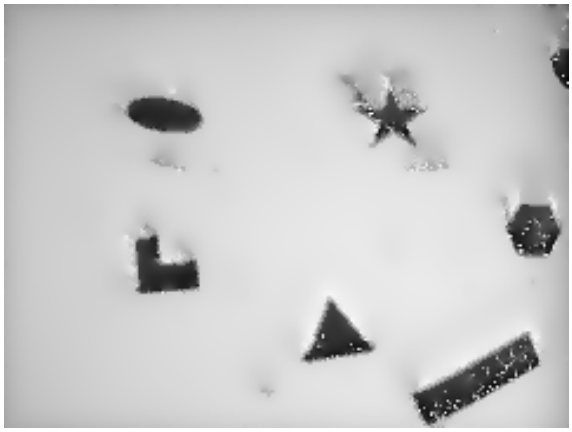
In figure 5.2 we see a marked improvement over the synthetic data set. Again, the contrast step size, noise level, and the scaling term that forces ADMM to converge were all hand-tuned for maximum PSNR, and the video was super-resolved by 5 times the original sample rate. With the noise from the DAVIS camera, and imperfect edge alignment that sometimes misses events that would be expected to actually occur, the averaged case results in far more noise and streaking artifacts than the reconstructed frames. The frame reconstructed for quality assessment was $n = 1234$. The pieces of

tape on the corners of the pages and the shadows from the pages not lying flat are not high enough contrast and do not create events, so they are filtered out as noise, but the shapes are kept intact. There are still some slight blurring artifacts from the uniform quantization assumption, but all in all, the Plug & Play algorithm performs high quality reconstructions.

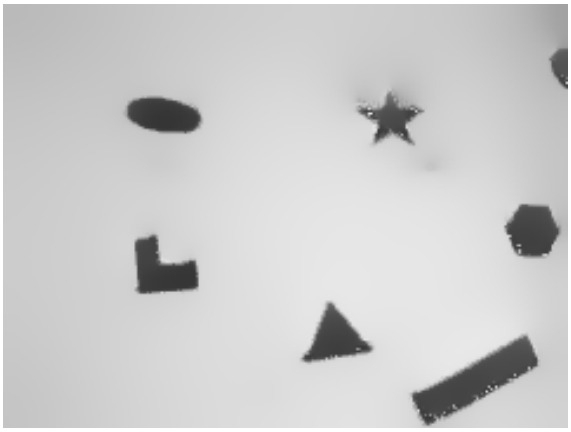


(a) "Ground Truth" $\tilde{y}_{s,n}, n = mT$.

(b) Averaged $\tilde{x}_{s,n}$, $n = mT$, PSNR = 20.45dB



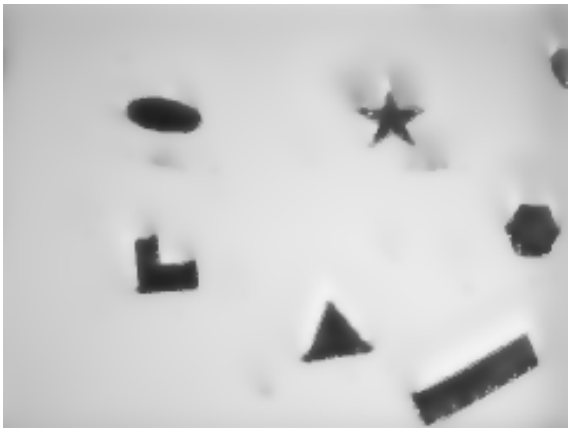
(c) Wiener Filter $\tilde{x}_{s,n}$, $n = mT$, PSNR = 23.25dB, λ=4e-5



(d) Guided Filter $\tilde{x}_{s,n}$, $n = mT$, PSNR = 24.93dB, λ=1.3e-4

Fig. 5.2.: Reconstruction of Real Data Set "shapes_translation"

(e) Recursive Filter $\tilde{x}_{s,n}$, $n = mT$, PSNR = 23.22dB, λ=1.51e-5



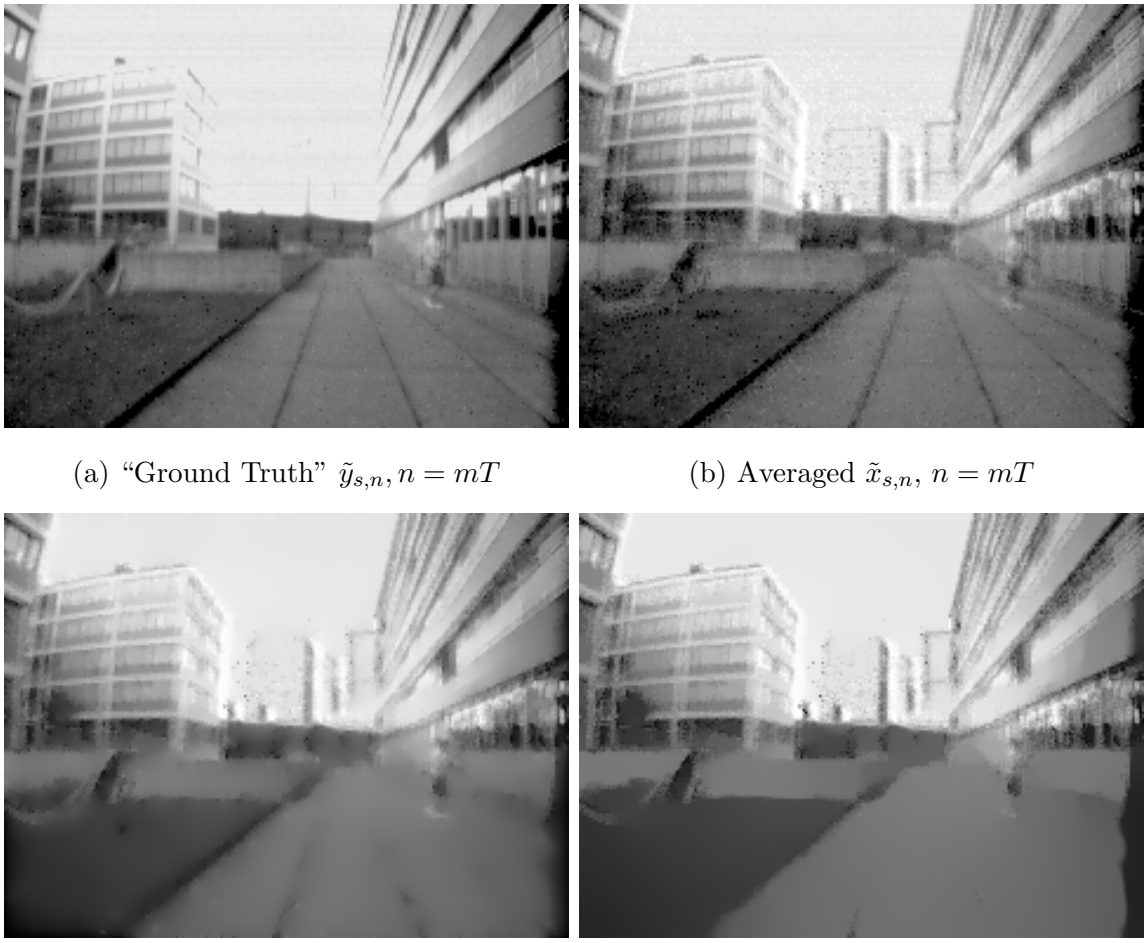
(f) TV Deconvolution $\tilde{x}_{s,n}$, $n = mT$, PSNR = 24.55dB, λ=8e-5

(g) NLM Filter $\tilde{x}_{s,n}$, $n = mT$, PSNR = 25.85dB, λ=5.25e-5

(h) BM3D Filter $\tilde{x}_{s,n}$, $n = mT$, PSNR = 26.38dB, λ=1.3e-4

Fig. 5.2.: Reconstruction of Real Data Set "shapes_translation"

To assess the reconstruction algorithm in a more realistic environment we reconstructed a high dynamic range scene of the DAVIS camera filming an urban environment while being held by a jogging researcher. No PSNR is given because the "ground truth" does not include the buildings being reconstructed in the background as those pixels are saturated in the APS.



(a) "Ground Truth" $\tilde{y}_{s,n}, n = mT$

(b) Averaged $\tilde{x}_{s,n}$, $n = mT$

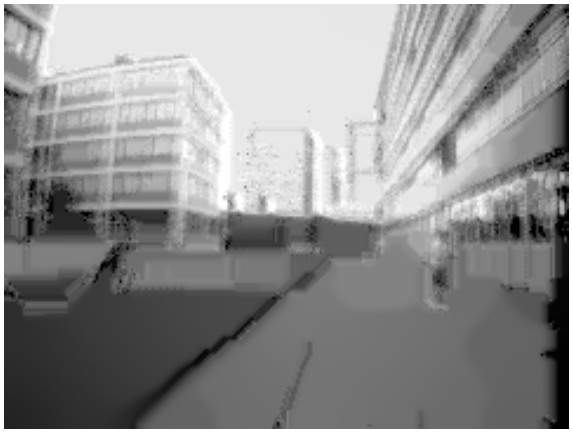
(c) Wiener Filter $\tilde{x}_{s,n}$, $n = mT$, λ=3e-6

(d) Guided Filter $\tilde{x}_{s,n}$, $n = mT$, λ=4e-6

Fig. 5.3.: Reconstruction of Real Data Set "outdoors_running"

(e) Recursive Filter $\tilde{x}_{s,n}$, $n = mT$, λ=5e-7

(f) TV $\tilde{x}_{s,n}$, $n = mT$, λ=2.5e-5



(g) NLM Filter $\tilde{x}_{s,n}$, $n = mT$, λ=4e-7

(h) BM3D Filter $\tilde{x}_{s,n}$, $n = mT$, λ=2e-7

Fig. 5.3.: Reconstruction of Real Data Set "outdoors_running"

5.4 Observations

It is apparent that the DAVIS reconstruction algorithm adds no improvement over hand-tuning the contrast step-size and computing the average between the last two frames estimates in the simulated data set. In fact, for the simulated data set, it is a waste of time and performs worse than the average of the quantization boundaries. This is likely due to the absence of noise in the simulation and there are no false positive events or false negative events. It is also possibly due to the simulation being

fairly homogeneous in contrast step sizes throughout the image and the denoiser adds too much bias to the low contrast textured regions, for which the DVS does not output events. Another possibility is that the model of the camera used in the simulated data does not fully represent the physics of the camera.

Reconstructing the shapes is far more successful for the proposed algorithm. All of the denoisers were quite successful at removing the motion blur seen in the averaged estimate as well as the noise from the imperfect contrast step sizes and noise. The edges of the paper and the tape were removed in all of the reconstructed estimates due to those edges not being strong enough to trigger events from the DVS, which resulted in them being washed out into the background. Computational complexity does not guarantee better PSNR, but it trends that way, except in the case of the recursive filter in figure 5.2. Performance would likely be further improved by implementing a spatio-temporal filter that takes the previous frames into account when estimating the current frame.

Qualitatively, the reconstruction of the "outdoors_running" scene in figure 5.3 is quite successful. The algorithm manages to perform a high dynamic range reconstruction of the side of the building on the left-hand side and the buildings in the background, which are saturated in the frames $\tilde{y}_{s,n-1}$ and $\tilde{y}_{s,n}$. There is some slight blurring of the high contrast edges of the buildings due to the quantization step size not being large enough in those regions, but the semantics of the scene are not lost, and playback of the reconstructed video shows the scene panning as the camera moves from the jogging motion. This is a scene in which previously proposed algorithms would likely fail, either due to the lack of intensity information in the case of pure event reconstruction, or due to the background being saturated and not reconstructed in the case of the warping algorithm.

There are a few shortcomings with the proposed algorithm. Hand-tuning hyperparameters can be time consuming and the hyperparameters must be tuned differently not just between data sets, but also between frames if the scene changes drastically. A brute-force fix would be to have a hyperparameter vector that is iterated over until

the settings that maximizes the PSNR is found. This is possible for data sets with ground truth, but tuning the parameters for a new recording would not be possible in this way. Additionally, using a spatial filter makes the reconstruction sensitive to the new frame rate. As the frame rate increases, the regularization must be scaled down, otherwise each successive frame will be more biased than the last, and the final frame could come out as a gray image if overly regularized.

There are many options when considering future work. One first order of business is to develop a robust data set with high-speed camera ground-truthing to better enable measuring performance and apples-to-apples comparisons of image reconstruction techniques. An adaptive control of the weight σ^2 in the uniform quantization forward model would better allow the model to constraint events in neighborhoods of lower contrast to the quantization boundaries, while high contrast neighborhoods would be given more leeway when reconstructing the intensity. Developing a forward model that takes into account the refractory period of the DVS and determining the contrast step size for each event would remove the streaking artifact from the uniform quantization assumption. Automatic hyperparameter tuning would potentially save time, and would be necessary if ever developed to the point of regular consumers using it. As previously stated, spatiotemporal filters are an obvious next step to improve the proposed algorithm. With the development of color DAVIS cameras, research can be done to reconstruct color images from events, perhaps by using the XYZ color space and performing the reconstruction on the luminance channel. It is important to note that these cameras are still more of a test unit than a commercial camera. They have relatively low resolution and large pixel areas. There is still much improvement that can be done on the sensors themselves to make them more comparable to cell phone cameras in size, resolution, and cost, which means they could ultimately replace those sensors and add more capabilities.

6. SUMMARY

Event-based cameras offer a new imaging modality that has the ability to generate high-speed video at a fraction of the cost and data of current high-speed cameras. This ability comes at the cost of computation, which is cheap in comparison. We successfully proposed and demonstrated reconstruction of temporally super-resolved video, at a user-desired framerate. This was done with reference intensity images and asynchronous 1-bit events using the Plug & Play algorithm and an idealized forward model to perform the reconstruction from the data sets. The Plug & Play algorithm offers greater flexibility in quality and computation than Maximum A Posteriori estimates because it does not require the formulation of a prior model, and instead utilizes an off-the-shelf denoiser in the split variable optimization of ADMM. The research area is still new and there are many directions for further work to take to make event-based cameras the next standard in imaging.

REFERENCES

REFERENCES

[1] C. Brandli, R. Berner, M. Yang, S. Liu, and T. Delbruck, "A 240 x 180 130 db 3 us latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, Oct 2014.

[2] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed point convergence and applications," *CoRR*, vol. abs/1605.01710, 2016. [Online]. Available: http://arxiv.org/abs/1605.01710

[3] H. Kim, A. Handa, R. B. Benosman, S.-H. Ieng, and A. J. Davison, "Simultaneous mosaicing and tracking with an event camera," in *BMVC*, 2014.

[4] G. G. Christian Reinbacher and T. Pock, "Real-time intensity-image reconstruction for event cameras using manifold regularisation," in *Proceedings of the British Machine Vision Conference (BMVC)*, E. R. H. Richard C. Wilson and W. A. P. Smith, Eds. BMVA Press, September 2016, pp. 9.1–9.12. [Online]. Available: https://dx.doi.org/10.5244/C.30.9

[5] C. Brandli, L. Muller, and T. Delbruck, "Real-time, high-speed video decompression using a frame- and event-based davis sensor," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, June 2014, pp. 686–689.

[6] P. A. Shedligeri, K. Shah, D. Kumar, and K. Mitra, "Photorealistic image reconstruction from hybrid intensity and event based sensor," *CoRR*, vol. abs/1805.06140, 2018. [Online]. Available: http://arxiv.org/abs/1805.06140

[7] A. Buades, B. Coll, and J. . Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, June 2005, pp. 60–65 vol. 2.

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug 2007.

[9] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, July 1990.

[10] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, Jan 1998, pp. 839–846.

[11] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, June 2013.

[12] Y. Chi and S. H. Chan, "Fast and robust recursive filter for image denoising," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 1708–1712.

[13] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, July 2017.

[14] E. Mueggler, H. Rebecq, G. Gallego, T. Delbrück, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *CoRR*, vol. abs/1610.08336, 2016. [Online]. Available: http://arxiv.org/abs/1610.08336

[15] J. M. Bioucas-Dias, M. A. T. Figueiredo, and J. P. Oliveira, "Total variation-based image deconvolution: a majorization-minimization approach," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 2, May 2006, pp. II–II.

[16] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1990, p. 548.