Event Based Simultaneous Localization and Mapping using Straight Lines

A Project Report

submitted by

ADINARAYANA NUTHALAPATI

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

DEPARTMENT OF Electrical Engineering
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2019

THESIS CERTIFICATE

This is to certify that the thesis titled **Event Based Simultaneous Localization and Mapping using Straight Lines**, submitted by **ADINARAYANA NUTHALAPATI**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bonafide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Kaushik Mitra
Research Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 5^{th} May 2019

ACKNOWLEDGEMENTS

I am very glad to thank my project advisor, Dr. Kaushik Mitra, Electrical Engineering, IITM, for his enduring support, guidance and inspiration. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own, and at the same time the guidance to recover when my steps faltered. I am also thankful to PhD scholar, Prasan Shedligeri, Electrical Engineering, IITM, for his enduring support. A special thanks to Head of the Department (HOD), Electrical Engineering, for providing resources in completing my project. I am also thankful to my family and friends for supporting me in the time of need.

ABSTRACT

Event cameras are bio-inspired vision sensors that output pixel-level brightness changes instead of standard intensity frames. These cameras do not suffer from motion blur and have a very high dynamic range, which enables them to provide reliable visual information during high-speed motions or in scenes characterized by high dynamic range. Event sensor also comes with low power consumption and less bandwidth required for storing and processing event stream, as there is no redundant data. Inspired from these advantages, an event camera is used to carry out SLAM application using line segments as features. Although several straight line based SLAM methods have been proposed using EKF filters, they can not be integrated directly with event cameras because of it's asynchronous nature. We propose a method to perform simultaneous localization and mapping, assuming the camera has linear motion, using line segments as features, with SFM approach.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

DVS	Dynamic Vision Sensor
DAVIS	Dynamic active pixel vision sensor
AER	Address Event Representation
DoF	Degrees of Freedom
SLAM	Simultaneous Localization and Mapping
EKF	Extended Kalman Filter
SFM	Structure from Motion

CHAPTER 1

INTRODUCTION

Simultaneous Localization and Mapping (SLAM) based on computer vision has got much attention over past few years, and this technology is now rapidly transitioning into a range of real-time products like smart phones, autonomous vehicles, and wearable devices. These real-time and real world applications required to be quick in reacting to the dynamics in the scene and should be able to capture as much as possible even in extreme lighting conditions. It is very important to take power efficiency into account as every application these days are running on battery.

However, the standard vision cameras on which they heavily rely run into problems when trying to supply these, either of huge bandwidth requirements and power consumption at high frame-rates, or diminishing image quality with blur, noise or saturation. Although several SLAM methods have been shown to be working efficiently in real-time up to 30-50 fps. Still standard cameras suffer form motion blur and low dynamic range limitations as shown in Figure 2.2 and also they carry redundant data mostly if the scene is static. These limitations on conventional imaging sensors led neuromorphic vision research community to develop new vision sensors which can understand the scene better than CMOS/CCD conventional sensors. A number of different sensing modalities have been proposed, such as spatial difference or contrast sensors which reduce spatial redundancy based on intensity differences or ratios over space, and temporal difference or contrast sensors which reduce temporal redundancy based on absolute or relative intensity changes over time. One such temporal contrast based camera is called an event camera which can mimick some of the superior properties of the human vision and it comes with great merits for real-time vision, with it's high measurement rate, low latency, high dynamic range and low data rate properties.

This thesis is organized as follows: In chapter 2, we talk about limitations on conventional imaging sensor and then introduce an event sensor, later it's advantages, limitations and calibration of event camera. Related work and proposed method are discussed in chapter 3. Finally, results are shown in chapter 4. The conclusion and future works are discussed in chapters 5.

CHAPTER 2

Event Sensor

2.1 Event camera

Events sensor is biologically inspired real time vision sensors with great advantages in storing and processing the data over traditional imaging sensors. The basic idea of an asynchronous vision sensor is that the output is in the form of address-events (x, y-coordinates) that are generated locally by the pixels. Unlike traditional camera, an events camera records not image frames but an asynchronous sequence of per-pixel intensity changes each with a precise timestamp. It also captures polarity as +1 or -1 representing intensity change whether it's positive or negative respectively.

So, each event is represented as

$$e_i = \{x, y, p, t\} \tag{2.1}$$

where $x, y =$ pixel's coordinates
$p =$ polarity (+1 or -1)
$t =$ timestamp in the order of μs

This representation is sometimes also referred to as address-event representation (AER). An event sensor efficiently encodes image dynamics with extremely temporal contrast and high dynamic range. Figure 2.1 shows that there are no events when the scene is static and stream of events when a black dot is moving circularly.

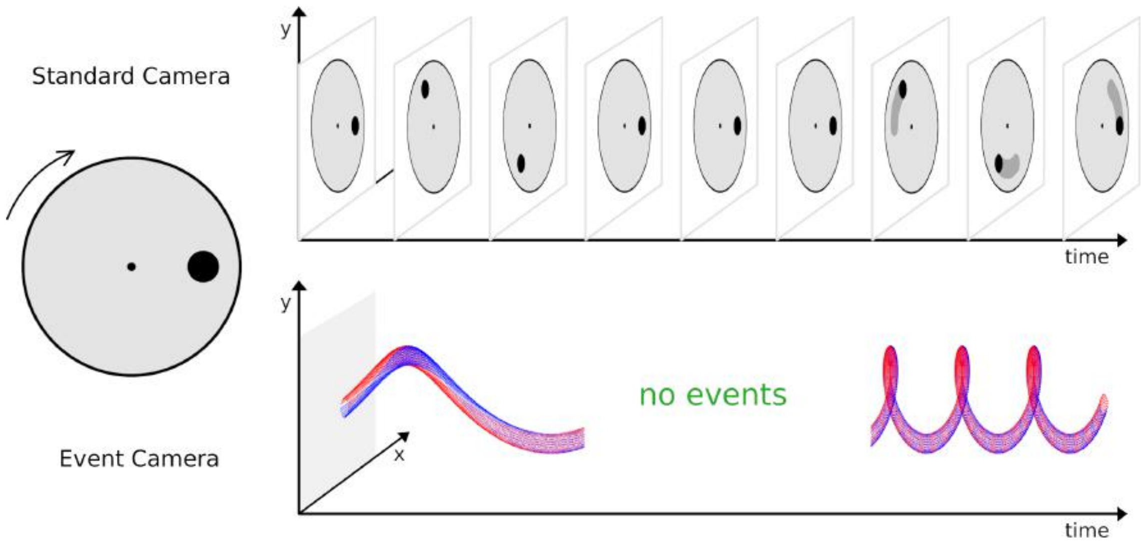

Figure 2.1: Standard camera vs event camera. Figure courtesy Kim et al. [6]

2.2 Standard camera vs Event camera

Standard cameras record scenes at fixed time intervals (i.e. global or rolling shutter) and output a sequence of image frames. For example, as shown in Figure 2.1, if a fixed standard camera is set up to capture the spinning disc with a black dot shown on the left, we get a sequence of images as illustrated in the upper spatial-temporal graph on the right. Even when there are no changes happening in the scene, the sensor keep sending the redundant data. In the same way, standard camera also suffers from motion blur and low dynamic range as shown in Figure 2.2. These limitations on the standard camera were tackled by event camera as discussed below.

On the other hand, Event camera fires asynchronous events (also called spikes), each with pixel location, microsecond precise timestamp, and polarity, indicating log intensity changes of a preset threshold size. By encoding only relative intensity changes, resources required for transmitting, storing and processing a stream of events is much lower than a standard camera. The lower spatial-temporal graph on right side of Figure 2.1 shows the event stream generation where red and blue dots represent positive and negative events respectively. The data rate is less when the disc is spinning slow, and is high when the disc is spinning fast. In the later case, we got tails along the trajectory of black dot on the standard camera frames.

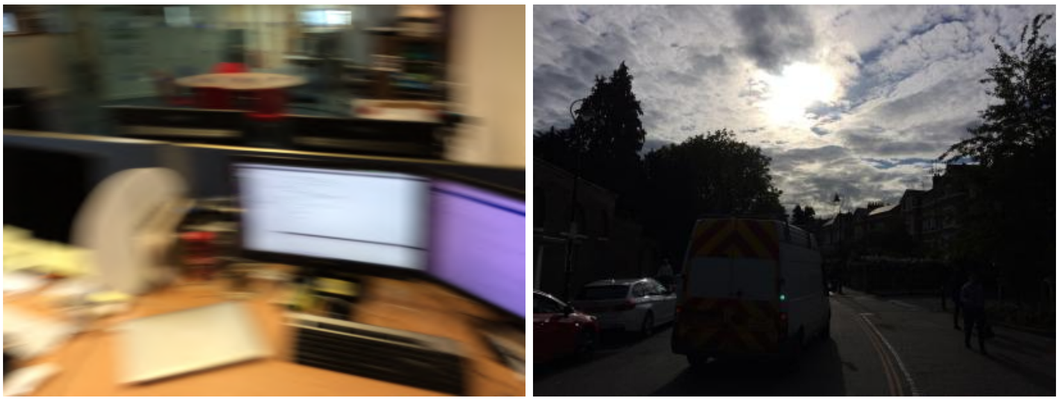

Figure 2.2: Standard camera limitations: Motion blurred image on left and low dynamic range image on right. Figure courtesy Kim et al. [6]

2.3 DAVIS camera

Neuromorphic vision research was trying to create complete neuromorphic systems which can mimic biological counterparts as precisely as possible till 2000's. Later in 2004, Brandli et al.[3], designed Dynamic and active pixel vision sensor (DAVIS) which interleaves conventional intensity frames rather than per-event intensity measurements. The main advantage of the DAVIS pixel design is sharing the same photocurrent between the asynchronous detection of brightness changes and the synchronous readout of intensities, and as a consequence it requires only five additional transistors per pixel to add a global and rolling shutter readout yielding a smaller pixel size. Figure 2.3 shows an image-like visualization of accumulated events and it's corresponding intensity frame. The DAVIS240C camera from iniLabs has following specifications:

Property	**value**
Resolution	180 x 240
Dynamic range	120dB
Temporal latency	15μs
Communication type	USB 2.0

Table 2.1: Specifications of DAVIS camera

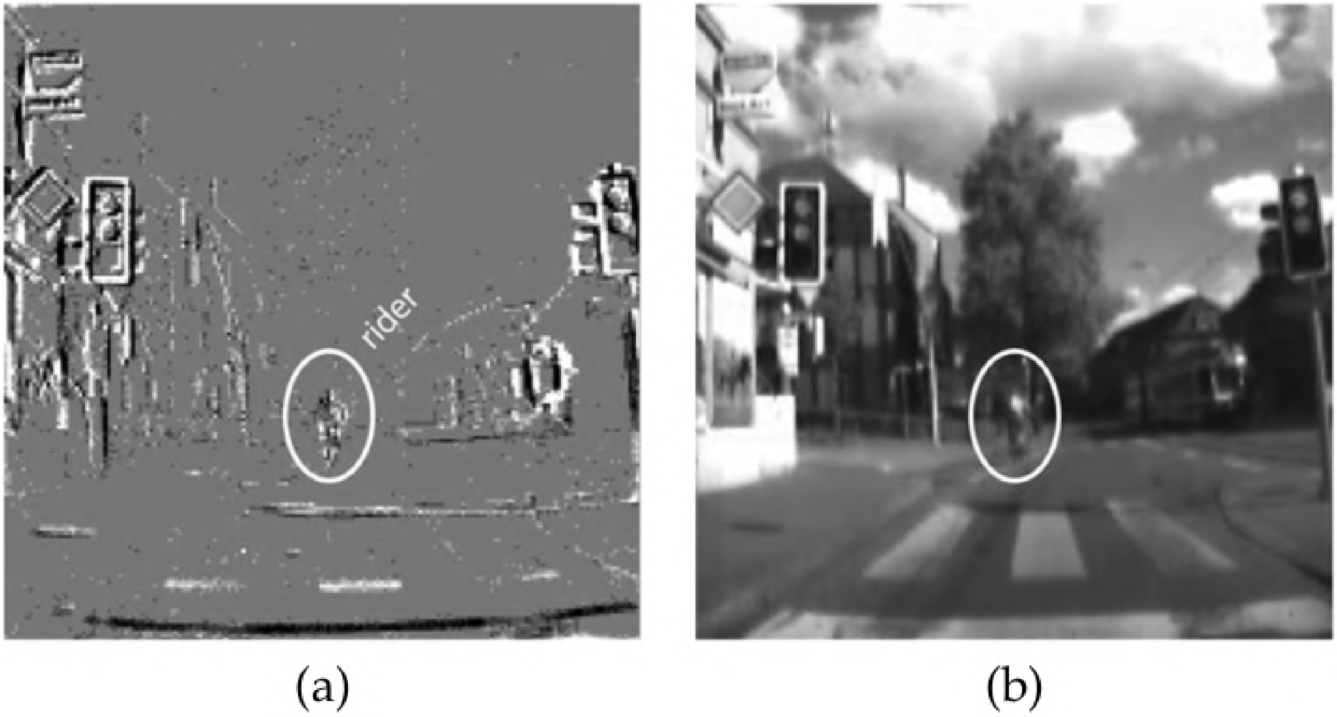

Figure 2.3: DAVIS output: (a) Accumulated events as a frame (b) an intensity frame from DAVIS camera. Figures courtesy of Brandli et al.[3]

2.4 Limitations

Even though event sensor has all these advantages it has it's own share of disadvantages as well. Event sensor has limits in terms of time resolution and bandwidth. Firstly, it has minimum timestamp resolution which is mostly 1μs. The chip bandwidth limits the maximum number of events that can be transmitted per second

Event cameras are also in practice subject to noise and limited in what they can perceive. Noises arise from two primary factors. First, all the electronic components such as photodiodes and transistors contribute some electronic noise. For instance, even in complete darkness, there is still a small electric current across photodiodes which could produce noise events especially noticeable in low-light conditions or in darker areas of scenes. Second, even in well-lit conditions with little electronic noise, all existing event cameras have undesired background events which are not correlated to scene changes. They are all positive events regularly produced at a certain rate which could be for instance once every 10 seconds depending on the positive event threshold.

2.5 Calibration

Camera calibration is the process of estimating parameters of the camera which includes camera matrix and distortion coefficients. This can be done by using images of a special calibration pattern like checkerboard in case of standard cameras which can not be applied directly to event cameras because of it's odd behaviour. For a good estimate of parameters, at least 15-20 images are required. To do this task with event cameras, there are two different ways to proceed. First one is to use a special flickering led pattern as used by Kim et al.[6] in his work. The second method is to reconstruct log intensity images first as discussed in [10] and then do the normal calibration procedure as if they are like intensity images. In this thesis, we have followed the later method which is simple to proceed but make sure that the reconstructed log-intensity images are visually better than event frames. Figure 2.4 shows one such reconstructed image of checkerboard scene and it's corners detected by Caltech toolbox in MATLAB.

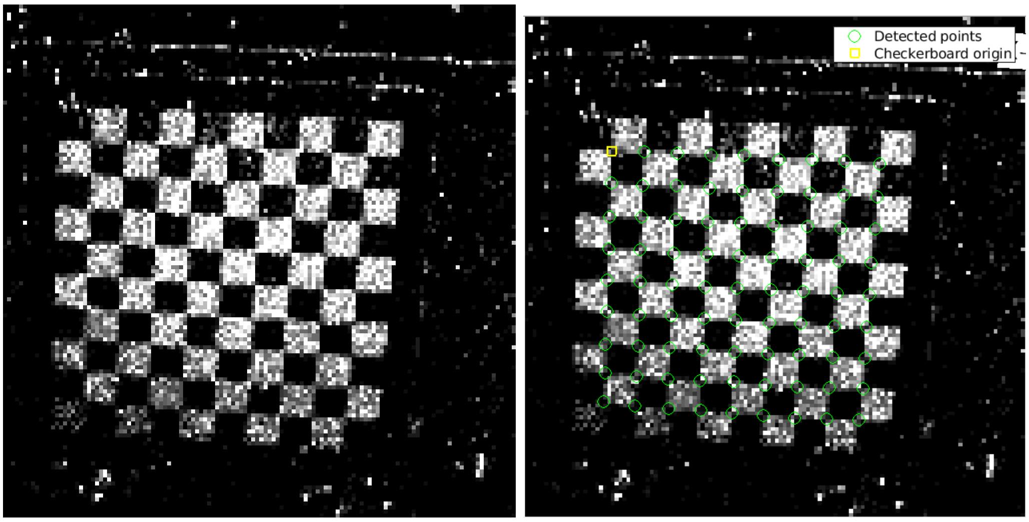

Figure 2.4: Log intensity image on left and the checker board corners detected by the toolbox on right

CHAPTER 3

Event based SLAM using straight lines

3.1 Motivation

Straight lines are so prominent in computer vision applications as they contain more information than point features like SIFT [7], ORB [9], SURF [1], .., etc. This led us to use line features to perform camera tracking and mapping task. Earlier works on SLAM using event cameras, uses probabilistic filters like EKF [5], particle filter [4] to update pose and inverse depth estimations which makes them real-time but they use only one observation at a time which let the systems to use little information about scene and the used events could also be noisy giving rise to false updates. Based on this observation, we propose an algorithm to update camera velocity and inverse depth of line's end points which takes a bunch of measurements each time.

3.2 Related work

As discussed in previous chapter, event camera comes with great advantages to embed in real-time computer vision applications, with high potential in robotics, autonomous vehicles and wearable devices, and it has proven very challenging to use them in most standard computer vision problems. There is need to come up with new algorithms to use event cameras in vision applications because standard computer vision algorithms can not be applied directly. The high temporal resolution makes the event cameras to track rapid movements of itself or objects in the scene in real-time. Autonomous driving application is perfect example for this.

Reinbacher et al.[8] in 2017 proposed a novel method to perform camera tracking using event camera in panoramic setting with three degree of freedom (only rotations: 3D). They claim that the minimal information needed for simultaneous tracking and mapping is the geometric information (spatial position of events) of the event stream, without using the gray scale intensity. The basic idea is that they make a 2D panoramic map using a probabilistic filter and according to which current camera pose is estimated. Those two blocks keep running simultaneously and this works in real-time.

Kim et al.[6] in 2016 proposed an approach which relies on three interleaved probabilistic filters to perform simultaneous localization and mapping. One tracks the global 6-DoF camera motion, Second one is to estimate the log intensity gradients in the keyframe image. Third one is to estimate the inverse depth of a keyframe. The gradient map estimated by second filter is then be upgraded to intensity map in parallel to tracking and mapping modules.

Strictly speaking, there were no prior works done on straight line based SLAM using event camera. But with standard camera there are many. In 2006, using a standard camera, Smith et al.[11] proposed a real-time monocular SLAM with straight lines as features which is implemented with Extended Kalman Filter (EKF). The observation is that straight line features contain more information than point features.

3.3 Line Representation and Detection

In the paper [2], Bay et al. devised a simple but reliable line extractor for an intensity image. In this thesis we are also using the same approach to detect line segments. Given an image, Canny edges are detected first and line segments are extracted as fallows: At an edge pixel the extractor connects a straight line with a neighboring one, and continues fitting lines and extending to the next edge pixel until it satisfies co-linearity with the current line segment. If the extension meets a high curvature, the extractor returns the current segment only if it is longer than 20 pixels, and repeats the same steps until all the edge pixels are consumed. Then with the segments, the system incrementally merges two segments with length weight if they are overlapped or closely located and the difference of orientations is sufficiently small. Figure 3.1 shows an event frame on left and lines detected in blue color on right image. In this thesis line segments are represented with their end points.

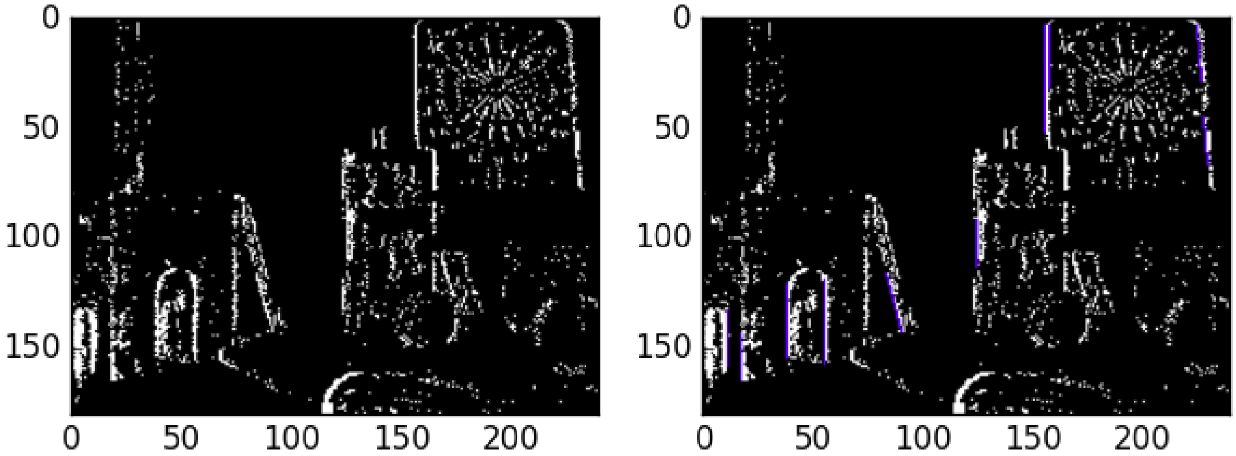

Figure 3.1: Left image represents an event frame and its lines detected in blue color on right image

3.4 Proposed method

As discussed earlier SFM approaches takes several images/measurements to update the pose and inverse depth parameters. Based on assumption that the camera is moving with uniform velocity, we just need to estimate one velocity parameter. For simplicity, assume that the camera has linear motion only. Coming to the inverse depth estimation, if we were to find inverse depth of every point on a straight line, we just need to find the same for any two points on the line and then interpolate to all other points on the straight line. So the task is to find the inverse depth d^l_{1or2} of end points of all detected line segments in an event frame and the camera velocity v_x. Given an event stream, our algorithm works as below:

- Accumulate every 2500 events into a single frame and we call it an event frame.
- Take a batch containing M number of such event frames (say F_j) and take any one of them as a reference frame for tracking module, in our case it's middle frame.
- Detect line segments in the reference event frame. The same straight line may not be detected in other frames because of the asynchronous nature of event camera. So, line correspondence matching is a difficult task.
- The detected line segments (say L_i) are then warped to other frames in the batch according to some random initialization on the parameters but make sure that they lie around the original line segment. Represent the warped line segments as L_i^j.
- In each F_j, and for each $L_i^j(L_i, v_x)$ take a patch along the line in normal direction such that N pixels cover on both sides of the line.
- Take normal distance of all events lie inside the patch to the line and add it to the loss function.

- Minimizing this loss function will try to keep the warped line and dense event strip, which we believe a line, closer.

The loss function can be mathematically represented as,

$$L(v_x, d^l_{1or2}) = \sum_i \sum_j W(e(u,v), L_i^j) d_\perp(e(u,v), L_i^j) \tag{3.1}$$

Where,

$$\begin{aligned}
L_i^j &= \text{warped line } L_i \text{ in frame } j. \\
e(u,v) &= \text{event pixel at } (u,v) \\
d_\perp(e(u,v), L_i^j) &= \text{perpendicular distance from event } e(u,v) \text{ to warped line } L_i^j \\
W(e(u,v), L_i^j) &= \begin{cases} 1, & \text{If event } e(u,v) \text{ lie inside the patch along the line } L_i^j. \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}$$

CHAPTER 4

Experiments and results

4.1 Dataset

Since established benchmarks in the computer vision field have greatly contributed to the advance of algorithms in many areas, an important piece of future work is to design and release suitable comparative benchmarks for event camera-based SLAM research. Recently, Mueggler et al.[6] collected a set of standard datasets using DAVIS camera for most of the event based vision applications. ’slider depth’ is one of them which is used in our work to test the proposed algorithm.

4.2 Results

Figure 4.1 shows the loss function minimization over 50 epochs, Figures 4.2 through 4.4 shows an event frame with depth estimates for line segments on right and their corresponding intensity frames on left. Far objects are represented with red color and blue for nearer ones. As we can see from the intensity images on left, The chair object is in background and several other objects are in foreground. The proposed algorithm has successfully encoded the object’s depths with colors.

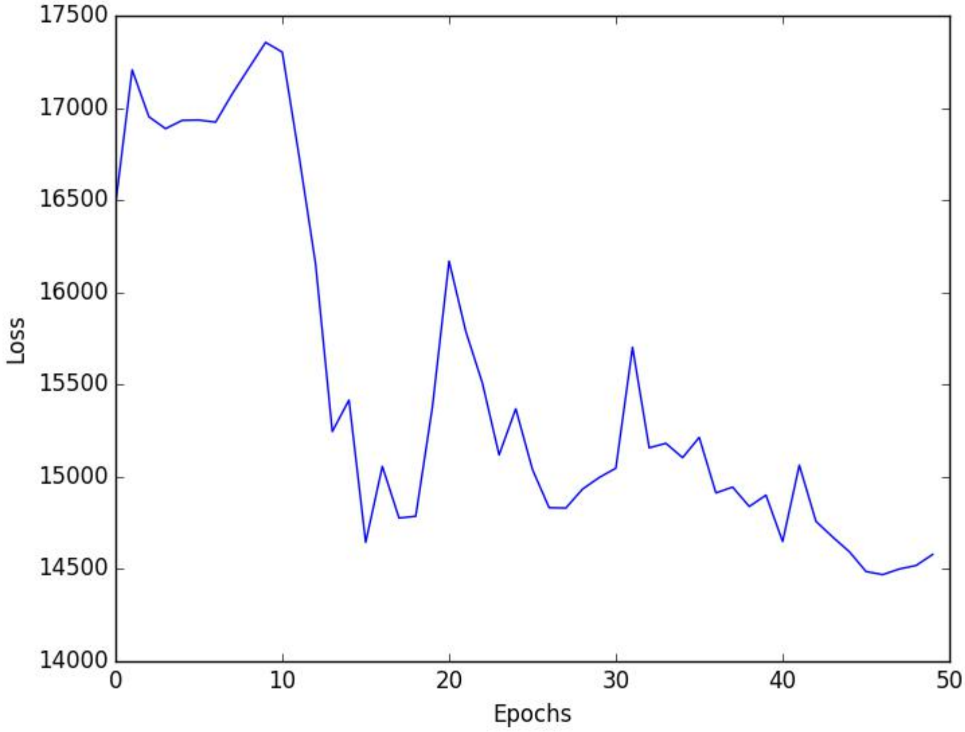

Figure 4.1: Loss minimization over 50 epochs

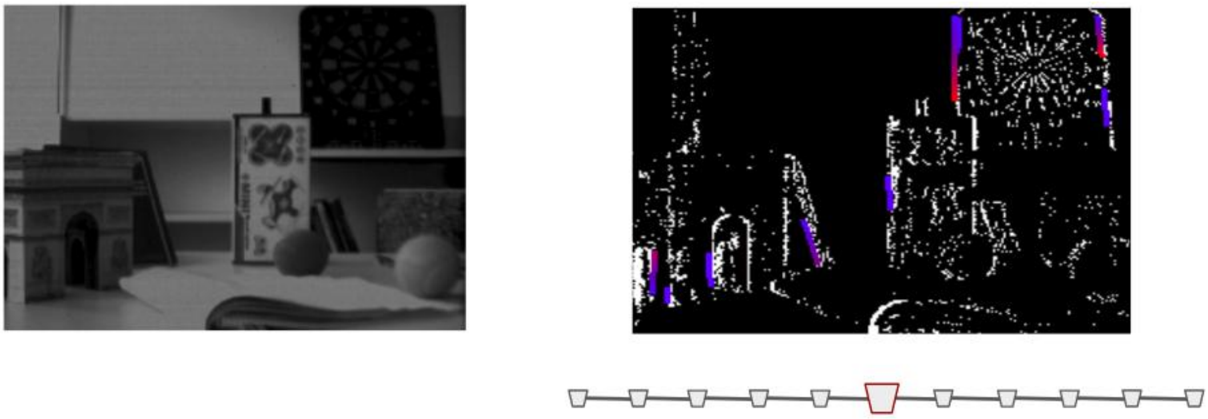

Figure 4.2: Frame 1: event frame with depth estimates for line segments on right and their corresponding intensity frames on left

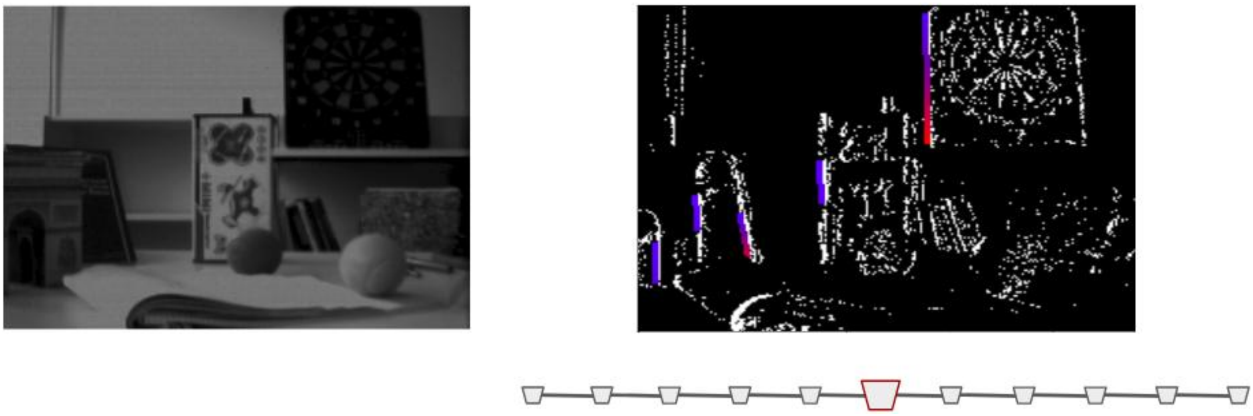

Figure 4.3: Frame 2: event frame with depth estimates for line segments on right and their corresponding intensity frames on left

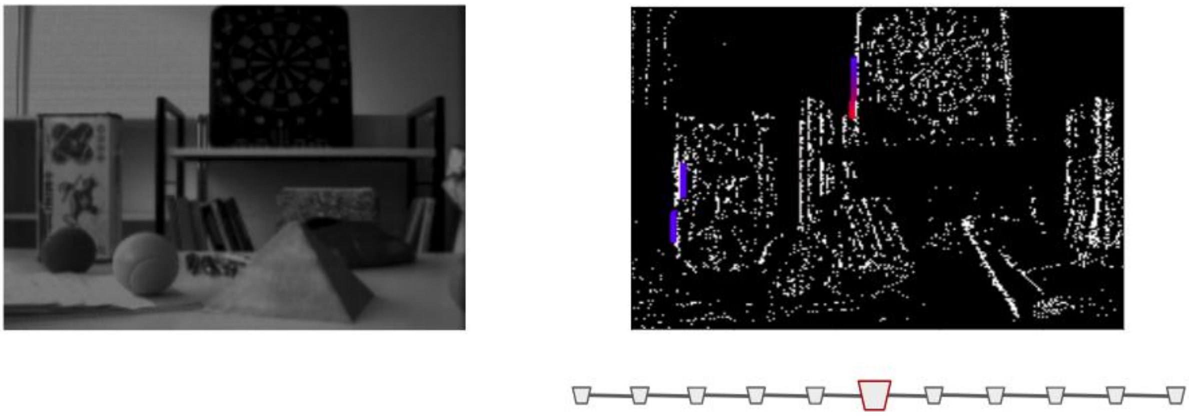

Figure 4.4: Frame 3: event frame with depth estimates for line segments on right and their corresponding intensity frames on left

CHAPTER 5

Conclusion and Future work

We proposed an SFM based algorithm to estimate camera motion and inverse depths using straight lines as features in the event frames and it works in most of the cases except when there is a similar line in the considered patch. As the loss function contains only geometric distance, it will try to converge warped line segment with the nearest one which would give false parameter updates. From the results above, it can be observed that the depth estimates are not accurate enough but can be used as an initialization and refine the estimates in EKF framework. To improve the same, we can also include a global loss on top of proposed local patch loss.

Loss function proposed in this work takes only the local information along the line segments which may not let the system to converge to the global minima always. To improve on the accuracy of inverse depth estimates, some sort of global loss can be included to make it robust. And the linear motion assumption on camera is not practical, which can be extended to 6-DoF as well. We can also take these estimates as initialization and employ an EKF to refine the estimates.

REFERENCES

[1] **Bay, H.**, **A. Ess**, **T. Tuytelaars**, and **L. Van Gool** (2008). Speeded-up robust features (surf). *Computer vision and image understanding*, **110**(3), 346–359.

[2] **Bay, H.**, **V. Ferraris**, and **L. Van Gool**, Wide-baseline stereo matching with line segments. *In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1. IEEE, 2005.

[3] **Brandli, C.**, **R. Berner**, **M. Yang**, **S.-C. Liu**, and **T. Delbruck** (2014). A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, **49**(10), 2333–2341.

[4] **Del Moral, P.** (1996). Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, **2**(4), 555–581.

[5] **Kalman, R. E.** (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, **82**(1), 35–45.

[6] **Kim, H.**, **S. Leutenegger**, and **A. J. Davison**, Real-time 3d reconstruction and 6-dof tracking with an event camera. *In European Conference on Computer Vision*. Springer, 2016.

[7] **Lowe, D. G.** (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, **60**(2), 91–110.

[8] **Reinbacher, C.**, **G. Munda**, and **T. Pock**, Real-time panoramic tracking for event cameras. *In 2017 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2017.

[9] **Rublee, E.**, **V. Rabaud**, **K. Konolige**, and **G. R. Bradski**, Orb: An efficient alternative to sift or surf. *In ICCV*, volume 11. Citeseer, 2011.

[10] **Scheerlinck, C.**, **N. Barnes**, and **R. Mahony**, Continuous-time intensity estimation using event cameras. *In Asian Conf. Comput. Vis. (ACCV)*. 2018.

[11] **Smith, P.**, **I. D. Reid**, and **A. J. Davison** (2006). Real-time monocular slam with straight lines.