High speed imaging using event data and low frame rate video

A Project Report

submitted by

DHRUV KUMAR, EE13B136

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

BACHELOR OF TECHNOLOGY AND MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

MAY 2018

THESIS CERTIFICATE

This is to certify that the thesis titled **High speed imaging using event data and low frame rate video**, submitted by **DHRUV KUMAR, EE13B136**, to the Indian Institute of Technology Madras, for the award of the dual degree of **Bachelor of Technology and Master of Technology**, is a bona fide 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, 600036

Place: Chennai

Date: 10th May 2018

ACKNOWLEDGEMENT

I would like to begin by thanking my guide, Prof. Kaushik Mitra for all his help, support and patience throughout the course of my project work. His approach and dedication to research has been incomparable and has always inspired me to keep pushing myself. All the students in the Computational Imaging Lab have been immensely cooperative, friendly and accommodating and have greatly inspired my own work. I would especially like to thank my team mates Prasan and Ketul for helping me out through the project.

ABSTRACT

High speed cameras are very bulky and costly and we want to construct high frame video sequence from low frame rate video. The idea is to obtain the ego motion of the camera and once we have the ego motion i.e. the trajectory of the camera we can use it to estimate the how the scene has moved between consecutive frame. We use event based sensors to calculate the ego motion. Event-based sensors are built with biological inspiration and differ greatly from traditional sensor types. A standard vision sensor uses a pixel array to produce a frame containing the light intensity at every pixel whenever the sensor is sampled. Event-based sensors, on the other hand, are typically substantially sparser in their output, producing output events that occur upon informative changes in the scene, usually with low latency and accurate timing, and are data-driven rather than sampled. The outputs produced by these novel sensor types differ radically from traditional sensors. Unfortunately, these differences make it hard to apply standard data analysis techniques to event-based data, despite the advanced state of computational techniques for image understanding and acoustic processing. Machine learning especially has made great strides in recent years towards scene understanding, and particularly in the area of deep learning.

We use two approaches. First is to reconstruct the in between frames by image to image translation using conditional adversarial generative network. Generative adversarial network have shown a great amount of success in the recent years. The network make use of the first SLR, last SLR and in between event frames(constructed by clubbing the all the events in between a time interval) to reconstruct the intensity image between the consecutive images. There are various drawback of this method so we move to second approach. Second approach is to first construct intensity image directly from the event frame. Once we have intensity image we can use it to get the relative pose between the SLR frames and the reconstructed event frame. There are two network DispNet and PoseNet. Pose net is used to obtain relative pose between the reconstructed intensity frames and DispNet is used to compute the depth. We use deep prior to train PoseNet and DispNet.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATION

LSTM long short term memory

GAN generative adversarial network

cGAN conditional generative adversarial network

CMOS Complementary metal-oxide-semiconductor

DVS Dynamic vision sensor

AER Address-Events Representation

EKF Extended Kalman Filter

CHAPTER 1

Introduction

1.1 Problem statement and method

Our motivation is to construct high frame rate video sequence from low frame rate CMOS camera using event based sensor. There are already high speed camera but they are bulky, costly and consume lots of power. All these cameras are not data driven. They capture lot of redundant information like they keep on capturing video event when there is no scene change thereby wasting power and storage. Instead of capturing high frame rate video we can capture low frame rate video and use event based sensors to estimate the in between frames.

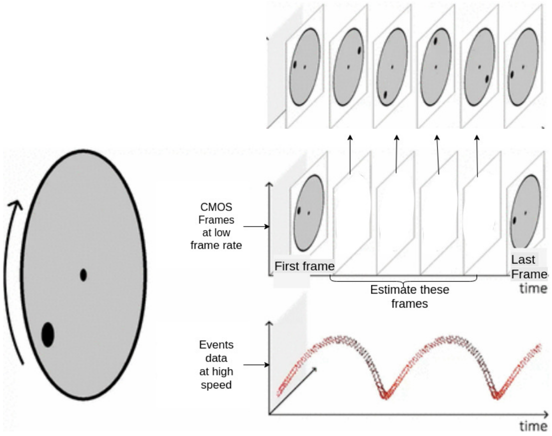


Figure 1.1: Reconstruction of in between frames using event based data

There are two ways of reconstructing in between CMOS frame. First is directly estimate the CMOS frame using deep learning architecture.We try conditional generative adversarial network to reconstruct the in between frames. Second is to learn the geometry of the scene. Using the images of low frame rate CMOS camera we estimate the depth of the scene and with the events data we first construct event frame by combining the events between in between time interval. The event frames are then used to reconstruct sudo intensity frames. Sudo intensity frames and the obtained depth map are used to compute the relative pose between the sudo intensity frame. Now we have the depth as well as pose for the in between event frames. We warp the first and the last image of the low frame rate camera to warp the in between frame.

We reconstruct the photo-realistic image using event based sensors keeping the following day to day problem in consideration First, power constraints are an ever-present constraint on real world implementations. We cannot consume arbitrary amounts of energy to perform an intended task.Second, real-world timing and latency constraints are perennial.

CHAPTER 2

Prior Work

There have been lot of recent work on Neuromorphic or event-based cameras. The low latency compared to traditional cameras make them particularly interesting for tracking rapid camera movement. Also more classical low-level computer vision problems are transferred to this new domain like optical flow estimation, or image reconstruction. We will survey on problems that benefits the most from the temporal resolution of event cameras: camera pose tracking as proposed in this work. Typical simultaneous localization and mapping (SLAM) methods need to perform image feature matching to build a map of the environment and localize the camera within [1]. Having no image to extract features from means, that the vast majority of visual SLAM algorithms can not be readily applied to event-based data. Milford et al. [2] show that it is possible to extract features from images that have been created by accumulating events over time slices of 1000 ms to perform large-scale mapping and localization with loop-closure.

A different line of research tries to formulate camera pose updates on an event basis. Cook et al. [3] propose a biologically inspired network that simultaneously estimates camera rotation, image gradients and intensity information. An indoor application of a robot navigating in 2D using an event camera that observes the ceiling has been proposed by Weikersdorfer et al. [4]. They simultaneously estimate a 2D map of events and track the 2D position and orientation of the robot. Similarly, Kim et al. [5] propose a method to simultaneously estimate the camera rotation around a fixed point and a high-quality intensity image only from the event stream. A particle filter is used to integrate the events and allow a reconstruction of the image gradients, which can then be used to reconstruct an intensity image by Poisson editing. All methods are limited to 3 DOF of camera movement. Guillermo Gallego et al. [9] propose an implicit Extended Kalman Filter (EKF) approach to localize the DVS with respect to a given dense map of the 3-D scene (consisting of geometric and photometric information) without additional sensing, just using the information contained in the eventstream. The map is not constrained to consist only of lines and it is also richer in brightness changes.They allow for localization in the general case of 6-DOF motion of the DVS and design the filter accordingly.

Benosman et al. [6] tackle the problem of estimating optical flow from an event stream. This work inspired our use of an event manifold to formulate the intensity image reconstruction problem. They recover a motion field by clustering events that are spatially and temporally close. The motion field is found by locally fitting planes into the event manifold. In experiments they show that flow estimation works especially well for low-textured scenes with sharp edges, but still has problems for more natural looking scenes. Barua et al. [7] use a dictionary learning approach to map the sparse, accumulated event information to infer image gradients. Those are then used in a Poisson reconstruction to recover the log-intensities. Bardow et al. [8] proposed a method to simultaneously recover an intensity image and dense optical flow from the event stream of a neuromorphic camera. The method does not require to estimate the camera movement and scene characteristics to reconstruct intensity images. In a variational energy minimisation framework, they concurrently recover optical flow and image intensities within a time window. They show that optical flow is necessary to recover sharp image edges especially for fast movements in the image. In contrast, in this work we show that intensities can also be recovered without explicitly estimating the optical flow.

Deep Learning have shown significant advancement in recent years. First we try to generate the images directly using conditional adversarial network.It has the disadvantage that it fails capture spatial information on those location where there are edges in scene but the event based sensor has failed to capture due to similar intensity nearby that edge. We obtain 6 DOF movement of the camera instead of calculating optical flow. We also obtain depth of the scene. Using these information to warp in between images. We forward warp from both side and then blend them to one. Since we are obtaining global information our way is robust to small error or noise in event data.

CHAPTER 3

Background

3.1 Event-Based Sensors

The sensor operates on logarithmic intensity, it has much greater dynamic range than a standard image sensor. At top, the log intensity of the scene changes in continuous time. When the log intensity increases by a threshold ON threshold) it produces an ON event (bottom plot) and is then reset; when the log intensity decreases by a threshold (OFF threshold) an OFF event is produced. The data rate therefore is dependent on the rate of change in the scene and not the duration of the recording. These sensors with both dynamic and active sensors allow easier comparison between traditional (frame-based) inputs and event based inputs.

A traditional image sensor would produce a frame-based snapshot that is discontinuous and marred by a blurring of this rapid input. The event-based input, however, maintains a smooth surface over time and does not suffer from blurring. Moreover, if the dot stops spinning, the frame-based sensor will continue to capture and transmit a full frame of unchanging data, while an event-based sensor will produce no further events until a change again occurs.

Standard CMOS cameras send full frames at fixed frame rates. On the other hand, retinal cameras such as a DVS have independent pixels that generate spike events at local relative brightness changes in continuous time. These events are timestamped and transmitted asynchronously at the time they occur using a sophisticated digital circuitry. Each event is a tuple $< x, y, t, p_t >$, where x, y are the pixel coordinates of the event, t is the timestamp of the event, and $p \in \{1, +1\}$ is the polarity of the event, which is the sign of the brightness change. This representation is sometimes also referred to as Address-Events Representation (AER). The DVS has a resolution of 128 * 128 pixels. Event cameras have numerous advantages over standard cameras: a latency in the order of microseconds, a very high dynamic range (140 dB compared to 60 dB of standard cameras), and very low power consumption (10 mW vs 1.5 W of standard cameras).

Most importantly, since all pixels capture light independently, such sensors do not suffer from motion blur.

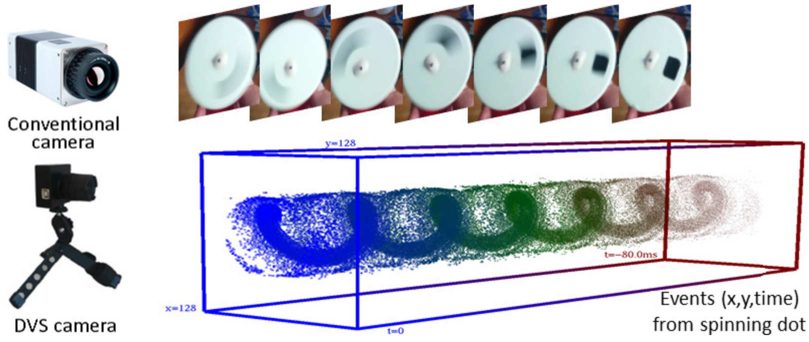


Figure 3.1: Event based sensor

3.2 GANs

GANs have two models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than G. The generator is simply a differentiable function. When z is sampled from some simple prior distribution, $G(z)$ yields a sample y drawn from P_{model} . The training procedure for G is to maximize the probability of D making a mistake. This framework corresponds to a minimax two-player game. The GAN framework pits two adversaries against each other in a game. Each player is represented by a differentiable function controlled by a set of parameters. The game plays out in two scenarios. In one scenario, training examples x are randomly sampled from the training set and used as input for the first player, the discriminator, represented by the function D . The goal of the discriminator is to output the probability that its input is real rather than fake, under the assumption that half of the inputs it is ever shown are real and half are fake. In this first scenario, the goal of the discriminator is for $D(x)$ to be near 1. In the second scenario, inputs z to the generator are randomly sampled from the model's prior over the latent variables. The discriminator then receives input $G(z)$, a fake sample created by the generator. In this scenario, both players participate. The discriminator strives to make $D(G(z))$ approach 0 while the generative strives to make the same quantity approach 1.

GANs learn a loss that tries to classify if the output image is real or fake, while simultaneously training a generative model to minimize this loss. Blurry images will not be tolerated since they look obviously fake. Because GANs learn a loss that adapts to the data, they can be applied to a multitude of tasks that traditionally would require very different kinds of loss functions. We will use GANs in the conditional setting. Just as GANs learn a generative model of data, conditional GANs (cGANs) learn a conditional generative model. This makes cGANs suitable for image-to-image translation tasks, where we condition on an input image and generate a corresponding output image.

CHAPTER 4

Reconstruction using conditional generative adversarial network

Generative Adversarial Networks (GANs) are state of the art architecture for generating images as they use a differential adversarial architecture loss function.

4.1 Proposed method using cGAN

GANs are generative models that learn a mapping from random noise vector z to output image y, G : z -> y. In contrast, conditional GANs learn a mapping from observed image x and random noise vector z, to y, G : x, z -> y. The generator G is trained to produce outputs that cannot be distinguished from real images by an adversarially trained discriminator, D, which is trained to do as well as possible at detecting the generator's fakes.

4.2 Objective function

The objective function of conditional GAN can be expressed as

$$L_{cGAN}(G, D) = E_{x,y}[\log D(x, y)] + E_{x,z}[log(1 - D(x, G(x, z))]$$

Previous approaches have found it beneficial to mix the GAN objective with a more traditional loss. The discriminator's job remain unchanged, but the generator task is not only to fool the discriminator but also to be near the ground truth. We try both L1 and L2 loss function. L1 loss function seems to give less blurring:

$$L_{L1} = E_{x,y,z}[|y - G(x, z)|_1]$$

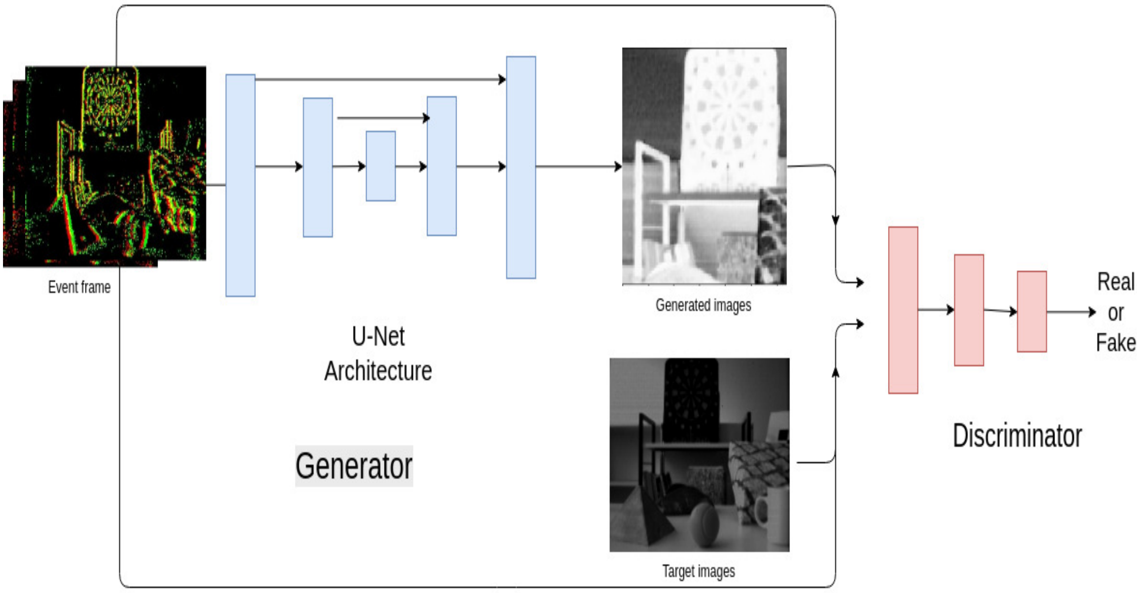


Figure 4.1: Direct intensity image reconstruction using cGAN

So our final objective function is

$$G^{*} = arg\ min_{G}\ max_{D}\ L_{cGAN}(G, D) + \lambda L_{L1(G)}$$

4.3 Network Architecture for cGAN

We tried two conditional adversarial network.First cGAN takes event frames as input and tries to reconstruct intensity frames directly.The problem with this approach is that it does not generalize well for the other data as the events data does not capture the intensity information. The images produced by this network are not temporally consistent as each frame is produced independent of the other frame.But it has great advantage if we train on some data and try to reconstruct high frame rate video on the same data. Since this does not generalize on other data we modify the architecture so that the network can produce good results for on the data which the network has not seen before.

Second we add a LSTM in the latent space of the Generator to remove the temporal inconsistency in the frame produced by the network. We also add another encoder module so that the network learn the texture information well, which it was not able to capture directly from the event data.

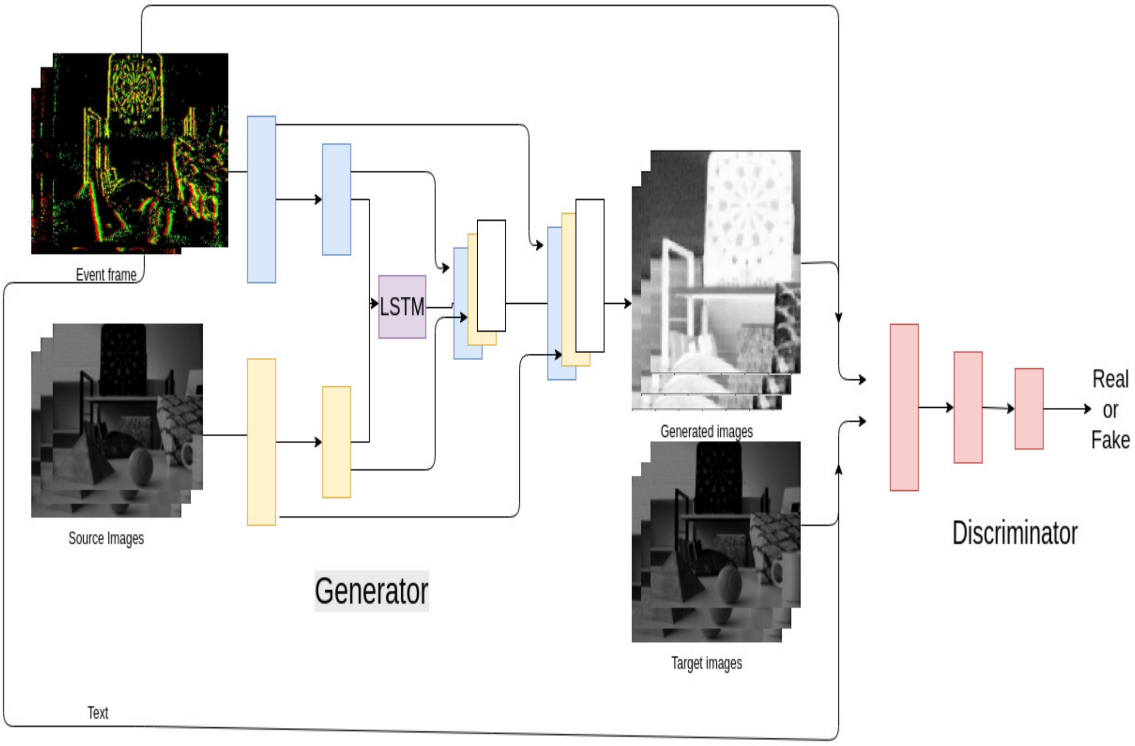


Figure 4.2: Modified cGAN architecture

4.3.1 Generator with skip connection

All the prior image to image translation problems used encoder decoder network. In these kind of network , input is passed through a series of layers that progressively downsample until a bottleneck layer at which point the process is reversed. For such a network all the information must flow through all the layers. The problem which we are considering the structure in input is roughly aligned to the structure of the output. The skip connections helps in retaining the overall structure.
Adding skip connection to the "U-Net" architecture helps us to bypass the bottleneck for the information. Specifically , we add skip connection between each layer i and layer n-i, where n is the total number of layers. Each skip connection simply concatenates all channels at layer i with those at layer n-i.

4.3.2 LSTMs in latent space

LSTMs helps preserve the error that can be backpropagated through time and layers. If we directly produce the images without LSTM, they donot have temporal consistency. So, to get temporal consistency we add LSTM in hidden layers. It has advantage that the video produced are temporally consistent. But it has one disadvantage that it remembers spatial information.

4.4 Results using cGAN

4.4.1 Results using cGAN

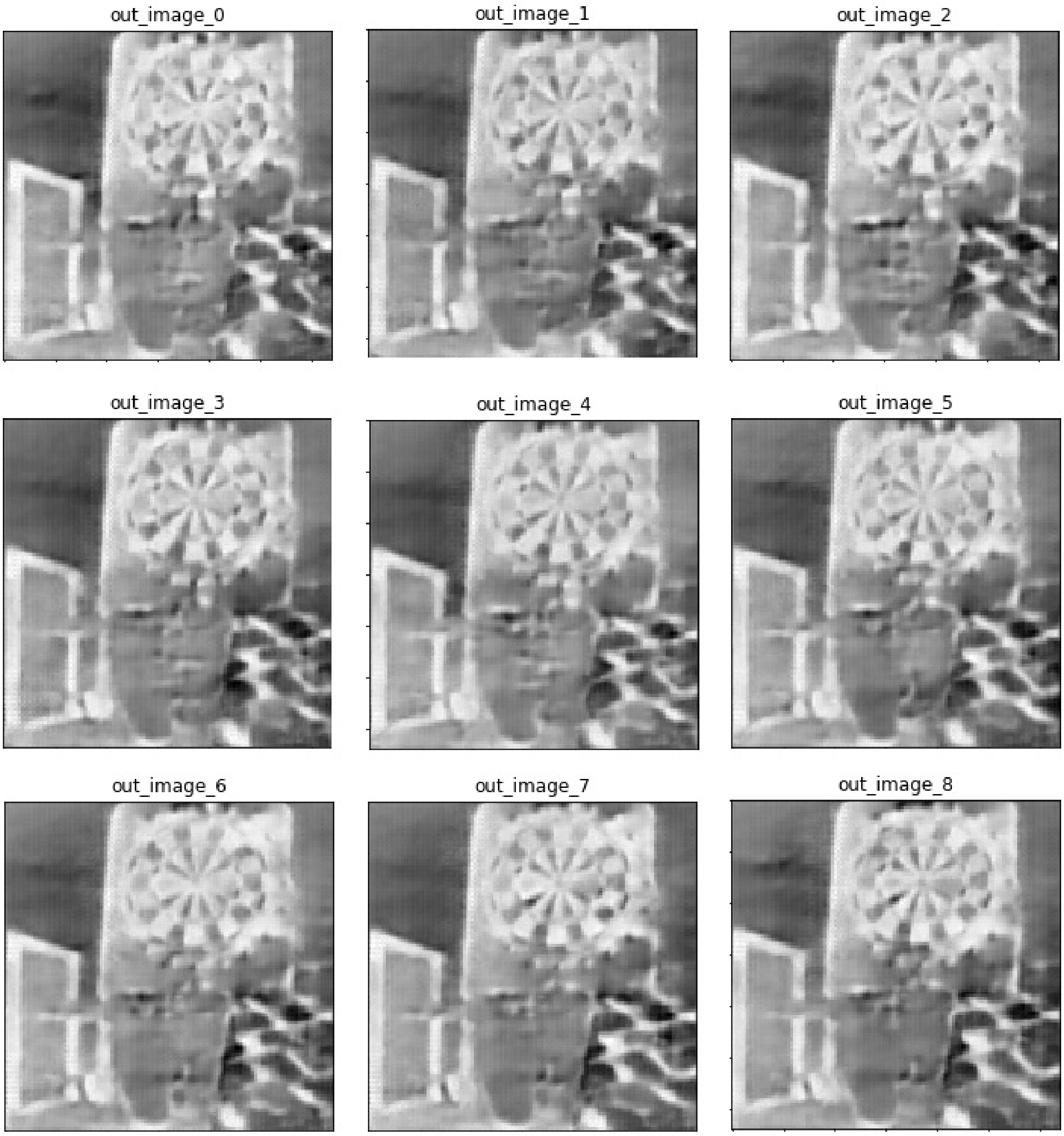


Figure 4.3: Results using cGAN directly

4.4.2 Results using modified cGAN

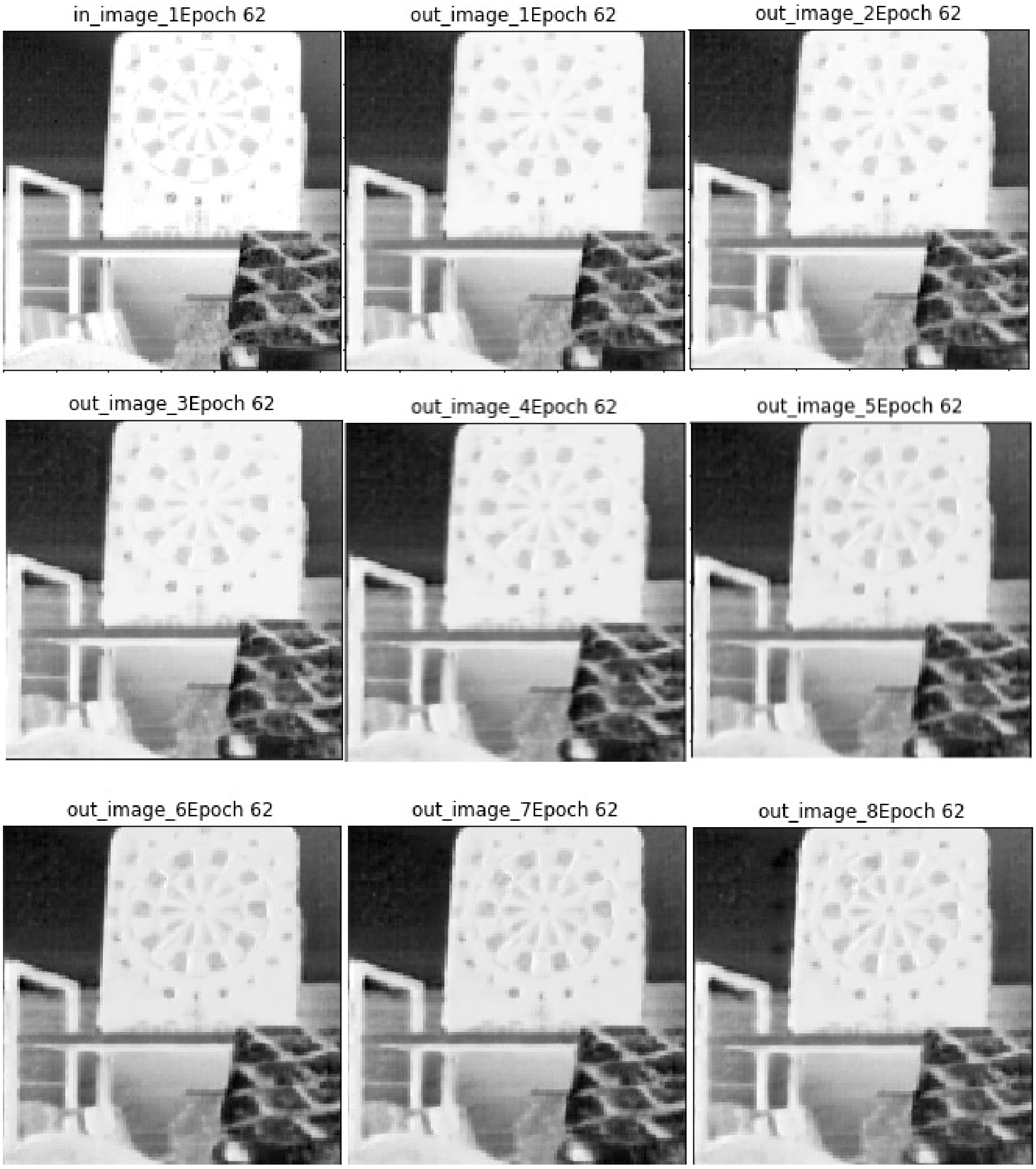


Figure 4.4: Intensity images generated using modified cGAN

CHAPTER 5

Reconstruction using unsupervised depth and ego motion

Since the conditional GANs learn only the local information. It fails for the cases in which there is scene change but the event based camera has failed to record any change may be due to similar texture.We try another geometric approach which is instead of learning the local information if we can capture global information like depth and ego motion of the camera. This will work even for the cases in which there is event based data are spatially not recorded properly.

We aim to use the DVS for ego-motion estimation. The approach provided by traditional visual-odometry frameworks, which estimate the camera pose at discrete times (naturally, the times the images are acquired), is no longer appropriate for event-based vision sensors, mainly due to two issues. First, a single event does not contain enough information to estimate the sensor pose given by the six degrees of freedom (DOF) of a calibrated camera. We cannot simply consider several events to determine the pose using standard computer vision techniques, because the events typically all have different timestamps, and so the resulting pose will not correspond to any particular time. Second, a DVS typically transmits 10^5 events per second, and so it is intractable to estimate the DVS pose at the discrete times of all events due to the rapidly growing size of the state vector needed to represent all such poses.

For obtaining photorealistic reconstruction we propose to use a conventional image sensor along with the event sensor. The conventional image sensor will compensate for the lost spatial information due to encoding of events. In order to reconstruct photorealistic intensity images we warp the low frame-rate intensity images to intermediate locations where only event data are available. Image warping requires the information of scene depth and the ego-motion estimate of the sensor. To estimate scene depth we use the low frame rate intensity images because they contain enough texture information and to estimate the sensor ego-motion we use the temporally dense event sensor data. We propose an unsupervised and learning free method for estimating scene depth by explicitly enforcing geometric and photometric constraints between successive intensity frames. We jointly estimate the dense depth map and relative pose between successive images by warping one image to the location of the other and minimizing the the photometric error between them.

As compared to intensity images, event data is more suitable for estimating the ego-motion because they acquire data at a much higher temporal rate. However, event data is generally noisy and the stochastic model of event generation makes it hard to rely on the actual values of the events. For this reason, we initially map the events to pseudo-intensity frames. We propose an autoencoder based deep learning model to learn this mapping. Events do not hold any spatial intensity information, hence autoencoder can only estimate pseudo-intensity images. We use the trained autoencoder based model to estimate pseudo-intensity images at all intermediate temporal locations where we would like to warp the conventional image frames. We now estimate the relative pose between the estimated pseudo-intensity images and the two nearest intensity frames in time. To estimate this relative pose we propose to use a direct matching based unsupervised and learning free method similar to the one used for depth estimation. At each of the temporal location we blend the two images obtained by forward mapping the two nearest intensity frames.

We use preexisting unsupervised learning framework for the task of monocular depth and camera motion estimation from video sequences. It has a limitation that that this network requires lot of training and we don't have lot of training data. The network is trained on KITTI dataset. We try to fine tune the network for our task using deep image prior. Deep prior can directly optimize the network but it fails in cases where there is lot of movement of the scene.

This network consists of two module. First is the disparity net. It calculates the disparity from the monocular image and inverse of the disparity gives depth. There is a second architecture which is Pose net. It calculates the relative pose between the input images.So we get the depth of one image(target image) from disparity net and pose with respect to target image. We use depth and pose to inverse warp the other images to the target image. Our loss function is photometric loss with respect to the target image. We also use explainability mask loss and depth smoothning loss.

Our idea is to use these network to get depth between the two SLR frames. We generate intensity image from the event frames between the two SLR images. Using these intensity frame and the depth of the SLR frame obtained using disparity net we calculate the relative pose between the reconstructed intensity frame and first SLR image. Similarly we do with respect to second SLR image. Now we have the first and the last frame i.e. two SLR images , their depth and relative pose for the in between intensity images. We forward warp from both side and then use blending to merge the images from both sides.

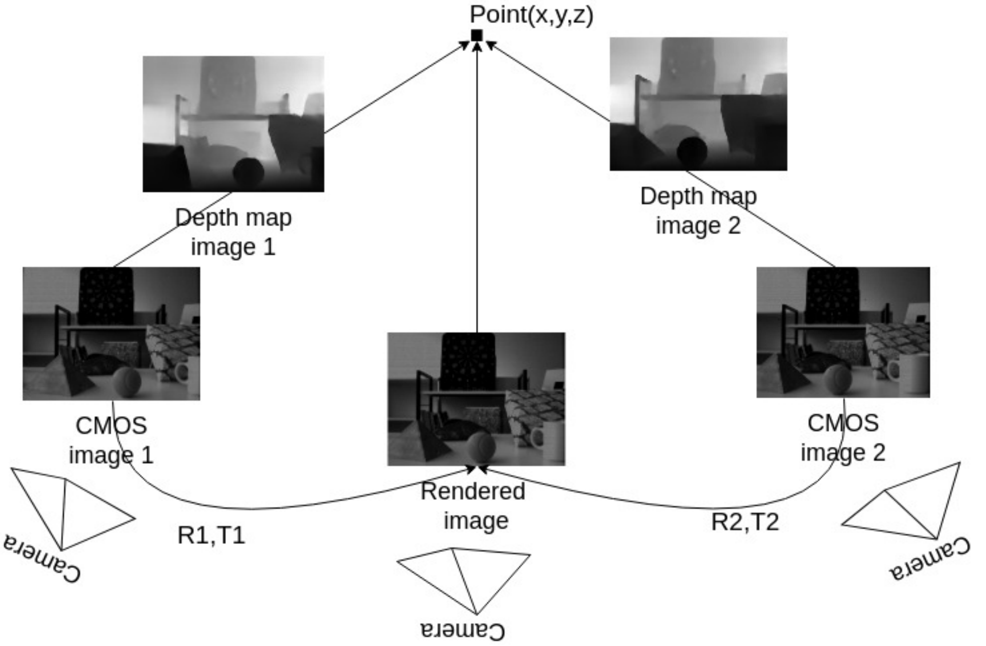


Figure 5.1: Basic framework of problem statement

5.1 Method to estimate depth from two unstructured image

This framework jointly trains a single-view depth CNN and a camera pose estimation CNN from unlabeled video sequences. Despite being jointly trained, the depth model and the pose estimation model can be used independently during test-time inference. Training examples consist of short image sequences of scenes captured by a moving camera. While our training procedure is robust to some degree of scene motion, we assume that the scenes we are interested in are mostly rigid, i.e., the scene appearance change across different frames is dominated by the camera motion.

5.1.1 View synthesis

The idea behind depth and pose prediction CNNs comes from the task of novel view synthesis : given one input view of a scene, synthesize a new image of the scene seen from a different camera pose. We can synthesize a target view given a per-pixel depth in that image, plus the pose and visibility in a nearby view.This synthesis process is implemented in a fully differentiable manner with CNNs as the geometry and pose estimation modules. Visibility, along with non-rigidity and other non-modeled factors, are modled using an "explanability" mask.

5.1.2 Depth image-based rendering

Here we have the we want to warp the reference image to target image. Let p_t denote the homogeneous coordinate of a pixel in the target image, K denotes the intrinsic matrix of the camera. We want to project p_t from target image to the source image. We first move from target image frame to camera frame by multiplying depth and inverse of intrinsic matrix ($\hat{D}_t K^{-1} p_t$). Now we apply translation and rotation in camera domain. Using pose net we get 4*4 transformation matrix$\hat{T}_{t \longrightarrow s}$. We multiply transformation matrix $\hat{T}_{t \longrightarrow s}$ and coordinates in camera frame ($\hat{D}_t K^{-1} p_t$) to get the transformed coordinates in camera frame $\hat{T}_{t \longrightarrow s} \hat{D}_t K^{-1} p_t$. The coordinates in the camera frame are back projected to pixel frame by multiplying with K. So we can obtain the projection of a point in target image p_t to the source image by

$$p_s \sim K \hat{T}_{t \longrightarrow s} \hat{D}_t K^{-1} p_t$$

The projected coordinates p_s are continuous values. We use bilinear interpolation to linearly interpolate the the value of 4-pixel neighbours of p_s to approximate $I_s(p_s)$ i.e. $I_s(p_s) = \sum_{i \epsilon \{t,b\}, j \epsilon \{l,r\}} w^{ij} I_s(p_s^{ij})$, where w^{ij} is linearly proportional to the spatial proximity between p_s and p_s^{ij} and $\sum_{i,j} w^{ij} = 1$. Whereas t, b, l, r represent top, bottom, left and right.

5.2 Photometric loss

Let $< I_1, ..., I_N >$ be the training image sequence with one of the frames I_t being the target image and the rest being source view $I_z(1 \geq s \leq N, s \neq t)$. The Objective function can be formulated as

$$L_{vs} = \sum_s \sum_p |I_t(p) - \hat{I}_s(p)|$$

where p indexes over pixel coordinates, and $\hat{I}_s$ is the source view I_s warped to the target coordinate frame based on a depth image-based rendering module, taking the predicted depth $\hat{D}_t$, the predicted 4*4 camera transformation matrix $\hat{T}_{t \longrightarrow s}$ and the source view I_s as input.

The above view synthesis formulation uses monocular videos assuming that the scene is static without object moving in it, there should be no occlusion or disocclusion between the target and the source images. There can be such cases in our data. To tackle this problem we use a explainability prediction network that outputs a per pixel mask $\hat{E}_s$ for each target source pair. This gives a network's belief in where direct view synthesis will be successfully modeled for each target pixel. Using $\hat{E}_s$ the objective function becomes:

$$L_{vs} = \sum_{<I_1,...,I_N> \epsilon S} \sum_p \hat{E}_s(p)|I_t(p) - \hat{I}_s(p)|$$

Since we do not have direct supervision for $\hat{E}_s$, training with the above loss would result in a trivial solution of the network always predicting $\hat{E}_s$ to be zero, which perfectly minimizes the loss. To resolve this, we add a regularization term $L_{reg}(\hat{E}_s)$ that encourages nonzero predictions by minimizing the cross-entropy loss with constant label 1 at each pixel location.

We also use smoothness loss that allows gradients to be derived from larger spatial regions directly.For smoothness, we minimize the L_1 norm of the second-order gradients for the predicted depth maps. So our final objective becomes:

$$L_{final} = \sum_l L^l_{vs} + \lambda_s L^l_{smooth} + \lambda_e \sum_s L_{reg}(\hat{E}^l_s)$$

5.3 Network Architecture for estimating depth

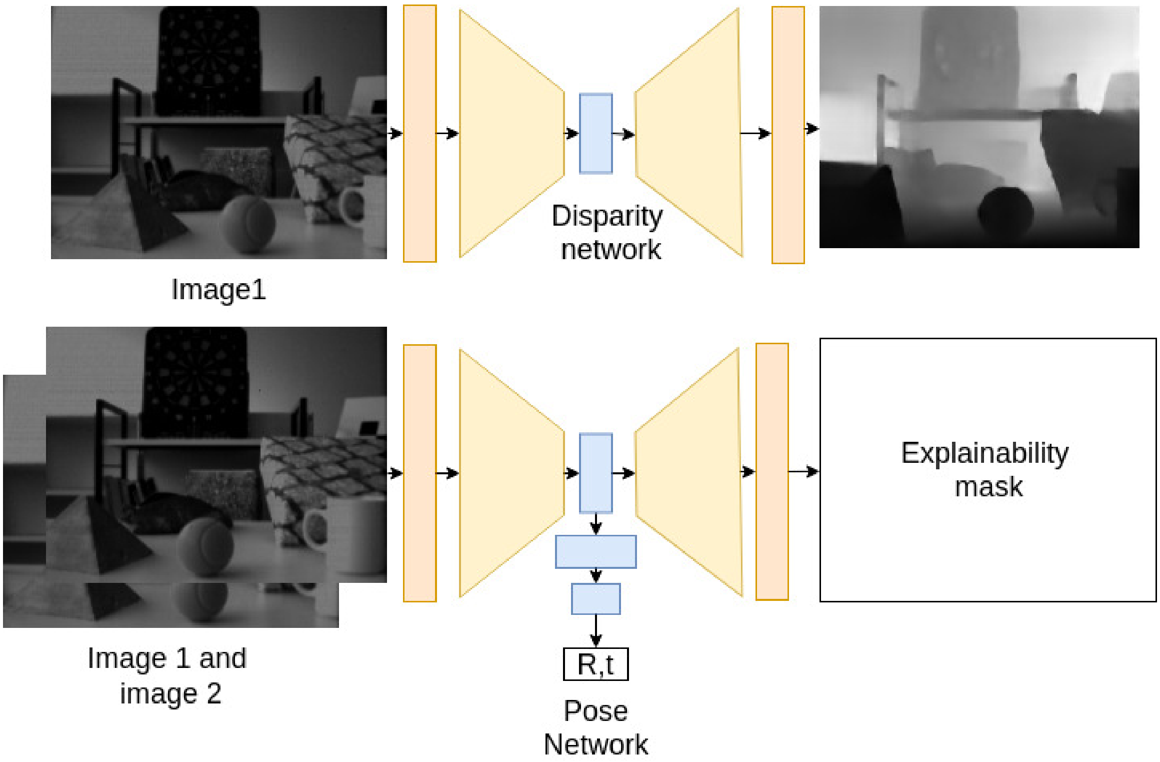


Figure 5.2: Disparity and Pose network

5.3.1 Disparity net

Forr single-view depth prediction, we use the Disparity net architecture that is mainly based on an encoder-decoder design with skip connections. All conv layers are followed by ReLU activation except for the prediction layers, where we use $1/(\alpha * sigmoid(x) + \beta)$ with α = 10 and β= 0.01 to constrain the predicted depth to be always positive within a reasonable range.

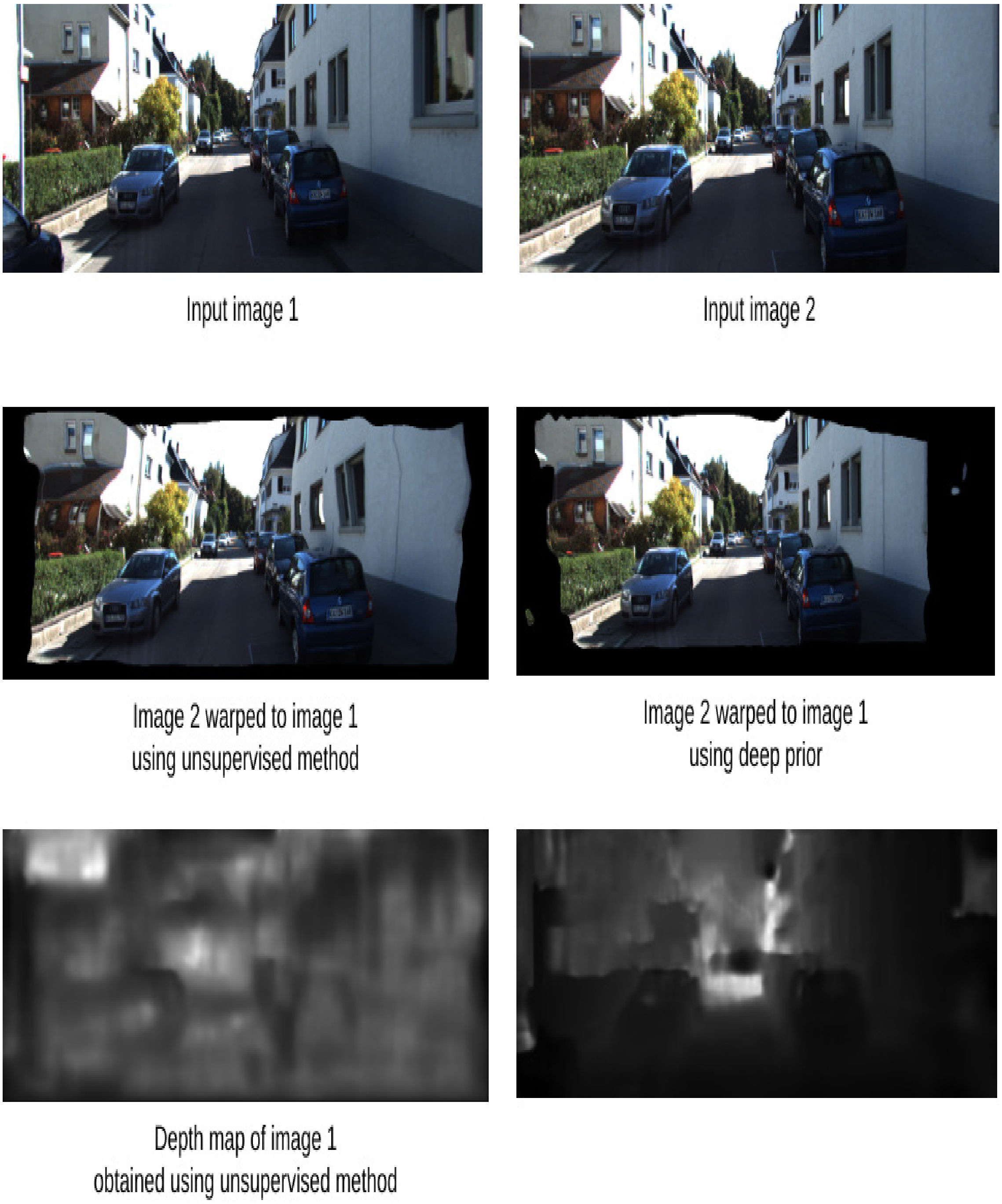
5.3.2 Pose Net

he input to the pose estimation network is the target view concatenated with all the source views (along the color channels), and the outputs are the relative poses between the target view and each of the source views. The network consists of 7 stride-2 convolutions followed by a 1*1 convolution with 6*(N-1) output channels (corresponding to 3 Euler angles and 3-D translation for each source view). Finally, global average pooling is applied to aggregate predictions at all spatial locations. All conv layers are followed by ReLU except for the last layer where no nonlinear activation is applied.

5.3.3 Explainability mask

The explainability prediction network shares the first five feature encoding layers with the pose network, followed by 5 deconvolution layers with multi-scale side predictions. All conv/deconv layers are followed by ReLU except for the prediction layers with no nonlinear activation. The number of output channels for each prediction layer is 2*(N-1) , with every two channels normalized by softmax to obtain the explainability prediction for the corresponding source-target pair.

5.4 Results



Input image 1

Input image 2

Image 2 warped to image 1
using unsupervised method

Image 2 warped to image 1
using deep prior

Depth map of image 1
obtained using unsupervised method

Figure 5.3: Result on kitti dataset

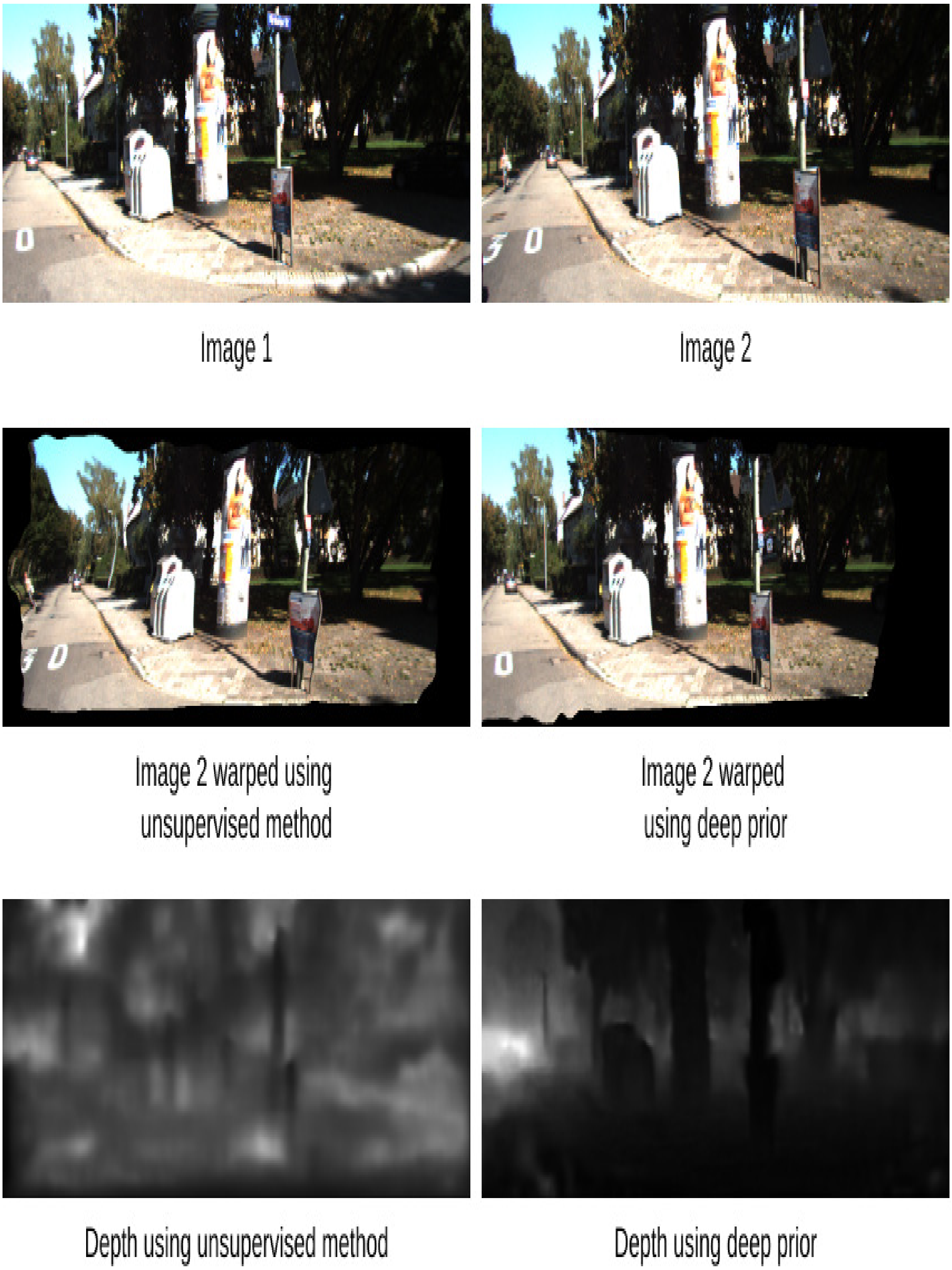


Image 1

Image 2

Image 2 warped using unsupervised method

Image 2 warped using deep prior

Depth using unsupervised method

Depth using deep prior

Figure 5.4: Result on kitti dataset

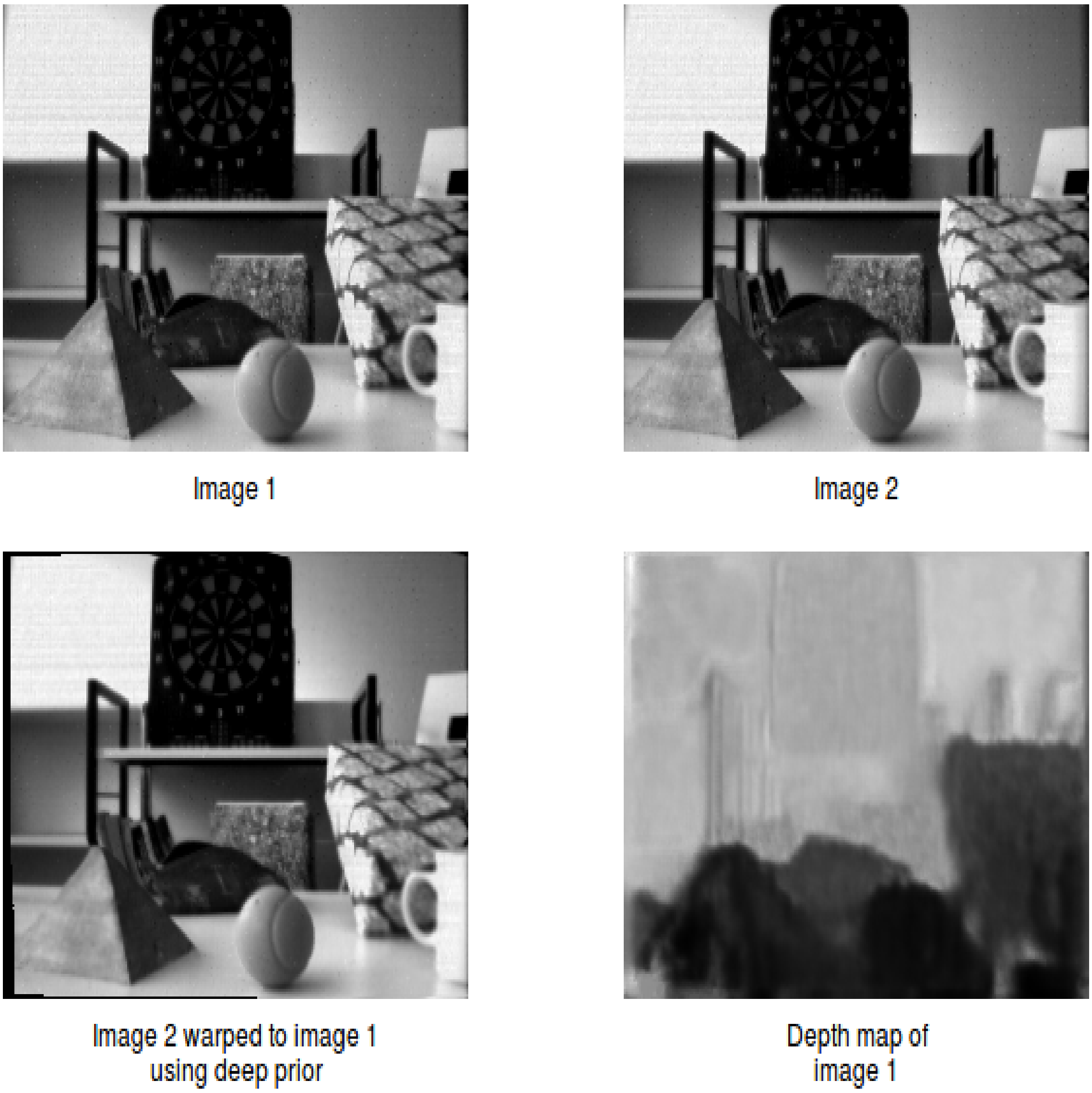


Image 1

Image 2

Image 2 warped to image 1 using deep prior

Depth map of image 1

Figure 5.5: Results on DAVIS dataset

CONCLUSION

We combine the strength of a texture-rich low frame rate conventional camera with a high temporal rate events camera to obtain photorealistic images at high temporal resolution. We achieve this by warping the low frame rate intensity frames from the conventional image sensor to intermediate locations. We compute dense depth maps from the low frame rate images and sensor ego-motion from the events data by direct matching of pseudo-intensity frames reconstructed from event frames. However, in this paper we have assumed a static scene with the sensor in motion. Extending this to dynamic scene will lead to a system for high speed imaging which will be far more power efficient than any of the existing systems.

REFRENCES

1. J. Hartmann, J. H. Klussendorff, and E. Maehle. A comparison of feature descriptors for visual slam. In European Conference on Mobile Robots, 2013.

2. Michael Milford, Hanme Kim, Stefan Leutenegger, and Andrew Davison. Towards visual slam with event-based cameras. In The Problem of Mobile Sensors Workshop in conjunction with RSS, 2015.

3. M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger. Interacting maps for fast visual interpretation. In Neural Networks (IJCNN), The 2011 International Joint Conference on, pages 770-776, July 2011. doi: 10.1109/IJCNN.2011.6033299.

4. David Weikersdorfer, Raoul Hoffmann, and Jorg Conradt. Simultaneous localization and mapping for event-based vision systems. In International Conference on Computer Vision Systems, 2013.

5. Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew Davison. Simultaneous mosaicing and tracking with an event camera. In BMVC, 2014.

6. Li-Tien Cheng, Paul Burchard, Barry Merriman, and Stanley Osher. Motion of curves constrained on surfaces using a level set approach. J. Comput. Phys, 175:2002, 2000.

7. S. Barua, Y. Miyatani, and A. Veeraraghavan. Direct face detection and video reconstruction from event cameras. In 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1-9, March 2016. doi: 10.1109/WACV.2016.7477561.

8. Patrick Bardow, Andrew Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In CVPR, 2016.