GESTURE RECOGNITION USING DYNAMIC VISION SENSORS

A Project Report

submitted by

BIMAL VINOD

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.

June 2017

THESIS CERTIFICATE

This is to certify that the thesis titled **Gesture Recognition using Dynamic Vision Sensors**, submitted by **BIMAL VINOD**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the 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.

Prof. 1
Kaushik Mitra
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 09th June 2017

ACKNOWLEDGEMENTS

It is a great opportunity to express my in depth gratitude to my guide ,professor Dr.Kaushik Mitra for his constant backing and unwavering support in my whole project tenure. I am grateful for his numerous and resourceful ideas and suggestions at each stages of my project.I also thank Renju,Anil,Prasan and all the members of the Computational Imaging Lab, Department of Electrical Engineering, IIT Madras for their valuable guidance throughout the course of the project. I thank profusely to all the 12 friends who helped me generate the datasets that is used for my project. I am extremely grateful to my parents , spouse and all my friends at IIT Madras for being a constant source of support and strength throughout my M-tech tenure.

Finally,I thank God for his blessings due to which I was able to think in the right direction and complete the project successfully.

ABSTRACT

KEYWORDS: Dynamic Vision Sensors(DVS), Optical Flow , stereo , disparity ,Multiclass SVM

Augmented Reality (AR) has revolutionized the way humans interact with computers. While the displays of mobile phones, tablets and PCs of today are limited to an electronic gadget, the AR headsets remove this barrier and project the information directly into the real world. It gives rise to a whole new level of interaction with the digital media and its applications seem to be endless. Undoubtedly, AR is going to be the future and hence its battery life becomes more critical. It is observed that one of the major power consuming components in these systems are the cameras. Current AR systems use conventional vision sensors. In this report, we propose to replace the traditional "power-hungry" cameras in AR systems with the low power Dynamic Vision Sensors (DVS) to boost their battery life. In DVS, independent pixels send out information in the form of "events" whenever the scene brightness changes above a threshold, at the time they occur. Thus, unlike the traditional vision sensors which captures redundant information(static scenes) over many frames, DVS captures only the dynamic content in the scene. This reduces the power consumption and storage/memory requirements drastically. Replacing all the tasks done by the static vision sensors in AR devices with DVS is indeed a big challenging task. Nevertheless we started our mission with a relatively simpler task which is the gesture recognition.This report explains gesture recognition performed on a simulated DVS dataset and on a real DVS dataset collected by us.Finally as expected,we were able to claim that classification accuracies from a simulated stereo DVS is better than a single DVS alone if depth related gestures are involved.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

IITM	Indian Institute of Technology, Madras
RTFM	Read the Fine Manual

NOTATION

r	Radius, m
α	Angle of thesis in degrees
β	Flight path in degrees

CHAPTER 1

Introduction

Augmented Reality(AR) aims to revolutionize how humans interact with computers. While the displays of mobile phones, tablets and PCs of today are limited to an electronic gadget, the AR headsets remove this barrier and project the information directly into the real world. This gives rise to a whole new level of interaction with the digital media and the applications of AR seem to be endless. Some examples of such devices include Microsoft's 'Hololens', the world's first fully untethered holographic computer, Metavision's 'Meta' and Magic Leap's 'Mixed Reality'(MR),which mixes Virtual Reality(VR)and the AR. It is so obvious that , the camera is a fundamental component in all these AR devices. For example, the head mount system in Microsoft's Hololens uses six camera sensors - four environment understanding cameras, one depth camera and one photo / video camera [18]. These cameras allow it to have gesture-based interface through hand gesture recognition, augment virtual objects in real world by capturing the 3D model of the scene and many such applications. All of these cameras are conventional static vision sensors.

1.1 Drawbacks of conventional static vision sensors

There are undeniable advantages to the conventional frame-based imagers - they have small simple pixels, leading to high resolution, large fill factor, and low imager cost. The output format is well understood and is the basis for many years of research in machine vision. On the other hand, frame-based architectures carry hidden costs because they are based on a series of time quantised "snapshots" recorded at a pre-determined frame rate [15]. During the period of transition from frame to frame , a problem similar to under-sampling may arise. This short coming may be tolerable for a human observer, but artificial vision systems that require real time processing such as autonomous robot navigation or high speed control, may fail as a consequence of this shortcoming [15]. Short-latency vision problems require high frame rate and produce massive output data

(e.g., 1 GB/s from 352 288 pixels at 10 kFPS in [14]. Another problem for frame based visual acquisition is the redundancy-the pixels are sampled repetitively even if their values are unchanged. Each frame contains the information from all the pixels regardless of whether that information is changed since the last frame. Obtaining and processing these un-necessary data waste resources and leads to increased bandwidth requirements , high transmission power dissipation and increased memory size. Dynamic range of a camera is typically limited by the identical pixel gain, the finite pixel capacity for integrated photo charge, and the identical integration time. For machine vision in uncontrolled environments with natural lighting, limited dynamic range and bandwidth can compromise performance [15]. Thus it is reasonable to conclude that the typical battery life of Hololens is limited to only 2-3 hours [18] mainly due to the presence of 6 static vision sensors.

1.2 Thesis Motivation

Alternative to conventional approach of capturing the temporal dynamics, people have come up with Dynamic Vision Sensor (DVS) [15] inspired from the functionalities of Y-cells in the human retina. The Y-cells carries visual information related to only the "changes" that is happening around us, including detection of movement ,distance and speed of an object. Similarly in DVS, instead of unnecessarily transmitting the entire image at fixed frame rates, only the change in pixel intensity values caused by movement in a scene are transmitted. Thus the output is just stream of events at a very fine resolution in time(order of micro seconds). The DVS does not deal with absolute pixel intensity values but rather with only the relative changes and hence it requires no Analog to Digital Convertors (ADC) within the chip. This reduces the power dissipation drastically to the order of milli watts. With this local pixel level processing, the data storage and the computational requirements also have come down , and the sensor dynamic range is increased by orders of magnitude slightly greater than 120dB(conventional cameras has dynamic range of approx. 60-70dB). Additionally, the events can be sent out with very low latency and at a temporal resolution as high as 1 micro second (corresponds to 1 Mega Events Per Second). Thus the equivalent frame rate is typically of the order of several kHz due to which the DVS camera will be highly useful in applications such as tracking of fast moving objects , traffic data acquisition ,

moving object in surveillance camera etc.[15][22]

In this thesis work ,the main motivation is the idea of improving the battery life of AR platforms by replacing all its static vision sensors with the DVS. As we know, one of the major purpose of camera in AR systems is the hand gesture recognition. This thesis explains our attempt to perform the hand gesture recognition using DVS camera outputs, with the intension of replacing the normal camera that is used for same purpose in the AR systems. The final objective is to achieve the recognition of gestures taken from a moving DVS. This may require some motion compensation using the inputs from an IMU(Inertial Measurement Unit) that must be integrated with the DVS.

1.3 Application Scenario

The hand gesture recognition using DVS camera finds a potential application in the portable AR systems. Replacing the "always-ON" "power-hungry" cameras in the present AR head mount systems with the DVS will make them more power efficient. Currently ,the cameras are used for many specialized tasks in AR systems like hand gesture/activity recognition, augmenting 3D objects in the real world scenes etc. Even replacing certain cameras like the ones used for hand gesture recognition with the DVS can boost its battery life significantly. Recently , methods have been explored to recover the intensity images from event sets of DVS by Yoshitaka Miyatani et al. [3] and Christian et al. [24]. This encourages us to explore the possibilities of replacing even the cameras that are used for intensity (RGB) images. In addition to applications in AR platforms, DVS based gesture recognition can also be used in various assisting devices. For example, a deaf-mute person can have a DVS based head mount device capable of understanding his hand gestures and communicate that orally through a loud speaker. Again ,since the DVS is capable of recovering the intensity images also, apart from recognising the activities that is happening around ,it can be employed as a low power solution to assist visually challenged people similar to the blind glass [12] proposed by Microsoft.

1.4 Previous work

1.4.1 Gesture recognition using conventional camera:

Frerk Saxen et. al, [25] carried out an image based gesture recognition in a mobile assistance head mount platform. The system helps the users in performing their routine tasks in the context of assembling complex products. An HMM(Hidden Markov Model) is used as a classifier to extract the dynamic hand gestures based on the motion parameters calculated from the optical flow. They handle the camera motion by subtracting the ego-motion from image flow. Recently many works were proposed using CNNs, Molchanov et. al, [20] have used RGB-D images for gesture recognition and show superiority of its performance across different subjects and widely varying lighting conditions. Molchanov et. al, [21] do online detection and recognition of hand gestures using recurrent 3D CNNs with RGBD and IR data, achieving state-of-the-art results. All these methods are based on inputs from conventional intensity based cameras sometimes with other modalities like the depth. In our method we want to replace them with DVS camera given its power saving capabilities. Several works were carried out for the hearing impaired based on the sign language /hand gesture recognition[5][22]. But none of the works utilizes the power efficiency of the Dynamic Vision sensors.

1.4.2 Gesture recognition using DVS camera :

Gesture-Based remote control using stereo pair of dynamic vision sensors is demonstrated by T. Delbruck et. al,[14] using HMM models. They detect the hand trajectory through spatiotemporal correlation of the output events from a stereo pair of DVS by using LIF(Leaky Integrate and Fire) neurons. DVS camera based bare hand gesture recognition was carried out by Eun Yeong Ahn et.al, [1], where they classified three hand gestures performed during a rock-paper-scissors game. They bin the events spatially and extract features like width of the hand for classification. Both the above works are done with a static DVS camera, whereas in our proposed idea the final aim will be the hand gesture recognition with a DVS on a moving platform(head mounted).

1.5 Objectives

The objective of the thesis are

1. To explore the feasibility of using the Dynamic Vision sensors for gesture recognition in AR platforms.
2. To implement an efficient and simple algorithm to detect the gestures from DVS events.
3. To study various feature extraction methods that exists for the normal RGB videos and adapt them to DVS event-based videos to do a performance analysis on the recognition rate.
4. To explore the possibility of improving the recognition rate of depth related gestures by simulating a stereo DVS using a monochromatic stereo camera.

1.6 Contribution of the Thesis

1. Generated a DVS Dataset of 12 hand gestures,10 sets each from 12 subjects (1440 gestures)
2. Analyzed the performances of various shape and flow based algorithms on a simulated DVS dataset as well as on the real DVS dataset mentioned above.
3. Generated a stereo dataset of 12 hand gestures ,10 sets each from 6 subjects (720 gestures).
4. Tried the effects of both the Semi Global Block Matching(SGBM) and the Graph Cut algorithms for calculating the disparity map using the standard stereo RGB image pairs.
5. Adapted the SGBM algorithm to the DVS event frames and the depth obtained is used as an additional feature to improve the recognition rate.

1.7 Thesis Organisation

This thesis report is organized as follows. chapter 2 explains about the DVS camera used for capturing gestures,chapter 3 describes the initial experiments carried out on a simulated DVS dataset , chapter 4 details the creation of DVS and stereo gesture dataset at IITM , chapter 5 explains the classification performed on the real DVS data , chapter 6 describes the simulation of a stereo DVS and the gesture clasification using the depth estimated as a feature.Finally the chapter 7 concludes the project with a hint to the future direction.

CHAPTER 2

Dynamic Vision Sensors(DVS)

2.1 Basic Concept

The figure 2.1hows the various models of dynamic vision sensors developed by INI Labs, switzerland.

Figure 2.1: Dynamic vision Sensors [23]

DVS is a 128 x 128 pixel CMOS vision sensor that uses a patented technology that works like your own retina.Its pixels respond asynchronously to relative changes in intensity rather than its absolute value. Each pixel independently and continuously measures the change in their log intensity values from time to time(at 1Âţs resolution). If the intensity value at a pixel location is increased above a particular threshold(typically 15 % contrast), it generates a positive (ON) event. Similarly negative (OFF) event is generated if it decreases below the threshold. There are cases wherein multiple DVS pixels request to output events at the same time and these events will be asynchronously output with sub-microsecond delays. This flow of asynchronous events is usually in the format of Address Event Representation (AER) [16].Thus the DVS directly encode scene reflectance changes through the asynchronous stream of pixel address-events (AEs) and reduces the data redundancy while preserving precise timing information [2].

2.2 Address-Event Representation(AER)

The output of DVS is in the form of Address-Events (AEs) generated locally by the pixels. AE is an encoded format which is represented as a collection of the quadruples (t, x, y, p). Here (x,y) is the address of the pixel in the 128x128 sensor array, p is the polarity of the event (ON or OFF) and t , the timestamp(in microseconds) .The Version 1.0 of the AEDAT format AEDAT 1.0 has a total of 6 bytes for one event [25] - 16 bits for the address and 32 bits for the timestamps.The address has to be interpreted according to a specific jAER AEChip class definition of that address. All integer data and fields are always signed and big-endian.The file format for AEDAT 1.0 is shown in table 2.1. This type of asynchronous event-based data format is called address-event representation (AER) protocol. This is similar to the protocol that is being used to model the transmission of neural information within our biological systems.

bits	Meaning	Description
15	External event	External event detected on the IN pin (TS-Master mode).
14-8	Y address	Y event address. (0, 0) in lower left corner of screen.
7-1	X address	X event address. (0, 0) in lower left corner of screen.
0	Polarity	Polarity (luminosity change): '1' means increase (ON), '0' means decrease (OFF).

Table 2.1: DVS128 AEDAT 1.0 data format

The AEDAT 2.0 is also later introduced which has 32 bit address bits and 32 bit timesatmps [10].

2.3 DVS Pixel Architecture and operation principle

2.3.1 Pixel Architecture

The objective for this pixel design was to achieve low mismatch, wide dynamic range, and low latency in a reasonable pixel area. All these challenges were solved with a fast logarithmic photoreceptor circuit, a differencing circuit that amplifies changes with high precision, and cheap two-transistor comparators [15]. Figure 2.2 shows how these three components are connected.

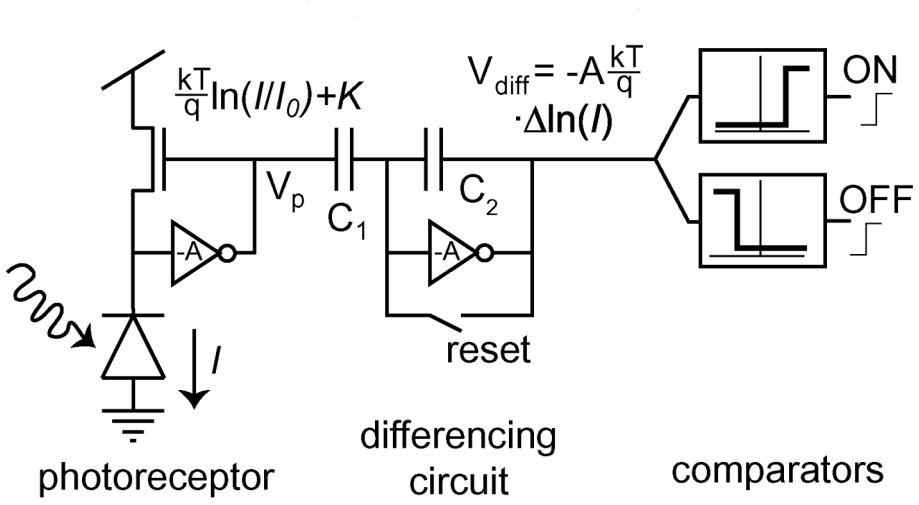

Figure 2.2: Abstracted pixel core schematic of DVS [1]

The photoreceptor circuit has the desirable properties that it automatically controls individual pixel gain (by its logarithmic response) while at the same time responding quickly to changes in illumination. The drawback of this photoreceptor circuit is that transistor threshold variation causes substantial DC mismatch between pixels, necessitating calibration when this output is used directly [17],[12] . The DC mismatch is removed by balancing the output of the differencing circuit to a reset level after the generation of an event. The gain of the change amplification is determined by the well-matched capacitor ratio . The effect of inevitable comparator mismatch is reduced by the precise gain of the differencing circuit.

Because the differencing circuit removes DC and due to the logarithmic conversion in the photoreceptor, the pixel is sensitive to temporal contrast, which is defined as in equation 2.1

$$TCON = \frac{1}{I(t)}\frac{dI(t)}{dt} = \frac{d(\ln((I(t)))}{dt} \tag{2.1}$$

2.3.2 Operating principle

DVS checks the log intensity difference against a threshold to create events . The threshold, T is generally set at 10 % of the contrast in the scene. The basic operation can be explained well using the following conditions.If [log(I(t+1) - log(I(t)] > T , then it generates an ON or positive event and if the [log(I(t+1) - log(I(t)] < -T , it generates

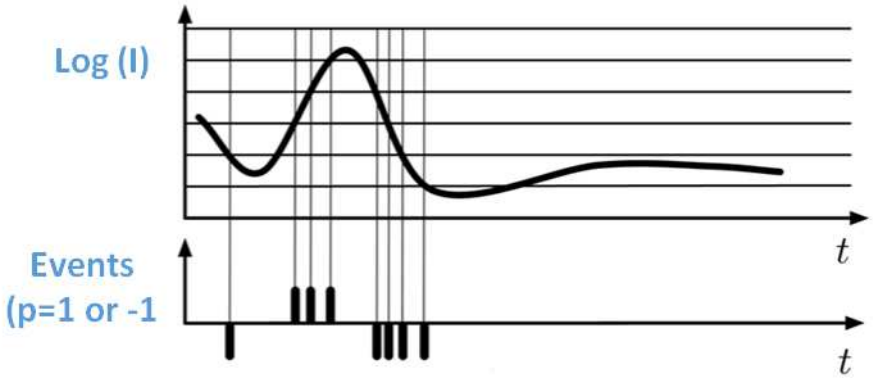

Figure 2.3: Principle of generation of the DVS events ,adapted from [2] Events with +1 or -1 polarity are emitted when the change in log intensity exceeds a predefined threshold.

an OFF/negative event , where I(t) is the pixel intensity value at time t . Figure 2.3 depicts the operating principle of DVS. Once a pixel gets fired and produces an event , it immediately comes back to a reset level which is shown as zero in the above figure.The temporal resolution of DVS is approximately 1μs ,which means each pixel can get fired at a maximum rate of 1 MEPS (Mega Events Per Second)

2.4 Specifications of DVS128

The table 2.2 gives an overview of the specifications of the DVS128.

Parameters	**DVS128 Specifications**
Process technology	CMOS , 0.35μm 2P4M
Array Size	128 x 128 pixels
Pixel Size	40 x 40 μm2
Dynamic Range	120 dB
	2 lux to > 100 klux scene illumination with f/1.2 lens with normal contrast objects. Moonlight (<0.1 lux) with high contrast scene.
Temporal resolution	1μs
Minimum Latency	12μs
Power consumption	Chip: 23mW @ 3.3V
Fixed pattern noise (FPN)	2.1 % contrast mismatch
Max. bandwidth	1 MEPS (Mega Events Per Second)
Fill factor	8.1 %

Table 2.2: DVS specifications [23]

2.5 High dynamic Range in DVS

In general the dynamic range is defined as the ratio of maximum light intensity measurable (at pixel saturation), to minimum light intensity measurable (above read-out noise).In the context of DVS events, it can be defined as the ratio of maximum to minimum scene illumination at which the events can be generated by high contrast stimuli. The HDR arises from the logarithmic compression in the front-end photoreceptor circuit and the local event-based quantization. Each pixel has its own circuit for the analog processing of the fractional change in illumination and thus it has a local gain control. Therefore pixels can individually adapt to bright or dark areas.

The photodiode dark current of 4 fA at room temperature limits the lower end of the dynamic range. Events are generated reliably ,down to less than 0.1 lux scene illumination using a fast f/1.2 lens. The sensor also operates up to bright sunlight scene illumination of 100 Klux. Thus the 120 dB stems from the pixel sensitivity (0.1 lux- moon light) and saturation limit (100klux-bright sunlight scene), resulting in the 6 decades of dynamic range − thanks to the design in which the pixels are sensitive to the time derivative of logarithm of the intensity (see equation 2.1).

2.6 jAER : Application to log and view DVS events

jAER is an acronym for "Java Address-Event Representation". jAER is an open-source Java-based GUI for PCs for visualisation of real-time or recorded event-based data and rapid development of real-time event-based algorithms and applications, licensed under the GNU Lesser General Public License (LGPL) v2.1.More information regarding jAER is described in [24]. The figure 2.4 shows a glimpse of the jAER software recording a moving hand.

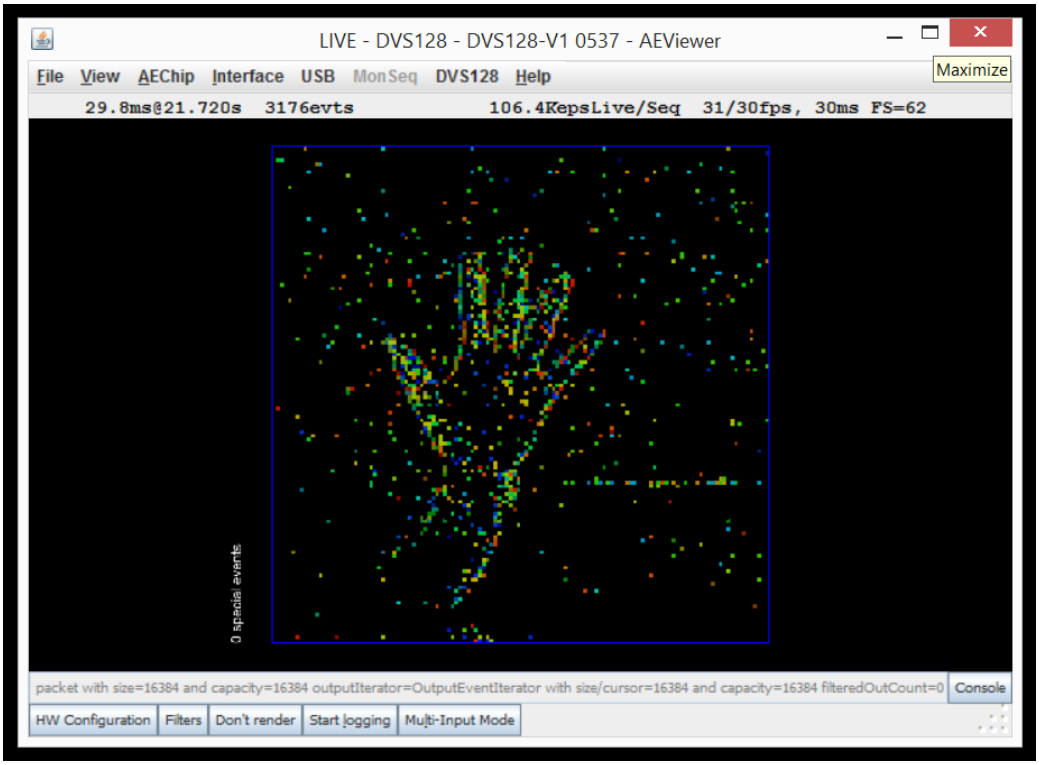

Figure 2.4: jAER Viewer to log DVS events

CHAPTER 3

Hand Gesture Recognition using simulated DVS events

The aim of this project is to recognise and classify hand gestures from DVS camera outputs (address events). Since the physical camera was not available in the initial stages of this project work, the DVS events were generated using simulations. The RGB frames in the video dataset [13] are converted to event dataset using the basic principle of generation of DVS events(as explained in section 2.3).

3.1 Flow of Hand gesture Recognition Model

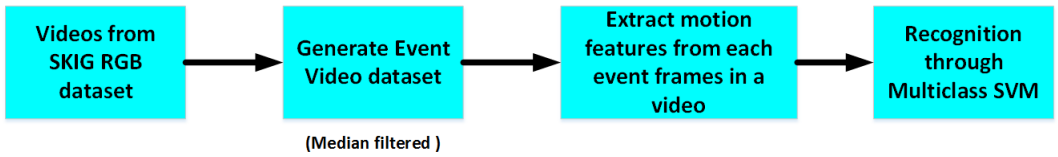

Figure 3.1: Basic Block Diagram for our Hand Gesture Recognition Model

The basic flow of our hand gesture recognition model using simulated DVS data is as shown in the figure above. Each video from the RGB dataset is converted into an event video by applying the basic principles of event generation and that forms the event dataset.Then various feature extraction methods are used on the events to generate a feature matrix and feed that to a multiclass SVM network for the recognition of various classes. Each blocks will be explained in detail in the following sections.

3.2 RGB Dataset Used for simulation of DVS : SKIG

Sheffield KInect Gesture (SKIG) Dataset [16][27] were used and it has a total of 2160 hand gesture sequences (1080 RGB sequences and 1080 depth sequences) collected from 6 subjects. In this thesis only 1080 RGB sequences were utilised for the gesture classification, since the depth videos cannot be used to properly generate the events.

There were 10 categories of hand gestures in total : circle (clockwise), triangle (anti-clockwise), up-down, right-left, wave, "Z", cross, comehere, turn-around, and pat.And all these gestures were performed with three hand postures : fist , index and flat in 3 different backgrounds (i.e., wooden board, white plain paper and paper with characters) with 2 illumination conditions (i.e., strong light and poor light. 10 gestures x 3 poses x 3 back grounds x 2 iluminations x 6 subjects forms 1080 RGB sequences which is used to generate the corresponding 1080 DVS gesture sequences.

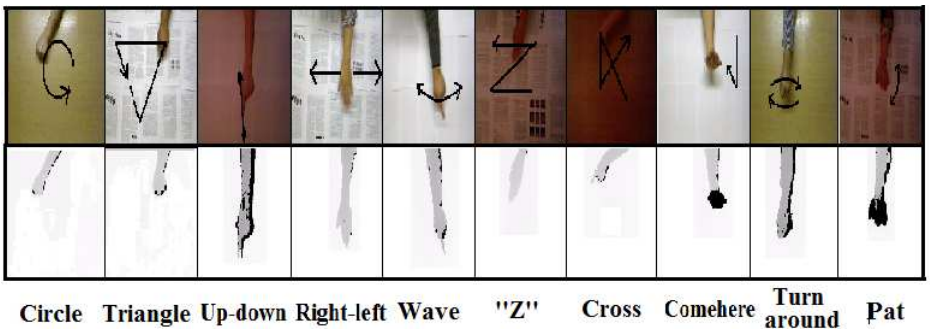

Figure 3.2: Gestures from SKIG RGB Dataset[27] used to generate DVS dataset

3.3 Simulation of DVS events

The flowchart given in figure below explains how one event frame is generated using the two consecutive frames from an RGB video.I(t) and I(t+1) are the image frames at time t and t+1 respectively and the T is the threshold value used to generate DVS events which is arbitrarily set at 0.1.

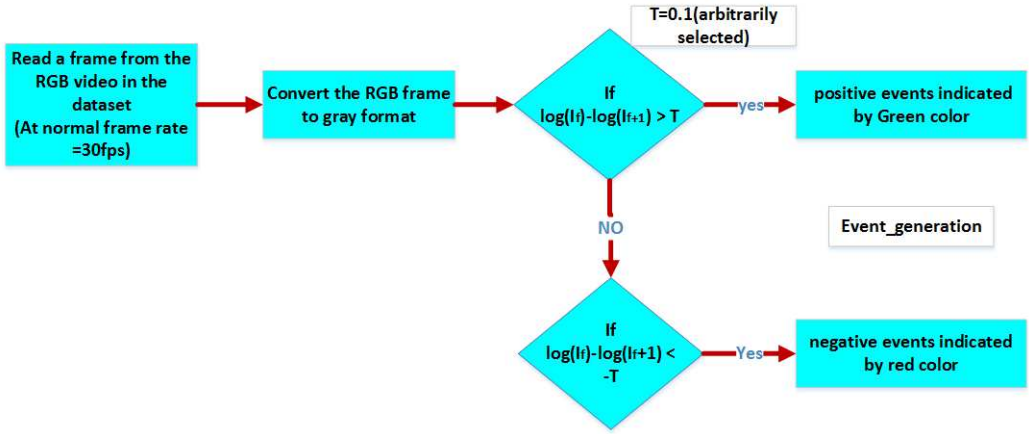

Figure 3.3: Basic flow for the event generation

All the 1080 videos from the RGB dataset are converted to its corresponding event videos comprising event frames.Each of the generated event frames are median filtered

with a block size of 5x5 so that the hand movements are clearly extracted out from the noisy events. The output of DVS sensor is sequence of events indicated by (x,y,t,e), where x, y correspond to co-ordinates on the sensor array, t is the time stamp and e indicates a positive or a negative event. Figure .5 shows the simulated event frames corresponding to the hand gesture of "circle". Stack of these consecutive DVS frames are used as inputs frames for feature extraction, instead of RGB frames.

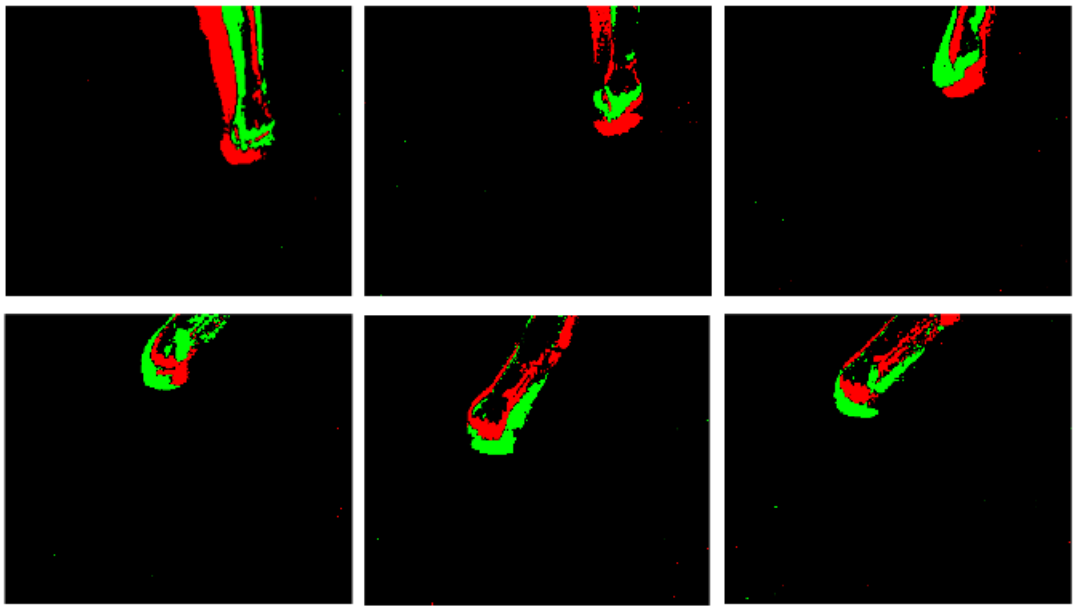

Figure 3.4: Event sequences for a hand movement in a circular fashion, red color showing positive events and green showing the negative events.

Apart from median filtering, unwanted events are filtered out using connected component methods. The flowchart shown in figure below explains the way the unwanted noisy events are filtered out using connected component method. If the number of connected pixels is greater than a threshold then only that pixel is maintained as a connected component.

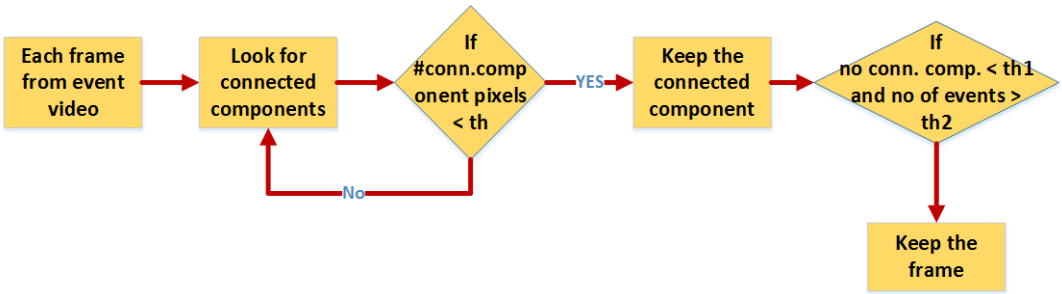

Figure 3.5: Removing noisy events using connected components

Since the DVS has a different architecture and data representation , it is not that straightforward to apply conventional algorithms for tracking or recognising objects to

DVS based applications. Some feature such as the hue of the pixel that can be easily obtained from the conventional cameras are not available in DVS. On the other hand some of the new features that are not easily available on normal vision sensors become readily available in DVS. In this thesis , the properties of DVS were explored to find useful features for the classification.

3.4 Feature extraction

Feature extraction is an essential step for the classification, but selection of good features is always challenging. In the case of DVS , it is even more interesting since features are to be extracted from the address events not from the RGB pixel values like in normal cameras.The following sections explains the various features that are explored on the simulated dataset.

3.4.1 (x,y) co-ordinates of the tip of the hand

In this, we estimate the tip of the hand region and use that as a feature.

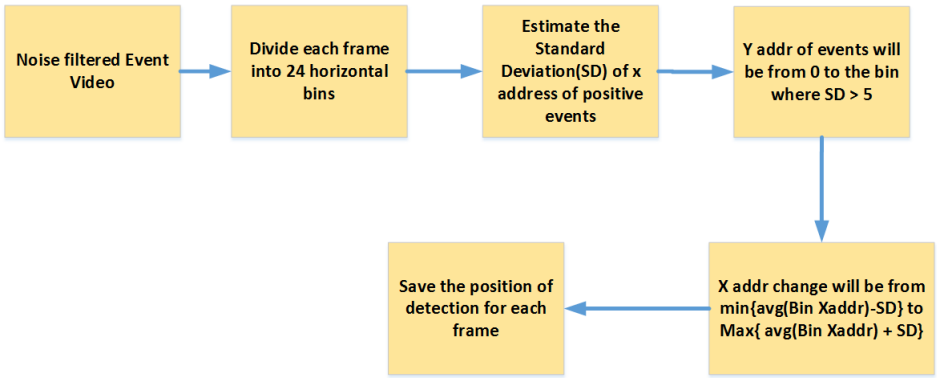

Figure 3.6: Block diagram for detection of arm

The figure 3.6 explains the algorithm used to detect the arm region alone in a noise filtered event frame.The entire event frame is divided into a fixed ,say N number bins horizontally as shown in figure 3.7. Each bin can be considered as a 2D rectangular slice.The column address is mentioned as the x address and the row as y.the The algorithm calculates the standard deviation of column address of all positive events. Then row address is taken to be from row 0 to the row where the standard deviation is greater than a threshold.Similary the column address is also decided using the logic shown in

the block diagram.The detected arm region in one of the event frame(blue rectangular box) is shown in the figure 3.7. The median of the (x,y) locations in the first and the last bin is taken as the feature.

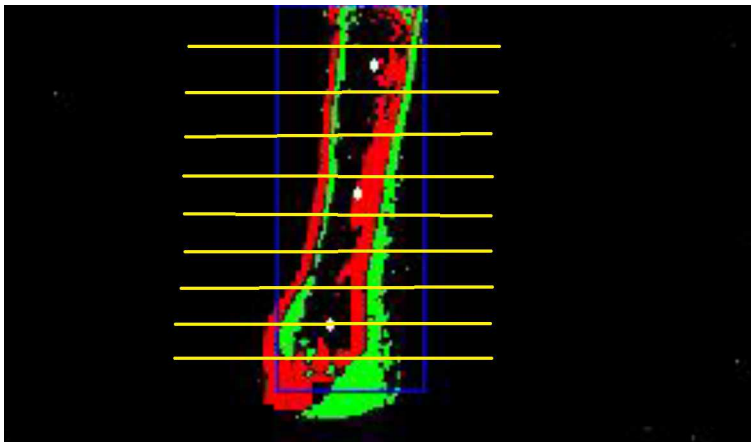

Figure 3.7: The detected Arm and the binning

3.4.1.1 Dynamic Time Warping(DTW) for classification

DTW [33] is an algorithm for measuring similarity between two temporal sequences which may vary in speed.Eg: similarities in hand gestures could be detected using DTW, even if some person shows the same gesture a little bit faster than the other.DTW of two vectors x and y basically stretches them onto a common set of instants such that the sum of the Euclidean distances between corresponding points, is the smallest.In our work we applied DTW algorithm between train and test data and estimated the cost each time. Then finally sorted the cost and took the smallest cost to make the prediction.

3.4.2 Compressed events over (x,y,t) cuboids

Another feature extraction method tried was compressed events and we call it as "cuboid" features. Basically we applied a thresholding over the number of events occurred over the time. This helped very much in reducing the noisy backgrounds in illumination2. The figure 3.8 shows the flow of the cuboid generation.The newly assigned 1âĂŹs and 0âĂŹs forms the feature vector (compressed events) , which is then passed to linear SVM stage.

The feature dimension was very large of the order of 36000 for a cuboid size of 5x5x5 , where as the number of samples were small(1080 videos).Linear SVM may not

be an ideal solution in this case.Hence we applied PCA(Principle Component Analysis) for dimensionality reduction to improve the accuracy.

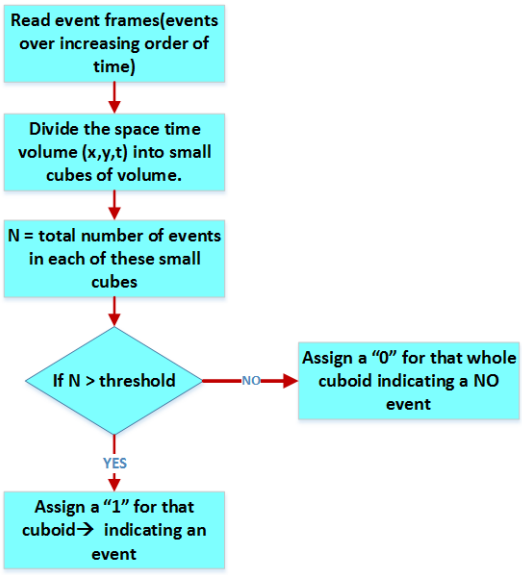

Figure 3.8: Flowchart for cuboid generation

3.4.3 Contour based features

Contour-based features are directly extracted from contour boundary coordinate [2].There are many types of contour-based features such as the Cartesian Coordinate Feature (CCF), the Fourier Descriptor Feature (FDF) [8, 11, 6], Centroid-Distance Feature (CDF) [34, 35], and ChordLength Feature (CLF) points . Out of these , CCF , FDF ,CDF and the contour centroid were tried. Since CCF is not giving good results it is omitted here.

3.4.3.1 Contour centroid

The positive and negative events will constitute a contour if the events are used to form an event video at 30 fps. This contour boundary points are estimated by using MATLAB command *bwboundaries*.The first object is always taken since that is considered to be the best approximation or we can say the significant boundary points. The mean of boundary points will give the centroid of the contour.The centroid in each event frame is given as a feature vector.The results are shown in figure 3.9

Iteration 1		Iteration 2		Iteration 3	
Input set used	Accuracy	Input set used (FULL SET)	Accuracy	Input set used	Accuracy
Illumination_1 ~~Illumination_2~~ actionType_1 actionType_2 actionType_3 actionType_4 actionType_5 actionType_6 actionType_7 actionType_8 ~~actionType_9~~ ~~actionType_10~~ backgroud_1 backgroud_2 ~~backgroud_3~~ pose1 pose2 pose3	90.625%	Illumination_1 Illumination_2 actionType_1 actionType_2 actionType_3 actionType_4 actionType_5 actionType_6 actionType_7 actionType_8 actionType_9 actionType_10 backgroud_1 backgroud_2 backgroud_3 pose1 pose2 pose3	69.7%	Illumination_1 ~~Illumination_2~~ actionType_1 actionType_2 actionType_3 actionType_4 actionType_5 actionType_6 actionType_7 actionType_8 actionType_9 ~~actionType_10~~ backgroud_1 backgroud_2 ~~backgroud_3~~ pose1 pose2 pose3	79.63%

Figure 3.9: Results with countour centroid as a feature

3.4.3.2 Fourier Descriptor feature(FDF)

Fourier descriptors are a way of encoding the shape of a two-dimensional object by taking the Fourier transform of the boundary, where every point (x,y) on the boundary is mapped to a complex number $x + iy$ [26].Then the list of coordinates is Fourier transformed using the DFT(Discrete Fourier Transform). The Fourier coefficients are called the *Fourier descriptors*.The basic shape of the region is determined by the first several coefficients, which represent lower frequencies.Higher frequency terms provide information on the fine detail of the boundary.

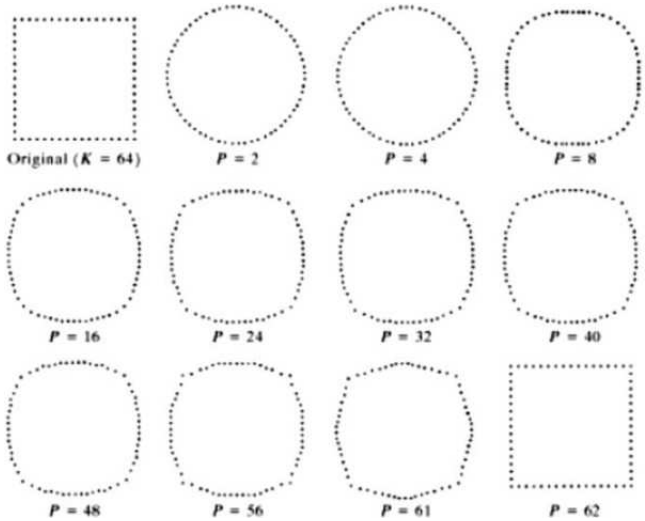

Figure 3.10: Examples of reconstruction from Fourier Descriptors. P is the number of fourier coefficients used in the reconstruction of the boundary

Figure 3.11: FDF example 1

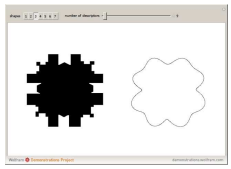

Figure 3.12: FDF example 2

Figure 3.13: Examples for Fourier Descriptors

The original shape can be recovered from the inverse Fourier transform. However, if only a few terms of the inverse are used, the boundary becomes simplified, providing a way to smooth or filter the boundary.

Steps involved in calculating the Fourier descriptors of a shape :

1. Contour extraction : Find the coordinates of the edge pixels of a shape and put them in a list in order, going clockwise around the shape.
2. Contour Normalisation : Before applying Fourier transform on the shape boundary, it is first normalized for matching purposes.This is done by sampling the boundary of each shape to have the same number of data points. There are three methods for normalization -equal angle sampling, equal points sampling, and equal arc-length sampling.In this thesis work , equal points sampling is used.Number of boundary points in each event frame in a video is made equal to the lowest boundary length within that video(good to do this before FFT is taken !)
3. Define a complex-valued vector using the coordinates obtained. For example:if a boundary point is (3,4) , it is written as $3 + 4i$
4. Take the discrete Fourier transform of the complex-valued vector.

Properties of Fourier descriptors inherited from the Fourier transform

1. **Translation invariance:** No matter where the shape is located in the image, the Fourier descriptors remain the same.To achieve translation invariance the first coefficient from the Fourier Transform which contains the shape position is set to zero, C0 = 0
2. **Scaling:** If the shape is scaled by a factor, the Fourier descriptors are scaled by that same factor.Scale invariance is achieved by dividing the magnitude values of all the descriptors by the magnitude value of the second descriptor which is the size of the shape.
3. **Rotation and starting point:** Rotating the shape or selecting a different starting point only affects the phase of the descriptors [23].Since the rotation of the shape affects only the phase information rotation invariance is achieved by taking only the magnitude values of the FDs and ignoring the phase information.

Because the discrete Fourier transform is invertible, all the information about the shape is contained in the Fourier descriptors. A common thing to do with Fourier descriptors is to set the descriptors corresponding to values above a certain frequency to zero and then reconstruct the shape. The effect of this is a low-pass filtering of the shape, smoothing the boundary. Since many shapes can be approximated with a small number of parameters, Fourier descriptors are commonly used to classify shapes.FDF is efficient in terms of speed as they only use a small number of points from the entire image.

But unfortunately FDF alone will not give good accuracy for classification since it is just a shape based feature and does not track the movements of the objects very well. So a combination of centroid and FDF was tried (next section).

3.4.3.3 Fusion of Contour Centroid and FDF

When we fused the shape based feature FDF with the centroid feature which can be used to track the flow of that shape , better results are obtained.The comparison of accuracies is as shown in the table 3.22 in the experimental results section.

3.4.4 Optical FLow

Optical flow or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer and a scene.Sequences of ordered images allow the estimation of motion as either instantaneous image velocities or discrete image displacements [32].The optical flow methods try to calculate the motion between wo image frames which are taken at times t and $(t + \Delta t)$ at every voxel position. For a 2D+t dimensional case (3D or n-D cases are similar) a voxel at location (x,y,t) with intensity I(x,y,t), will have moved by Δx , Δy and Δt between the two image frames, and the following brightness constancy constraint can be given

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

Assuming the movement to be small, the image constraint at $I(x, y, t)$ with Taylor series can be developed to get:

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t + H.O.T.$$

From these equations it follows that:

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0$$

$$\Rightarrow \frac{\partial I}{\partial x}\frac{\Delta x}{\Delta t} + \frac{\partial I}{\partial y}\frac{\Delta y}{\Delta t} + \frac{\partial I}{\partial t}\frac{\Delta t}{\Delta t} = 0$$

$$\Rightarrow \frac{\partial I}{\partial x}V_x + \frac{\partial I}{\partial y}V_y + \frac{\partial I}{\partial t} = 0$$

where V_x,V_y are the x and y components of the velocity or optical flow of I(x,y,t) and $\frac{\partial I}{\partial x}$, $\frac{\partial I}{\partial y}$ and $\frac{\partial I}{\partial t}$ are the derivatives of the image at (x,y,t) in the corresponding directions. I_x, I_y and I_t can be written for the derivatives in the following.Thus: $I_xV_x + I_yV_y = -I_t$

This is an equation in two unknowns and cannot be solved as such. This is known as the aperture problem of the optical flow algorithms. To find the optical flow another set of equations is needed, given by some additional constraint. All optical flow methods introduce additional conditions for estimating the actual flow.

There are variety of methods to estimate the optical flow - phase correlation based , block based (SAD, cross correlation) ,differential methods , discrete optimisation methods etc.

In this thesis work we selected one of the differential methods viz. Lukas Kanade method.The **Lucas-Kanade method** divides the original image into smaller sections and assumes a constant velocity in each section.Then, it performs a weighted least-square fit of the optical flow constraint equation to a constant model for $[V_x, V_y]^T$ in each section.

The Lukas Kanade method was applied only on some corner points estimated using a Harris Detector.Corners are regions with two different directions of gradient (at least).Aperture problem disappears at corners. At corners,the 1st order approximation fails.The visualisation of the flow vectors at the corner points is as shown in figure 3.14

Figure 3.14: Visualisation of optical flow at corner points

3.4.5 DVS Motion Maps

DVS event video can be considered as an (x,y,t) volume. DVS motion maps are the various slices of this volume such as x-y , y-t, x-t slices averaged over the other axes.For example ,x-y motion map is the time averaged x-y slices.Similarly if we average the y-t values along x axis we get y-t motion maps and if x-t values were averaged along y axis , we gets x-t motion maps. Normalisation was done before averaging to obtain the maps.The figure 3.15 shows the xy slices of all the 10 gestures in the SKIG event dataset.Obviously whichever gesture having an x-y motion ,the x-y motion map becomes a good feature.But for depth related gestures it does not deliver much.

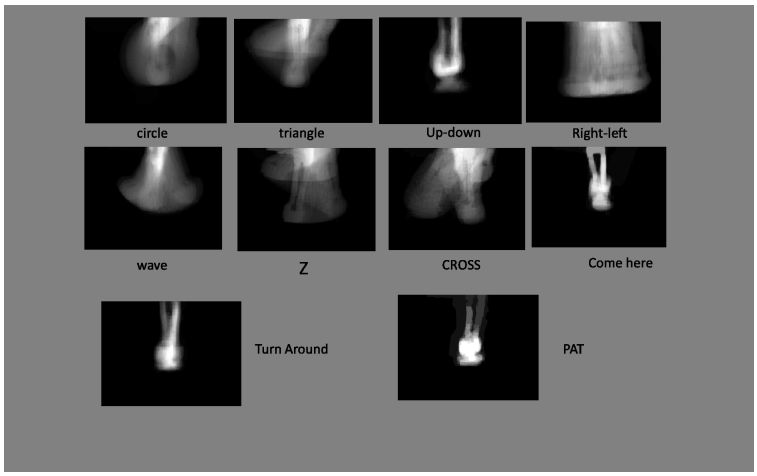

Figure 3.15: x-y slices of various gestures in SKIG event dataset

The figure 3.18 shows the x-t motion maps of the circle and the triangle gestures.Similarly the figure 3.21 shows the y-t motion maps of the two gestures.

The best classification **accuracy of 85 %** was given by the yt motion maps when the noisy background3 , illumination2 and the depth related action10 was removed

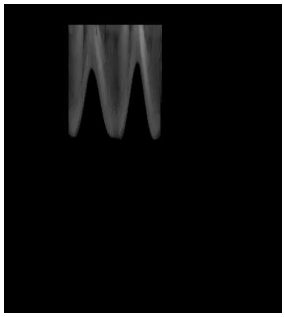

Figure 3.16: circle

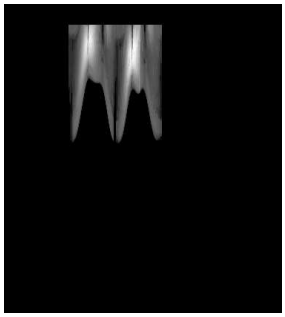

Figure 3.17: traingle

Figure 3.18: x-t Motion Maps

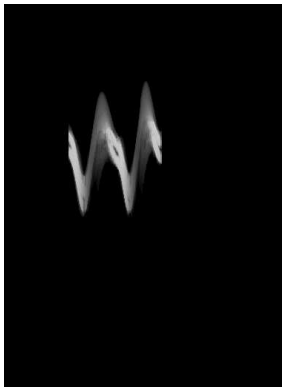

Figure 3.19: circle

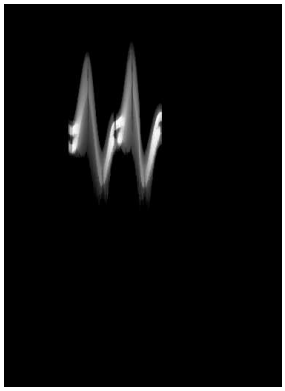

Figure 3.20: traingle

Figure 3.21: y-t Motion Maps

and when we applied ***Principal Component Analysis(PCA)*** **for dimensionality reduction**.Without performing PCA , the accuracy was 76 %

3.5 Classification

The data samples available were very less(total number of videos itself was 1080 only) and hence we did not use any of the neural network based models for classification.Instead,we selected the Support Vector Machine(SVM) [28] for the gesture classification. Since there are 10 gestures to be classified , multi-class SVM [28] is required to be used.In this thesis, we used the *svmtrain* and *svmclassify* functions available in matlab and the reference code used [19] is a one vs. all SVM model.The kernel function selected was Radial Basis Function (RBF) with a sigma of 0.1.Method used to find the separating hyperplane is Sequential Minimal Optimization(SMO). Least squares and quadratic programming were also tried which did not yield a better results than SMO.

3.6 Experimental Results

3.7 Summary

This section summarises the results of various features tried on the simulated event datasets. From the table 3.1 it is obvious that, by eliminting one dept related gesture,low illumination and noisy background from the dataset, the best classification accuracy was

Sl.No:	Features Tried	Classifier Params	Remarks on the Input Data	Classifiction Accuracy
1	Co-ordinates of Bottom tip of the hand	DTW(Dynamic Time Warping)	Without bg3 , illum2 , action10 Without bg3 , illum2 , action 8,9,10	81.48% **100%**
2	Compressed events over (x,y,t) cuboids of size 5x5x5	Linear SVM with PCA	With all gestures	55 %
3	Centroid from the contour boundaries	Linear SVM	Without bg3 , illum2 , action10 Without bg3 , illum2 , action 9,10	79.63% 90.63 %
4	FDF(normalised by fd(2))	Linear SVM	Without bg3 , illum2 , action10	38 %
5	Centroid + FDF (normalised by fd(2))	Linear SVM	Without bg3 , illum2 , action10	67.7 %
6	Optical flow vectors (Vx, Vy) of corner points (Lucas Kanade method)	Linear SVM	Without bg3 , illum2 , action10	74 %
7	y-t DVS motion maps with PCA	Linear SVM	Without bg3 , illum2 , action10	85.2 %

Table 3.1: Consolidated Classification accuracy of various features tried on simulated DVS dataset

Iteration 1				Iteration 2			Iteration 3			
Input set used	Accuracy (only centroid)	Accuracy (centroid + FD) -	Accuracy (centroid + FD) -	Input set used (FULL SET)	Accuracy (only centroid)	Accuracy (centroid + FD)	Input set used	Accuracy (only centroid)	Accuracy (centroid + FD)	Accuracy (centroid + FD)
Illumination_1 ~~Illumination_2~~ actionType_1 actionType_2 actionType_3 actionType_4 actionType_5 actionType_6 actionType_7 actionType_8 ~~actionType_9~~ ~~actionType_10~~ backgroud_1 backgroud_2 ~~backgroud_3~~ pose1 pose2 pose3	90.625%	82.3% (FDF normalised with Max value)	70.8% (FDF normalised with 2nd FD value)	Illumination_1 Illumination_2 actionType_1 actionType_2 actionType_3 actionType_4 actionType_5 actionType_6 actionType_7 actionType_8 actionType_9 actionType_10 backgroud_1 backgroud_2 backgroud_3 pose1 pose2 pose3	69.7%	58 %(FDF normalised with 2nd FD value)	Illumination_1 ~~Illumination_2~~ actionType_1 actionType_2 actionType_3 actionType_4 actionType_5 actionType_6 actionType_7 actionType_8 actionType_9 ~~actionType_10~~ backgroud_1 backgroud_2 ~~backgroud_3~~ pose1 pose2 pose3	79.63%	74.1 % (FDF normalised with Max value)	66.67% (FDF normalised with 2nd FD value)

Figure 3.22: Results with the fusion of FDF and countour centroid

given by the y-t motion map feature(85%), followed by the naive features like hand tip co-ordinates (81.5 %) , the contour centroid (79.63 %) and finally the optical flow with 74 % .

It was observed that for the dataset we selected (SKIG) , the yt motion map gave better results. There were 4 depth related gestures - *up-down* , *come-here* , *turn around and pat* . Obviously for all these gestures xy motion map alone will not produce good results.But a combination of all the slices will have to give a better gesture recognition accuracy. But as shown in table 3.2 we get a relatively less accuracy of 68.52% when comapared to the higher accuracy for the yt map alone.

It is to be noted that the accuracy increases irrespective of any feature ,with the elimination of depth related gestures.This left us with the thought of involving depth also as a feature , for which the gestures taken with stereo camera were required.

Feature description	**Classifier Params**	**Remarks on the Input Data**	**Classifiction Accuracy**
DVS motion maps : Normalised xy , xt , yt slices averaged on the 3^{rd} axis	SVM with linear Kernel	All 3 slices (Without bg3 , illum2 , action10)	68.52 %
		Only xy slice (Without bg3 , illum2 , action10)	65.74%
		Only xt slice Without bg3 , illum2 , action10	79.63%
		Only yt slice Without bg3 , illum2 , action10	85.2 %

Table 3.2: Analysis of results from DVS motion map feature on simulated DVS dataset

CHAPTER 4

Creation of DVS and Stereo Hand Gesture Dataset

Through survey it was observed that no DVS hand gesture dataset was publicly available. So was the case with the stereo hand gestures.In this thesis work, a small dataset for both the DVS and the stereo hand gesture dataset were recorded.The gestures recorded were commonly used in Automotive or Augmented Reality platforms.The DVS gestures were recorded through the open source software *jAER*, which records it in *aedat* format. The stereo gestures were recorded as textitavi files using the 3rd party software *NCH suite*.

4.1 IITM-DVS gesture dataset

We have recorded 10 gestures with 10 sets of each gesture from 12 subjects . Thus a total of **1200 gestures** were captured. The 10 gestures are *right swipe , left swipe , swipe up , swipe down , rotate clockwise , rotate counter clock wise , X , V , Z and "come here" action*. Out of these ,the only depth related gesture is the "come here" action. The DVS recordings of first 6 subjects had one RGB camera(webcam) for the ground truth reference and has 10 sets of 10 gestures thereby constituting 600 gestures in total.

4.2 Stereo Gesture Dataset

For the next 6 subjects , we recorded gestures with a Stereo Camera(explained in section 6.3) fixed above the DVS. This time we added two more depth related gestures to experiment with .Thus 10 sets of each of the 12 gestures were recorded which forms a total of 720 stereo and DVS gestures.

Figure 4.1: Set-up for DVS gesture recording

Figure 4.2: Front View

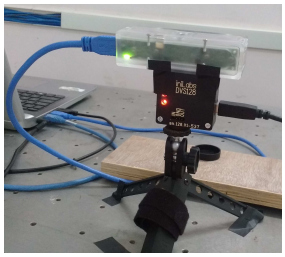

Figure 4.3: Rear view

Figure 4.4: Set-up for stereo gesture recording

CHAPTER 5

Gesture Recognition using IITM- DVS Dataset

5.1 Block Diagram

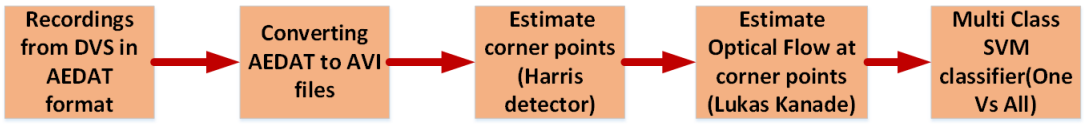

Figure 5.1: Detailed Flow Diagram for gesture recognition with optical flow using real DVS data

The detailed flow diagram for the gesture recognition performed on the real data recorded from the DVS is shown in 5.1.In this thesis, we use frame based techniques to analyse how it works with DVS events.Also since we records gestures relatively at a lower speed ,it does not matter much even if we sample events at 1/30 seconds. Therefore ,the recorded aedat file which contains the x,y locations of all events and the time stamps in microseconds are converted to avi files at 30 frames per second for easiness.Once the aedat files are converted to video files the same optical flow algorithm mention in section 3.4.4 is used to form a feature vector for the multi SVM classifier described in 3.5 .

Apart from the optical flow algorithm ,the yt motion map (as such without any clustering and KNN) which gave better results in the simulated data was also tried for comparison.But in this case, the accuracy obtained was only 61.82 % .

To improve the results, the concept of the dense trajectories [29] were used in which the tracking is done using the optical flow vectors of some densly sampled points over the video frames. The (x,y,t) volume is divided into many cuboids and descriptors like HOG , HOF are calculated within each cuboids.Once we extract these descriptors Bag of words technique is applied separately on them. The code for dense trajectories and the descriptor extraction was publicly available and we used that to improve the results on our DVS dataset. As expected very good performance were obtained as shown in the table 6.1

5.2 Experimental Results

Sl.No:	Features Tried	Classifier Params	Remarks on the Input Data	No: of classes	Classifiction Accuracy
1	Optical flow vectors (Vx, Vy) of corner points (Lucas Kanade method)	Linear SVM(12 fold cross validation)	12 subjects	10 gestures	79.18 %
			6 subjects	11 gestures(pull hand excluded)	69.7 %
			6 subjects	12 gestures	63.6 %
2	y-t DVS motion maps with PCA	Linear SVM	12 subjects	10 gestures	61.82 %
			6 subjects	11 gestures(pull hand excluded)	59.4 %
			6 subjects	12 gestures	56.74%
3	Dense Trajectory followed by HOG HoOF Fusion and BoW BoW and Fusion	Linear SVM	12 subjects	10 gestures	 86% 96.91 % 96.58% **97.91** %
4	Dense Trajectory followed by HOG HoOF Fusion and BoW BoW and Fusion	Linear SVM	6 subjects	12 gestures	 73.75% 89.44 % 90.42% **91.53** %

Table 5.1: Consolidated Classification accuracy of some good features tried on real DVS dataset

5.3 Summary

This section summarises the results of some of the best and popular features tried on the real DVS data captured at IITM. The optical flow algorithm gives its best for the 10 gesture dataset of 12 subjects by giving 79.18 % and the accuracy comes down when two more depth related gestures - *push hand* and *pull hand* is added to it.It is observed that the if the Bag of Words(BoW) performed over HOG and HOF is done first and then the histograms are fused afterwards ,that gives the best result as shown in the sl.no: 3

of the table 6.1 . The Sl,no: 4 shows the result for the same as explained above but with 6 subjects and 12 gestures with the addition of the 2 depth related gestures.This is to be compared against the results from simulated stereo DVS in the next chapter.

CHAPTER 6

Gesture Recognition using simulated Stereo DVS

This chapter explains the simulation of stereo DVS using a normal monochromatic stereo camera . Since there was no two DVS cameras available to achieve stereo vision, we used a normal stereo camera and converted the left and right channel frames to event frames.

6.1 Why using Stereo DVS ?

Stereo vision yields depth information. Therefore, the effect of the background movements can be reduced by filtering out the irrelevant events from the scene behind the movement plane of the hand.That is ,robustness to background movements is acheivable.Another advantage is that depth related gestures will get more recognition accuracy with stereo DVS compared to one DVS camera alone.

6.2 Basic Flow of the Model

The figure 6.1 shows the basic steps involved in simulating a stereo DVS and the gesture recognition using the depth also as a feature combined with the optical flow vectors.

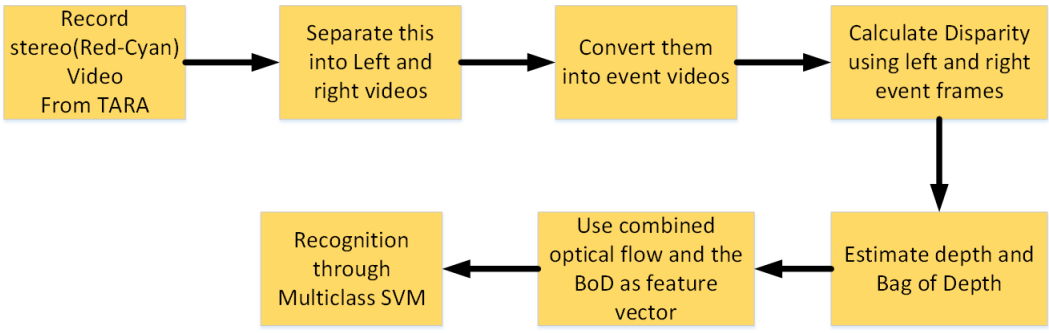

Figure 6.1: Flow Diagram for the gesture recognition using simulated DVS

The various steps in the flow diagram will be explained in the following sections.We recorded the stereo hand gestures using the stereo camera - *TARA*, which is briefy explained in section 6.3

6.3 Monochromatic Stereo Camera - See3CAM_Stereo (TARA)

Tara is a UVC complaint USB 3.0 SuperSpeed Stereo vision camera from e-con Systems [7].It is based on MT9V024 stereo sensor from OnSemi imaging which supports WVGA (752x480) at 60fps over USB 3.0 in uncompressed format.

Formats and resolutions supported :Y16 format and RGB24 formats (8bpp and 10bpp respectively).

- 752 x 480 (752x480 from each sensor) at 60 fps in USB 3.0 and 30 fps USB 2.0
- 40 x 480 (320x480 from each sensor) at 60 fps in USB 3.0 and 30 fps USB 2.0
- 20 x 240 (320x240 (binned from 640x480) from each sensor) at 60 fps in USB 3.0 and USB 2.0.

Figure 6.2: Stereo Camera - TARA

It is capable of streaming camera frames in WVGA resolution at 60 fps when connected to USB3.0 host port by leveraging the full throughput of USB3.0. Other resolutions supported are VGA (cropped) at 60 fps and QVGA at 60 fps. The **base line of this stereo camera is 60 mm** . This camera comes with a **6-axis Inertial Measurement Unit (IMU)**, which comprises a 3D accelerometer and a 3D gyroscope in it. The accelerometer in the IMU is useful for measuring the linear accelerations and the gyroscope helps in measuring the angular accelerations. The Stereo camera uses S-mount lens holders and any compatible M12 lenses can be used on this. Any change in the lens or lens position will certainly require lens re-calibration.

The Tara can be connected to the PC through the USB cable and using the aplication "eCAMview" the left and right images live streaming can be done. But for recording the gestures we used a 3rd party software(NCH suite) that records the combined left and right (red-cyan) stereo video.

6.4 Getting Stereo Parameters Through calibration

Camera calibration is the process of estimating parameters of the camera using images of a special calibration pattern. The parameters include camera intrinsics(lens) , distortion coefficients, and camera extrinsics(pose) Even though the TARA was pre-calibrated before delivery ,the stereo parameters were not available with us.So in order to find out the parameters , we have calibrated and estimated the parameters using the MATLAB application " *Stereo Camera Calibrator*". For the calibration, almost 20 pairs of left and right images of checker board pattern were captured using TARA (using eCAMviewer) and loaded to the Matlab App.The stereo parameters obtained thereby are used later for the calculation of disparity.The figure 6.5 shows one of the left-right pairs of the checkerboard pattern loaded to the *Stereo Camera Calibrator*.

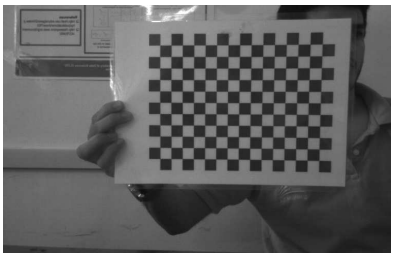

Figure 6.3: Image from right eye

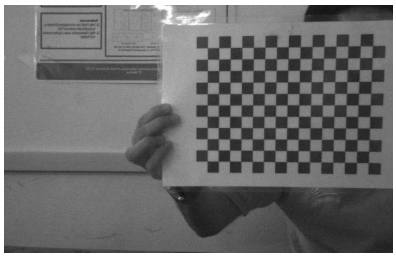

Figure 6.4: Image from left eye

Figure 6.5: Left and Right checkerboard images loaded to Matlab App for calibration

6.5 Stereo rectification

Stereo matching is used to correlate points from one digital image of a stereo pair with the corresponding points in the second image of the pair. Using the stereo Parameters obtained from the Matlab Calibration app , stereo rectification between the left and right images is carried out.Stereo image rectification [31] basically projects the pair of images onto a common image plane in such a way that the corresponding points have

the same row coordinates. This image projection makes the image appear as though the two cameras are parallel.

6.6 Algorithms attempted for estimating Disparity or Depth

The basic stereo algorithm says for each epipolar line and for each pixel in the left image, compare with every pixel on same epipolar line in right image and pick that pixel with minimum match cost as the corresponding point.In general there are **local and global stereo matching** methods. Local methods includes SSD(Sum of Squared Differences) , SAD (Sum of Absolute Differences) , Zero-mean Normalized Cross-Correlation (ZNCC) etc.

In a global method, the matching between a pixel in the left image and a pixel in the right image does not depend only on their neighbours, but also on the matches of their neighbours. Hence, the match of a pixel influences the matches of its neighbour pixels. There are mainly two global methods: dynamic programming and graph cuts.

In this thesis we attempted SGBM and the graph cut methods.

6.6.1 Semi Global Block Matching (SGBM)

One of the algorithms used in this thesis were the Semi-Global Block Matching(SGBM) [9] . In the basic 'BlockMatching' method [13], the function computes disparity by comparing the sum of absolute differences (SAD) [30] of each block of pixels in the image. In the 'SemiGlobal' matching method, the function additionally forces similar disparity on neighboring blocks. This additional constraint results in a more complete disparity estimate than in the 'BlockMatching' method.

The algorithms perform these steps:

- Compute a measure of contrast of the image by using the Sobel filter.
- Compute the disparity for each pixel in the left Image.
- Mark elements of the disparity map, that were not computed reliably.

The computed disparity map has the same size as the input images since for each of the corresponding pixel in the reference image(left image), the disparity value is calculated.The range of disparity set for *TARA* can be either 0 to 128(better) or 0 to 64 .

6.6.2 Graph Cut

The basic theory is explained in [36][4]. Computing the disparity can be formulated as a graph labelling problem which can be solved using Markov Random Fields (MRF). MRF is a generative model often used in Image Processing and Computer Vision to solve labelling problems. It is a set of random variables having a Markov property (memoryless)described by an undirected graph.The energy minimization task is solved by some popular global optimization methods, i.e. Graph Cut and Belief Propagation.

In a labelling problem, we have a set of sites and a set of labels. Sites represent features extracted from the image (pixels, segments, etc.), for which we want to estimate some quantity. Labels represent the quantities associated to these sites: intensity, disparity, region number, etc.

In the case of Disparity Map Calculation , each pixel corresponds to a graph node and each disparity to a label.The detailed theory behind Graph Cut-based Stereo matching is given in the reference [36] The goal is to find a labeling $f : P \rightarrow L$ which minimizes

$$E(f) = E_{data}(f) + \lambda E_{prior}(f) = \sum_{p \in P} D_p(f_p) + \lambda \sum_{p,q \in N} S(f_p, f_q) \qquad (6.1)$$

P represents the set of pixels and L represents a discrete set of labels corresponding to different disparities. $D_p(f_p)$ is the cost of assigning label fp to pixel p and it is given by the SSD value calculated for the disparity corresponding to label f_p(ie., each disparity value).N is the set of neighboring pixels and $S(f_p, f_q)$ is the cost of assigning labels fp and fq to neighboring pixels p and q.The idea is to penalize neighboring pixels having different labels.

In this thesis the graph cut algorithm was applied to a standard stereo image pair (Tsukuba pair) and then compared with the results from SGBM .

Eventhough the literature survey indicates better results for the grapgh cut methods , we obtained comparitively a better disparity with SGBM. So we decided to use SGBM for the calculation of disparity.

6.7 Experimental Results

Sl.No:	**Features Tried**	**Classifier Params**	**Remarks on the Input Data**	**No: of classes**	**Classifiction Accuracy**
1	Optical flow only (used only the left eye videos)	Linear SVM(6 fold cross validation)	6 subjects	10 gestures 11 gestures(pull hand excluded) 12 gestures	72.33 % 70.61 % 64.44 %
2	Fusion of Optical flow and the depth Histogram	Linear SVM(6 fold cross validation)	6 subjects	10 gestures 11 gestures(pull hand excluded) 12 gestures	73 % 69 % **67.2% (improved over single DVS- 64.44%)**
3	Dense Trajectory followed by HOG HoOF Fusion and BoW BoW and Fusion	Linear SVM	6 subjects (used only the left eye videos)	12 gestures	 49.17% 80 % 81.67% **87.5** %

Table 6.1: Consolidated Classification accuracies on simulated Stereo DVS dataset

6.8 Summary

This chapter explains the simulation of the simplest stereo DVS using a normal monochromatic stereo camera.As expected we observed that the clasification accuracy with optical flow only as a feature using a single DVS is improved using the stereo DVS since now depth is also given as a feature along with the optical flow. The depth was estimated from events using the conventional SGBM algorithm only.

CHAPTER 7

Conclusion and Future scope

The main goal of this thesis was to explore the feasibility of performing Hand gesture recognition which is one of the functionalities performed by normal cameras in the AR systems. It explored the Dynamic Vision Sensors(DVS) in a way such that it gave us the confidence to work further towards the final goal of using it for various other functionalities done by cameras in current AR systems.Initially ,since DVS were not available and also none of the DVS hand gesture datasets were publicly available , we generated a synthetic event dataset from a normal RGB hand gesture data, using the basic principles of DVS . Various conventional features like shape ,contour , silhoutte and motion based vectors were tried with simulated DVS data and observed that good accuracies were obtained with the features like yt Motion maps ,contour centroid and optical flow.

We created our own DVS(1200 gestures) and stereo(Red-cyan) hand gesture dataset with popular gestures used in AR and Automotive industry. Classical algorithms like optical flow,HOOF(Histogram Of Optical Flow) and HOG (Histogram of gradients) were evaluated on this DVS dataset and obtained very good classification accuracies.

Finally we observed that, if the depth estimated from a simulated stereo DVS is augmented with the above features like optical flow etc , then there is an improvement in the classification results, especially when there are more depth related gestures.

So this thesis basically acts as a starting point towards our final goal as explained in 1.2. Since we dealt with no high speed gestures, the frame based techniques(captures at 30 fps) used in this thesis work gave us good results. But in future, in order to fully utilise the high speed capability of DVS , we should try processing the events at microsecond level time windows rather than (1/30)seconds windows.

Also the next experiment could be the capturing of hand gestures by fixing the DVS camera on a head mounted platform to emulate a head-mount AR device. In this case,to cancel the effects of head movements of the user ,the DVS shall be intergrated

with an Inertial Management Unit(IMU) to carry out the motion compensation.Since the stereo camera TARA already has an IMU in-built in it, the motion compensation can be experimented on that and the depth estimated can be utilised to extract only the region of interest(which is the person's arm length).

REFERENCES

[1] E. Y. Ahn, J. H. Lee, T. Mullen, and J. Yen, *Dynamic vision sensor camera based bare hand gesture recognition*, in Computational Intelligence for Multimedia, Signal and Vision Processing (CIMSIVP), 2011 IEEE Symposium on, IEEE, 2011, pp. 52–59.

[2] S. Al-Ali, M. Milanova, H. Al-Rizzo, and V. L. Fox, *Human action recognition: Contour-based and silhouette-based approaches*, in Computer Vision in Control Systems-2, Springer, 2015, pp. 11–47.

[3] S. Barua, Y. Miyatani, and A. Veeraraghavan, *Direct face detection and video reconstruction from event cameras*, in Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, IEEE, 2016, pp. 1–9.

[4] V. Boykov and Zabih, *Disparity Estimation by Graph Cut*, 2015.

[5] H. Brashear, T. Starner, P. Lukowicz, and H. Junker, *Using multiple sensors for mobile sign language recognition*, Georgia Institute of Technology, 2003.

[6] R. D. De Leon and L. E. Sucar, *Human silhouette recognition with fourier descriptors*, in Pattern Recognition, 2000. Proceedings. 15th International Conference on, vol. 3, IEEE, 2000, pp. 709–712.

[7] e-con Systems, *Stereo Camera - TARA*, 2017.

[8] R. Gonzalez, R. Woods, and S. Eddins, *Digital image processing using matlab gatesmark publishing*, (2009).

[9] H. Hirschmuller, *Accurate and efficient stereo processing by semi-global matching and mutual information*, in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 2, IEEE, 2005, pp. 807–814.

[10] Inilabs, *AEDAT file format*, 2017.

[11] H. Kauppinen, T. Seppanen, and M. Pietikainen, *An experimental comparison of autoregressive and fourier-based descriptors in 2d shape classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17 (1995), pp. 201–207.

[12] S. Kavadias, B. Dierickx, D. Scheffer, A. Alaerts, D. Uwaerts, and J. Bogaerts, *A logarithmic response cmos image sensor with on-chip calibration*, IEEE Journal of Solid-state circuits, 35 (2000), pp. 1146–1152.

[13] K. Konolige, *Small vision systems: Hardware and implementation*, in Robotics research, Springer, 1998, pp. 203–212.

[14] J. LEE, T. DELBRUCK, P. K. PARK, M. PFEIFFER, C.-W. SHIN, H. RYU, AND B. C. KANG, *Live demonstration : gesture-based remote control using stereo pair of dynamic vision sensors*, in Circuits and Systems (ISCAS), 2012 IEEE International Symposium on, IEEE, 2012, pp. 741–745.

[15] P. LICHTSTEINER, C. POSCH, AND T. DELBRUCK, *A 128x128 120 db 15 μ s latency asynchronous temporal contrast vision sensor*, IEEE journal of solid-state circuits, 43 (2008), pp. 566–576.

[16] L. LIU AND L. SHAO, *Learning discriminative representations from rgb-d video data.*, in IJCAI, vol. 4, 2013, p. 8.

[17] M. LOOSE, K. MEIER, AND J. SCHEMMEL, *A self-calibrating single-chip cmos camera with logarithmic response*, IEEE Journal of Solid-state circuits, 36 (2001), pp. 586–596.

[18] MICROSOFT, *Hololens*, 2017.

[19] A. MISHRA, *Multi Class Support Vector Machine,MATLAB file exchange*, 2015.

[20] P. MOLCHANOV, S. GUPTA, K. KIM, AND J. KAUTZ, *Hand gesture recognition with 3d convolutional neural networks*, in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 1–7.

[21] P. MOLCHANOV, X. YANG, S. GUPTA, K. KIM, S. TYREE, AND J. KAUTZ, *Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 4207–4215.

[22] M. OUHYOUNG AND R. LIANG, *A sign language recognition system using hidden markov model and context sensitive search*, in Procs. of ACM Virtual Reality Software and Technology Conference, 1996, pp. 59–66.

[23] R. G. RE, *Woods, digital image processing*, 2002.

[24] C. REINBACHER, G. GRABER, AND T. POCK, *Real-time intensity-image reconstruction for event cameras using manifold regularisation*, arXiv preprint arXiv:1607.06283, (2016).

[25] F. SAXEN, O. RASHID, A. AL-HAMADI, S. ADLER, A. KERNCHEN, AND R. MECKE, *Image-based methods for interaction with head-worn worker-assistance systems*, Journal of Intelligent Learning Systems and Applications, 2014 (2014).

[26] W. SETHARES, *Fourier Descriptor*, 2012.

[27] L. SHAO, *SheflňĄeld KInect Gesture (SKIG) Dataset*, 2015.

[28] J. A. SUYKENS AND J. VANDEWALLE, *Least squares support vector machine classifiers*, Neural processing letters, 9 (1999), pp. 293–300.

[29] H. WANG, A. KLÄSER, C. SCHMID, AND C.-L. LIU, *Action recognition by dense trajectories*, in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 3169–3176.

[30] WIKIPEDIA, *Sum of Absolute Differences*, 2015.

[31] ——, *Image rectification*, 2017.

[32] ——, *Optical Flow*, 2017.

[33] WIKIPEDIA1, *Dynamic Time Warping*, 2017.

[34] D. ZHANG AND G. LU, *Review of shape representation and description techniques*, Pattern recognition, 37 (2004), pp. 1–19.

[35] D. ZHANG, G. LU, ET AL., *A comparative study on shape retrieval using fourier descriptors with different shape signatures*, in Proc. of international conference on intelligent multimedia and distance education (ICIMADE01), 2001, pp. 1–9.

[36] A. ZUREIKI, M. DEVY, AND R. CHATILA, *Stereo Matching and Graph Cuts*, INTECH Open Access Publisher, 2008.

LIST OF PAPERS BASED ON THESIS

1. Authors.... Title... *Journal*, Volume, Page, (year).