Thesis Abstract

This thesis is a compilation of approaches aimed at event-based pattern recognition applications, in a variety of contexts. The first part of my thesis (Chapter 2) proposes solutions for static pattern recognition problems, where the patterns are primarily objects. Here, I document two different ways of realizing a classifier for event-based data streams. Moreover, I describe specifically target methods to make the learner invariant to numerous factors of variation, some of which can only be associated with event-based vision. I experiment with deep neural networks, such as convolutional neural networks, and also later with shallow networks, such as Extreme Learning Machines augmented with unsupervised feature learning techniques such as slow feature analysis. In both cases, we demonstrate excellent performance in event-based object recognition, in response to variations in orientation, pose, speed, motion direction and elastic distortions. The second part of my thesis (Chapter 3 and 4) demonstrates methods for dynamic pattern recognition problems, such as gesture recognition, based around intelligent feature learning and extraction. First, the proof-of-concept for a spatiotemporal feature learning method is demonstrated, where I conclusively show the benefits of learning features from raw event-data. The key inspiration again lies with finding features which show less variation in space and time, generated using slow feature analysis (SFA). We show that inculcating spatiotemporal slowness to the features makes them robust and invariant to visual transformations, such as translation, scaling and rotation. I show the applicability of such a feature learning method to the problem of feature tracking. Next, a slight modification of this approach is proposed for classifying spatiotemporal sequences of event-data. By integrating the excellent classification abilities of CNN with robust feature maps generated by my proposed method, the system outperforms the state-of-the-art in event-based gesture recognition.

Chapter 1

Event-based visual sensors: A novel approach to visual information acquisition

Asynchronously operating activity-driven sensors show a bio-inspired, radically different approach to sensory data acquisition. Be it vision ([LPD08a],[PMW08], [BBY$^+$14]) or sound [LvSMD10], asynchronous event-driven sensors provide a workaround to the drawbacks involved with conventional sensors. Visual event-based sensors such as the DVS [LPD08a] (Fig. 1.1) encode the accumulated, relative, pixel-intensity changes at each pixel. Such sensors are inspired from the brain, in the sense that the output produced by these sensors is a stream of spikes, through which they encode the sensory data.

Figure 1.1: Image of the Dynamic Vision Sensor, a commercially available neuromorphic event-based imager.

Conventional computer Vision relies on 'Frames' to acquire sensory data. However, natural image sequences exhibit a lot of redundancy w.r.t pixel intensity changes. Most of the scene has minimal change in intensity values, and only the moving objects/edges will elicit major changes. This redundancy will be even higher for image sequences captured at a very high temporal resolution For example, a surveillance scene video only has pixel intensity changes at the pixels occupied by persons or moving objects. If one wishes to efficiently capture high frame-rate visual data, the logical option is to only encode the changes in the scene for each pixel and the exact times of these changes. This is exactly what is achieved in asynchronous vision sensors [LPD08b], in which every pixel independently responds to changes in the environment, rather looking for the exact points in time when the changes occur and the

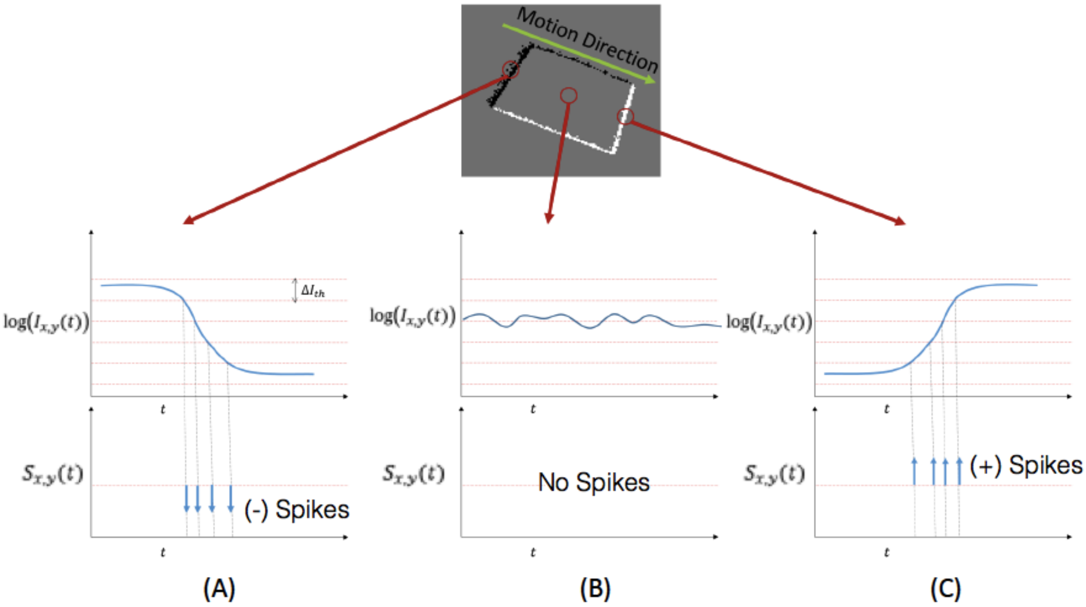


Figure 1.2: Principle of Spike-event generation. (A),(B) and (C) show the temporal progression of different pixels as the object executes motion. The plots in the first row are indicative of the logarithm of the absolute intensity values. Each pixel responds to a fixed threshold, giving a positive spike when intensity increases and vice-versa. A sparse stream of spikes is generated localized to the moving edges of the scene at any time.

nature of the change (Fig. 1.2). Calling each such change as a "visual change event", these event-driven sensors produce an action-potential like pulse (or a 'spike'), only when an visual change event is detected. However, the absolute pixel intensities always changes in time (ambient light fluctuations or slight edge gradients combined with micro-motions of objects) and, therefore, one needs to define a proper threshold of change after which the spiking occurs. Also, if the ambient light is low intensity, then all the changes in the scene will be attenuated correspondingly, and vice-versa. Therefore, instead of a fixed threshold one requires a dynamic threshold which changes proportionally to the light intensity at the pixel. This problem is solved by making the sensors responsive to changes in the logarithm of light intensity with a fixed threshold, which ensures higher intensities require effectively higher thresholds for an "event" to happen.

1.1 A brief review of work in Event-driven Vision

Activity-driven encoding of visual information marks the beginning of a new era of vision algorithms, and are often quoted as bio-inspired due to the "spiking" nature of the pixels, which encode exact event times (Evidence in biology found in [TG08]). Due to their spiking nature, the spiking output from these visual sensors can be readily integrated with spiking neuronal architectures as well ([PPG+13],[Fuk88]) for tackling problems such as object recognition and tracking. Below the algorithms used with event-based visual sensors are highlighted for different problems, with comparisons to conventional algorithms used in Frame-based vision for the same problems.

The following broad categories of problems are elaborated and discussed in detail in the subsequent sections of this chapter.

- Optical Flow estimation
- Object Tracking
- Pattern Recognition (Static)
- Pattern Recognition (Dynamic)
- Feature Detection

Note that the work in this thesis is applicable to only event-based data streams. Therefore, we avoid the discussion of literature relating to cameras such as DAVIS [BBY+14], which supply frame-based snapshots of the scene, in addition to a continuous stream of spike-events. Also, since the contributions in this thesis primarily relates to event-based pattern recognition, and feature extraction, we avoid the discussion of methods in visual odometry related applications, which are mostly 3D vision based. There has been considerable work done in both visual odometry estimation and localization/ego-motion estimation [HWC13, MKLD15, KMGS16, CS14, ZAD17], the discussion of which is beyond the scope of this thesis.

1.1.1 Optical Flow

One of the most important pre-processing step in motion estimation problems involves the estimation of optical flow, and this approach has been extensively studied with frame-based cameras (fundamental works being [LK81, HS81]). Optical flow estimation concerns with modelling the direction and magnitude of apparent movement elicited through moving edges in a visual recording. Naturally, event-driven silicon retinas being responsive only to the changes in the scene, also require the estimation of visual change direction/flow. Most event-based cameras have low spatial resolution (e.g. DVS is 128x128, ATIS is 240x304), and thus utilizing precise spike-times becomes crucial to realizing low error flow detection algorithms. The work by [BCL+14b] is one the more popular approaches to flow estimation. Responding to changes in a scene corroborates to responding to moving edges, and spike-events only sparsely appear through a set of spikes localized to the position of the edge (Fig. 1.2). Modelling the spike-events as a part of a surface as shown in Fig. 1.3, the method in [BCL+14b] involves fitting the events in a local spatio-temporal region to a planar surface, estimating the normal to the plane. In doing so, the algorithm is successfully able to estimate the motion direction of the edges with high temporal-precision and relatively low computational burden. The surface of active events is a good fit to edges which are smooth and can be locally approximated in a linear manner locally. Thus, with edges that have a high curvature, or object corners, the motion estimation the algorithm would not be able to produce a reliable estimate. Also, the assumed constancy of flow in a local spatial region will not hold for more complex motion profiles.

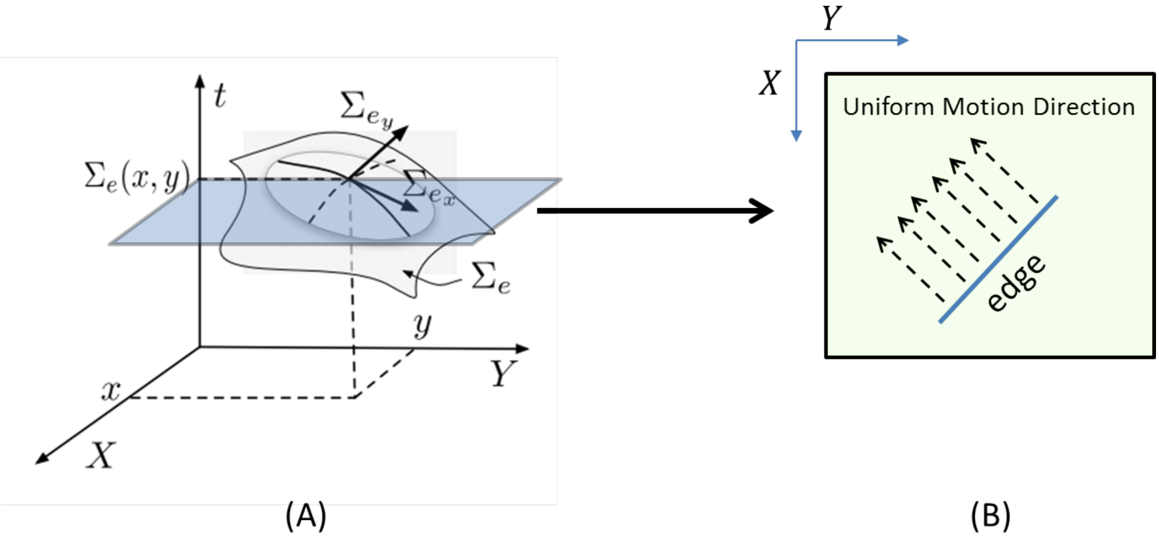


Figure 1.3: Surface of active events. (A) (reprinted, with permission, from [BCL$^+$14b], ©IEEE) shows the surface of 'active' events . The blue plane is cut parallel to the temporal axis, thereby demonstrating the event generating edge(s) at a particular time. (B) shows how the planar approximation of events affects the assumptions of the algorithm. In this case, fitting a plane is equivalent to locally approximating a straight line edge undergoing consistent motion in a certain direction.

This method is in stark contrast to the classic Lucas-Kanade approach [LK81] of determining optic flow for frame-based cameras. Although it operates with the same assumption in the constancy of local-flow fields, the intensity change information is available at all pixels, which is not the case with event-based sensors like DVS, as spikes only occur in pixels accumulating a certain amount of intensity change. A workaround to this is through using the Asynchronous Temporal Image Sensor (ATIS) for data acquisition, which alongside change information also records intensities of the pixels which have changed.

In [OEC14], however, a different approach to modelling the visual flow is demonstrated. They mimick the massively parallel nature of neuronal colonies tuned to different spatial regions, with each having specific neurons tuned to a unique combination of direction and speed of edge stimulus. Also, they exploit the temporal coincidence of phase-shifted spikes from neighbouring regions, following the principles of first order motion perception ([Rei61]). This algorithm uses temporal co-incidence of the variably delayed spikes to detect a flow "event" and therefore makes use of the high-temporal resolution of spikes. Unlike [BCL+14b], this algorithm maintains the "event-driven" approach for flow estimation being consistent with the sampling of the sensor.

1.1.2 Object Tracking

Object tracking involves locating a moving object over time. With frame-based sensors, the natural approach is to have a *template* of the shape and model the possible deformations and transformations on the template image to locate the shape in a complex visual scene. Hence, algorithms iteratively have to estimate object position in the next frame, given their belief of the shape's location in the previous frame.

However, one cannot adopt a similar approach with event-based cameras, as the "template" of the shape is partially dependent on the direction of it's movement. Also, without the concept of a frame, one needs to always keep updating the estimated location and rotation of the shapes as the spike-events continue to happen. In this scenario, the definition of a "template" changes from an image to a point-cloud, which is a set of 3D points in space and time, each containing the tuple (x, y, t). [1] With the incoming spikes being modelled as a point cloud, one can use point cloud matching algorithms to iteratively arrive at a matching with simulated rotation and translation, as in [NPB+12]. To re-create the notion of a frame, the authors introduce an exponential delay (as is the norm) function to find the "active pixels", the ones which constitute the point cloud to be matched at each moment. The algorithm shows real-time operation, but only deals with translational and rotational transformations on the template. Similarly, [LMI+15b] also introduce activity dependent trackers, which retain location information of the shapes even when the shapes stop moving (ergo, spike-events stop being generated from the imager).

The continuous nature of events generated from event-based cameras allows one to have a generative model of continuous geometric transformations including affine distortions of an object/shape [NIP+15]. For simpler objects, such as a ball, more specialized methods which use a particle filter based method to keep track of the location and radius of the ball is proposed in [GB17]. They demonstrated an effective procedure that maintained ball location and size in spite of heavy background noise generated from the handheld motion of the DVS camera.

The aforementioned methods deal with rigid objects with fixed shapes. For more complex deformable shapes, [RVLC+15] breaks down the shape into parts and models each "part" as samples from a gaussian distribution. The parts are connected by hypothetical springs which ensure that the global structure of the trackers are not compromised. This way, the

[1] In fact, the natural definition of spike-events also follows similar convention, with the additional polarity information, making it a four-dimensional vector.

algorithm is also made robust to occlusions as the trackers return to their minimal energy configurations once the occluded part of the shape comes back into the field of view.

A more holistic look at tracking, by using gaussian mixture models to fit entire human silhouettes was proposed in [PNBSG12]. There, the expectation-maximization algorithm was utilized to generate the 2D gaussians, along with their mean and variance parameters. The parameters were updated at regular intervals in time, to enable tracking of the clusters in response to motion.

1.1.3 Pattern Recognition (Static)

Static pattern recognition involves algorithms which infer aspects of the input which are only related to its spatial configuration (for e.g. shape/object recognition, face recognition). For frame-based input, object recognition usually involves the acquisition of a frame, followed by the feature extraction step which ultimately leads to knowledge of object class. With event-driven data, the concept of a frame is compromised and instead the asynchronous stream of spike-inputs means that a classifier will either have to "frame" the spike-events before making a decision, or keep updating its decision as the spike-events arrive in real-time. As such, an immediate question to answer is the methodology involved in *framing* the spike-events, as neural networks only accept fixed-size, structured matrices as the input. In Chapter 2, we show how partial speed-invariance can be attained by simply constraining the number of spikes in effecting a classification decision (using CNNs) [GMOT14a]. As such ([PCZS+13a]) realizes completely asynchronous modular convolutional neural networks, with the output class updating with each incoming spike with minimal delay. [ZYY+13] also includes an AER motion sensor which only performs a classification when it detects sufficient motion activity, thereby decreasing the computational redundancy. [ZYY+13] [OMEC+15] also mimic the bio-inspired HMAX architecture by Gabor filtering each image sub-region for the first layer of features. Such *simple* and *complex* cell based methods can also be found in [ZDC+15, OME+15]. These works show the advantage of adding pre-processing steps which output edge-maps (extraction at multiple scales in [ZDC+15]) and subsequently more complex features for object recognition.

All of the methods mentioned above employ supervised learners. Restricted Boltzmann Machines([Hin02]) use an unsupervised learning strategy to model the data distribution as a product of multiple 'expert' distributions (thereby reducing redundancy). Stacking RBMs on top of each other naturally leads to more abstract concepts useful for recognition (Deep Belief Networks; [HOT06]). [ONL+13] proposes a learner for DBNs with spike inputs, by normalizing the spike rates into probabilities. However, to achieve a good unsupervised learner, MNIST dataset consisting of 120000 images was first converted into a spiking neuronal input representation. This enabled the architecture to respond correctly to actual event input from the Dynamic Vision Sensor. The trained network was made robust to scaling, rotation and translation of the digits by simulating them by modifying the dataset.

In spite of the successful event-driven computing demonstrated in all the above works, there are a few open questions. It remains unclear how unstructured data generated from event-based sensors as point-clouds can be effectively represented as an input for DNN based recognition. Also, signal-to-noise ratio is naturally higher for event-based sensors, as multiple sources of noise, either through false spike-events (circuit artifacts) or through erroneous spike-timings (because of lower *effective* temporal precision of 0.25 ms, as reported in [AMC+15a]). A classifier thus needs to be robust to noisy events, either through training data augmentation (by simulating noise addition), or by filtering for edges at various orientations, as discussed before. Finally, when compared to conventional frame-based cameras, event-based cameras reveal only edge information about the scene, a limitation which affects recognition performance with spike-event data. [2] This, coupled with the low spatial resolution of the sensor, necessitates effective use of temporal precision for better object recognition methods.

1.1.4 Pattern Recognition (Dynamic)

Dynamic pattern recognition, in contrast to its static counterpart, involves additional reasoning about the temporal aspects of input, for inferring category (for e.g. gesture recognition, action recognition). When an event-based camera is static, only moving objects will produce a response of spike-events. Naturally then, responding only to visual motion, at a very high temporal acuity, makes such cameras apt for recognizing actions and gestures. In spite of these advantages, there has been a shortage of literature which specifically targets such problems at scale. In [PCP+16], the temporal spike-event history information from local spatial neighborhoods was used to produce a higher spatial resolution input, termed as demosaicing. After doing so, the upsampled output of higher spatial resolution was then used as an input to a convolutional neural network.

This problem was tackled in more detail and scope in [ATB+17], where a deep spiking convolutional neural network was trained, with regularization methods, for recognizing more complex gestures (11 classes). This was the first large scale dataset for event-based gesture recognition to be released, as it comprised of 29 participants, recorded in a variety of lighting conditions. The system achieved an impressive latency of 100 ms in recognizing gestures, and had a completely event-based implementation on a spiking neural network chip, Trunorth [MAAI+14]. The system achieved 94.5% accuracy in categorizing 11 gestures across 200 ms. This work conclusively showed that in spite of the very low spatial acuity of event-based cameras, classifiers are able to accurately infer gestures in short periods of time, across a

[2] Although, it may be noted that the advent of a new generation sensors which encode both frame-based and event-based information, such as the DAVIS [BBY+14], presents a solution to this information reduction.

wide range of illuminations, owing to the information present in the precise temporal stamps.

Clearly, dynamic pattern recognition with event-based cameras represents a relatively new aspect of research in this field, with scope for further improvement. In chapter 4, we contribute to this domain, and demonstrate a system that outperforms the state-of-the-art by 1% on the IBM gesture recognition dataset.

1.1.5 Feature Detection

Apart from the architectures mentioned in the previous section which work with image features computed with the purpose of object recognition, there have been a few other approaches towards general local feature extraction. [CIB15a] extends the work with plane fitting in [BCL$^+$14b] to detection of corners by modelling them as an intersection of two or more planes. Precise temporal information is used in estimation of the velocities of the edges, to be later combined into a velocity profile for each corner, which can be used in efficient tracking of the corners over time. In this way, the authors realize a spatio-temporal feature descriptor containing a corner point and its direction of movement. As discussed before, [LMI$^+$15b] and [RVLC$^+$15] define features from the statistical point of view, with a gaussian assumption on the local event generating distribution. In this manner it is able to model orientation + thickness of local image regions, while keeping computational costs low.

[Jae01] models the temporal evolution of a spatial region using Echo-State Networks. It poses the feature detection as a classification problem, with each class emulating a different spatio-temporal dynamic. In this manner an unsupervised feature learner, which correctly recognizes features varying in complexity is realized. For foreground-background separation, [BTFA15] uses structured random forests in a supervised setting to label edge regions as belonging to foreground or not. In doing so, it defines features with events based on orientation and temporal-texture[3]. The system shows real-time operation. This demonstrates the importance of the temporal precision of spike-events for segmentation applications as well.

Contrary to the local intensity variations modelled by the well known Scale-Invariant Feature Transform (SIFT) algorithm for frame-based data, it can be seen that one needs to have a completely different approach for event-driven data. Lower spatial resolution of the DVS limits the detail to which one can capture spatial structure, but however high temporal granularity be used to infer image intensity profile at a higher resolution as in [KHB$^+$14]. The algorithm requires a static scene coupled with enough spike information accumulated through repetitive camera motion, thereby not being real-time. Therefore, there exists a need to surpass the intensity estimation step so as to directly compute informative features

[3] Temporal texture at (x,y) was defined as a local array of LxL near (x,y), each element of which contains the difference of the current timestamp and the timestamp of the previous spike-event at that pixel

derived from temporal stamp information of the events.

More recently, [CMBB17] proposes an optical flow based, illumination-invariant feature, which is combined with a B ayesian framework for inferring simple hand gestures. The features are derived from the histograms of the local optical flow vectors, quite similar to HoG for the frame-based, static domain. In [PZY$^+$17], a bag-of-words like feature representation is obtained which is then subsequently used for recognition.

In spite of a plethora of work done on event-based feature detection and representation, there are no methods which propose the *learning* of interesting and important features that are computed from the raw spike-event data itself. Therefore, most of such work compute features which are quite general, and do not exploit the common spatiotemporal spike-event patterns found in narrower domains. For example, a domain such as Traffic, would produce mostly certain kinds of spike-event patterns, the knowledge of which could certainly benefit a feature learner, or a tracker. I demonstrate in Chapter 3, that one can learn more specialized and informative features in narrow domains (such as Traffic), which can be used for recognition and tracking tasks more effectively. Furthermore, very few of these methods propose the learning of spatiotemporal features, mostly because the evaluation is more commonly done on static, spatial-only recognition problems. In Chapter 3, I propose a completely spatiotemporal feature learning framework where features are computed on the raw, spatiotemporal, spike-event data itself.

1.2 Discussion and Conclusions

This section highlights different aspects of information processing with asynchronous data. Working with event-based data requires consideration of multiple factors which would not apply to the frame-based scenario. For instance, how much of the spike-timing precision can be relied upon? Ideally, it is better to keep the computation completely event-based throughout for an event-based system.

1.2.1 More Factors of Variation

Given the high temporal resolution of event-based cameras, the spike-event space is a spatiotemporal 3D space, which is mostly sparse. Therefore, it is a challenge to acquire relevant, denoised information from such sparse spike-event point clouds. One of the fundamental aspects of change-driven cameras, is that the spike-event response is dependent on a number of factors, such as: direction of camera motion, direction of object motion, intensity gradient of the object edge, relative direction of visual edge to motion. As a general rule, spike-events tend to occur most at the object edges which are oriented perpendicular to its motion. This creates more factors of variation for the same stimulus, and pose a greater challenge for pattern recognition problems. Other factors of variation include illumination levels, and also the speed of the objects in motion.

In this thesis, we propose solutions to the above problem, such as for speed and motion direction invariance in Chapter 2 (a), illumination invariance in Chapter 4, viewpoint invariance in Chapter 2 (b) (Global) and Chapter 3 (Local). Interestingly, the solutions we propose adopt from a wide range of methods. However, a more commonly used approach in this thesis is that of incorporating slowness into the extracted feature representations. The concept of slowness was originally proposed in [WS02b], where the authors recognized the need to obtain features which change slowly in response to temporal changes to the stimulus. Slow features are expected to encode more category relevant, high-level information about the input, and therefore are desirable for classification related problems. Slow feature analysis (SFA) has thus been incorporated into most of the work presented in this thesis, whenever the need for extraction of feature representations was present. We find that projection functions obtained from SFA create robust feature representations, when compared with other ways to generate representations.

1.2.2 How useful is temporal information?

The importance of temporal information from different perspectives has been highlighted in [RVOIB15], [AMC$^+$15b] and [OMEC$^+$15]. [RVOIB15] analyze the performance of the tracker when the temporal information is lost, by taking N consecutive spikes and assigning all of them one time-stamp. They observed that with higher values of N, the performance inevitably dropped but it remained at a high level for up until a certain value of N. In spite of single spike level temporal resolution giving the best performance, they settled with a higher value of N (50 in this case). That was because computing with single spikes increases computational time and algorithmic latency as a whole. Therefore, a question to be asked here is: to what extent is the additional temporal information worth the respectively additional processing time?

[AMC$^+$15b] looks at the importance of temporal information by computing the mutual information between the spike-trains and the ground truth stimulus. It repeats the computation for differently sized temporal bins, thereby aiming to demonstrate the importance of temporal information. Their results suggested that a temporal "maxima" exists with respect to the mutual information at just below 1 ms, which is much greater than the $1\mu sec$ resolution of the spike-events. [OMEC$^+$15] shows how recognition accuracies change when the temporal stamps are added a gaussian noise of different variances. They find that the accuracies actually increase slightly when the standard deviation of the noise is about 0.25 ms.

The above results primarily suggest that temporal resolution is only important up to a certain extent, and collapsing spikes very near in time (<1ms and 0.25 ms in those cases) could actually be useful and improve performance. This is partially due to the noisy nature of the threshold in the asynchronous sensors (as highlighted for DVS in [LPD08b]), as the noise renders the temporal order of spikes irrelevant when the spikes are near in time. Another intuitive way to look at this is from the perspective of the events which make up the instantaneous "structure" of an object (As shown in figure). Naturally, one would expect the temporal order information of the spikes that constitute the instantaneous structure to be less and irrelevant, as in this scenario those spikes constitute only one spatial entity.

In this thesis, we demonstrate the use of precise temporal information for the dynamic pattern recognition problems, while refraining from reading too much into the temporal resolution for static pattern recognition problems.

1.2.3 Event-driven computation

As explained in [PCZS$^+$13a], event-driven processing enables computing on every spike-event, thereby highlighting the *coincidence* property of the convolutional networks in that case. They provide an example of a completely event-based convolutional neural network implementation in the same paper. However, as discussed in the previous section, one often does not require processing after each spike-event and in some cases aggregating spikes can be useful (also investigated in [RVOIB15]). Therefore, this brings to light the issue of whether there exists other ways to define "event-driven" processing. For e.g., in [PCZS$^+$13a] we have different convolutional *modules*, iteratively updating the outputs with each new spike event. Instead, better way would be to aggregate a certain number of events which is different for each module before making a new convolutional computation. A similar approach was in [ZYY$^+$13] when the recognition architecture involving a tempotron ([GS06]) 'woke up' when the system detected a burst of events.

With the high sensitivity to intensity change in time combined with gaussian distributed circuit noise in the measurements, the DVS sensors are prone to noisy events. Therefore, computing with each spike could result in false 'features' being detected. Thus attentional tuning to a burst of spikes also inherently acts as a denoising step, involving the computational elements only when sufficient activity is observed. Ideally, we require the feature detection to be at different levels of a hierarchy, motivated from the cortical hierarchy. As such, complex and spread-out features of higher layers should contain more abstract information about the stimulus. Naturally, 'changes' in representation will be faster at lower-levels and slower at higher levels, as highlighted in [WS02a]. Hence a true 'event-driven architecture' should ensure that the event-driven activity of different spatial-scales in the hierarchy should corroborate to different temporal scales as well. In other words, high level modules must update slowly and less often in time, giving the perception of lesser 'events' detected at that level and vice versa. As an example in object recognition, a low-level module will regard local translation as an event, whereas for a high-level module a change in object category will be regarded as an event.

The methods in this thesis have not been described from an event-based perspective, but an event-based extension to the methods can be implemented in most cases. We therefore refrain from the discussion of neuromorphic hardware based (e.g. spiking neural network) implementation of the proposed methods in the subsequent chapters. However, in the appropriate cases where an event-based hardware implementation is possible, relevant discussions are made.

1.2.4 Thesis organization

The thesis is organized as follows. Chapter 2 describes object recognition approaches, to adjust to factors of variation in the event-based domain. Chapter 3 proposes a novel spatiotemporal feature learning and tracking approach. Chapter 4 extends the feature learning framework for dynamic pattern recognition problems such as action and gesture recognition. Chapter 5 presents the broad conclusions and lessons learned from the various works documented in this thesis, along with an eye on possible future directions.

Chapter 2

Object recognition with the Dynamic Vision Sensor

In this chapter we present two different works, both targeted towards object recognition albeit for different applications. Both the works are summarized below:

- We describe a real-time bio-inspired system for object tracking and identification which combines an event-based vision sensor with a convolutional neural network running on FPGA for recognition. The system is demonstrated for two tasks. The first is proof of concept for a remote monitoring application in which the system tracks and distinguishes between cars, bikes, and pedestrians on a road. The second task targets application to grasp planning for an upper limb prosthesis and involves detecting and identifying household objects, as well as determining their orientation relative to the camera. The second task is used to quantify performance of the system, which can discriminate between 8 different objects in 2.25 ms with accuracy of 99.10% and is able to determine object orientation with ±4.5% accuracy in an additional 2.28 ms with accuracy of 97.76%.
- We present a system capable of recognizing objects invariant to their pose. Building on the effective principles of event ROI acquisition found in the previous work, we demonstrate the importance of finding features invariant to transformations. In particular, features which are informative and change smoothly and slowly over object pose changes are preferred. A novel slow-ELM architecture is proposed which combines the effectiveness of Extreme Learning Machines and Slow Feature Analysis. We quantify the 2D pose range of viewing necessary for such a system to generate reliable object estimates. The system is able to classify 10^4 times in one second, giving 1% classification error for 8 sample objects with views accumulated over 90 degrees of 2D pose.

2.1 Object Recognition and Orientation Estimation

Much of the following text has been adapted from ©2016, IEEE BioCAS [GMOT14b].

2.1.1 Introduction

Visual tracking and recognition of moving objects in cluttered scenes is typically regarded as a computationally intensive task for artificial vision systems, yet biological vision systems perform the task with ease. Modern asynchronous time-based vision sensors, which operate more similarly to the human retina, provide a robust and efficient representation of dynamic visual scenes. As shown in the previous chapter, numerous advantages of such sensors, including asynchronous operation, temporal precision and good response within a dynamic range, all contribute to an effective alternative for such tasks. Only detecting changes means that the pixels are blind to static distracters in the background, while the high temporal resolution with which changes are detected greatly simplifies tracking.

Therefore, in using such data for object recognition tasks, one creates a sparse binary image, with the non-zero locations showing the pixels at which changes were recorded. This also implies a significant reduction of computational requirements and also the time taken to train a Convolutional Neural Network (CNN) classifier [PCZS+13a]. CNNs are recognised as a powerful, bio-inspired tool for visual classification, providing high accuracy for tasks such as digit classification [LGTB97] [CMGS11]. They have been gaining popularity recently as interest has been increasing in using deep learning to handle large datasets.

Previously, Perez Carrasco *et al.* [PCZS+13a] have shown how frame-based CNNs can be used to process data from event-based vision sensors to achieve high recognition accuracy, and how a learnt frame-based CNN architecture can be mapped to a spiking neural network to tackle high speed tasks (stimulus present for less than 20ms) in real time.

In this work we combine an asynchronous event-based sensor, known as the Asynchronous Time-based Image Sensor (ATIS) [PMW10], with a CNN implemented with the Neuflow architecture [FMA+10], achieving real-time tracking and identification of objects. Such a system can find many applications, two of which are addressed here. Notably, the initial temporal binning method described later is along similar lines as [PCZS+13a], but further analysis is largely different. Moreover, we do not use gabor filtering to extract features for orientation estimation.

The first is for monitoring, which can occur at remote locations, such as a border post, or for monitoring a traffic intersection. One can imagine such a system integrated into the infrastructure of a *smart* traffic intersection, giving it the capability to detect and locate pedestrians, bikes, and cars within the intersection and communicate this information to new vehicles arriving on the scene, or to control timing of traffic signals. The ATIS has already found application in highway traffic monitoring [BBD+07] for car counting and speed estimation.

A second, more near term application, is to aid grasp planning for an upper limb prosthesis. The market for upper limb prostheses has grown rapidly in the last decade as medical treatment improves and a larger percentage of traumatic injury patients now survive their injuries. Along with this growing market has come a concentrated and well funded effort to improve the state of the art in upper limb prostheses, which has resulted in impressive upper limb prostheses, capable of matching the human arm in terms of size, weight, strength, and dexterity. However, dextrous control is impeded by low communication bandwidth between the patient and prosthesis, and remains an unsolved problem limiting the capability provided to patients.

In this work we propose an approach which incorporates a dynamic visual sensor into the prosthesis to the object to be grasped and its orientation to aid in grasp planning. We describe our system in Section 2.1.2, before describing the system testing in Section 2.1.3. We then discuss results in Section 2.1.4.

2.1.2 Methods

The system consists of an asynchronous event-based vision sensor [PMW10] for raw visual data acquisition, and Neuflow [FMA+10] running on a Virtex 6 FPGA for object recognition and orientation estimation. Neuflow is designed to work on static images, but the vision sensor outputs events which can occur at any time and pixel location. To artificially create a static image for Neuflow to process, we need to define a spatiotemporal Region Of Interest (ROI) containing events which will be converted into a static image for further processing.

We begin by preprocessing the events using a simple noise filter [DL07], before determining the temporal (Section 2.1.2) and spatial (Section 2.1.2) boundaries of the spatiotemporal ROI to be considered. Finally, once the ROI has been defined, we need to determine how to convert the spikes contained therein into a static image (Section 2.1.2).

Temporal ROI

We explored two methods of defining the temporal ROI. The first method uses a constant time window, in other words just looking back a fixed time period from the present time. The second method uses a dynamic time window, adjusted such that a fixed number of events are contained within the ROI. This number is fixed to a certain constant before the entire process of classification.

The sensor generates events which correspond to temporal contrast, which is typically generated by the combination of spatial contrast and motion, as defined by the image constancy constraint below.

$$\frac{dI(u,v,t)}{dt} = -\frac{dI(u,v,t)}{du}\frac{du}{dt} - \frac{dI(u,v,t)}{dv}\frac{dv}{dt} \tag{2.1}$$

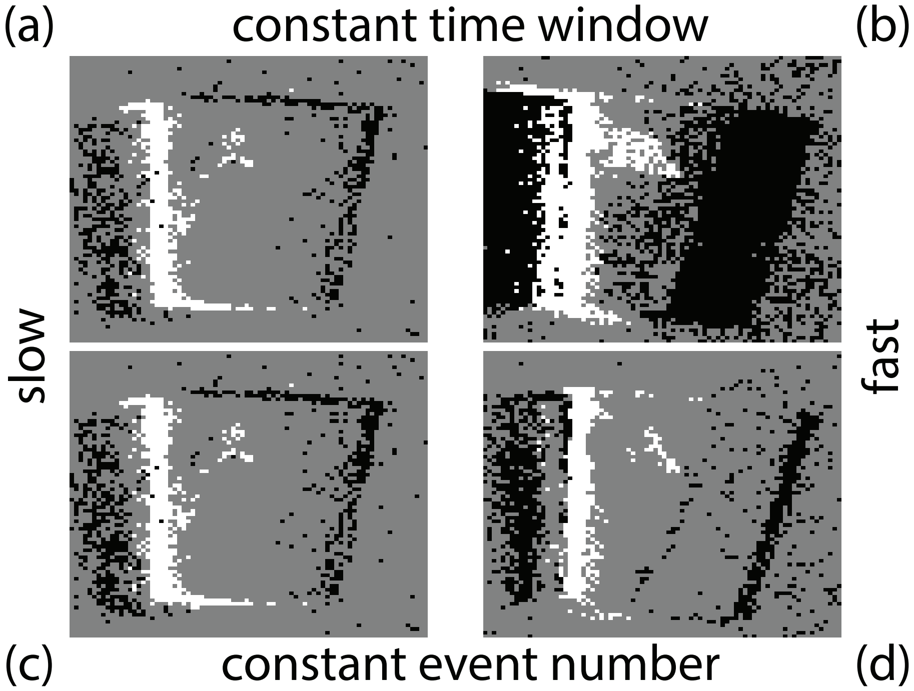


Figure 2.1: Reducing speed dependence using a dynamic time window. The top row show views of the same object moving slow (a) and fast (b) extracted using a constant time window method (each image contains 33ms of data), while the bottom row shows views of the object moving slow (c) and fast (d) at the exact same points in time, extracted using the constant event number method (each view contains 1500 events). The constant event number method provides a more consistent view of the object as speed changes.

where $I(u, v, t)$ is intensity on the image plane, and u and v are horizontal and vertical pixel coordinates respectively.

The faster an object is moving, the more pixels it will activate within a fixed time period. If a constant time window is used, the appearance of the object will be heavily dependent on the speed at which it is travelling, but by using a dynamic time window and keeping the number of events constant, we can largely remove the effect speed has on the object appearance. The high speed of the change detection circuitry in the ATIS also means that it can capture the motion of fast moving objects making it less prone to blurring, which is a problem for traditional frame-based cameras.

Fig. 2.1 provides a visual comparison of the constant time and constant event number methods for defining the temporal ROI. In the example, the constant event number method provides a more consistent image of the object.

Spatial ROI

A rectangular spatial ROI is used to contain objects to be classified. The ROI is defined by a location (the centre of the object) and a size (the size of the object). The locations of objects within the scene are determined using a simple activity tracker which has been previously published [DL07].

Two methods were investigated for determining the extent of the ROI in each of the four directions (up, down, left, right) from the object centre. The first method uses a fixed size bounding box of size 60×60 pixels, while the second method uses a dynamic bounding box, with the extent in each direction chosen such that 95% of the events used by the tracker are contained by the ROI. The ROI is then resized to 60×60 pixels for classification by Neuflow, which improves scale invariance.

Converting Events to an Image

Once the spatiotemporal ROI has been defined, the spikes contained therein must be converted into a static image. Three obvious methods exist for generating the image. Note that in this study, each data sample extracted from the scene for both training and testing has only object present in the same. The method

The first method counts the number of events for each pixel and assigns that sum as the pixel value. This method results in a non-negative value for each pixel. The second method assigns to each pixel the polarity of the most recent event, or a value of 0 if no events are received from that pixel in within the spatiotemporal ROI. This method restricts pixel values to {-1, 0, +1}. The third method assigns a value of 1 to any pixel which had at least one event in the ROI, and 0 for all other pixels.

Once one of the methods above has been used to create a static image, the image is resized to 60×60 pixels using nearest neighbour interpolation before being sent for classification by Neuflow.

2.1.3 Testing

The system was setup as a live demonstration for tracking and classifying vehicles and pedestrians passing by on a road (see Fig. 2.2) and subjectively appeared to provide accurate results. To objectively analyse the system and quantify its performance, four further tests were performed. These tests relate to the prosthesis application and were designed to investigate the sensitivity of the system accuracy to object speed, motion direction, and orientation, as well as the system's ability to discriminate between the same object presented at different orientations. All tests were performed under ambient lighting.

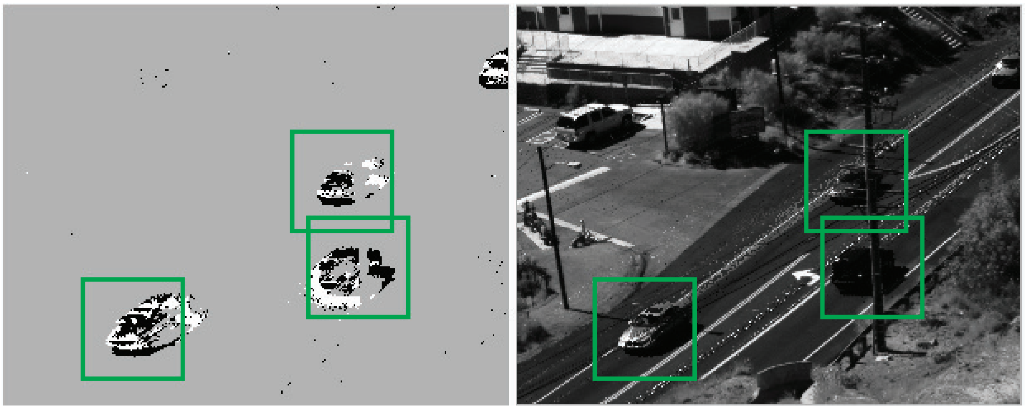


Figure 2.2: Screenshot from live operation while tracking and identifying moving targets. Boxes indicate tracked objects, while colour indicates the object class (green indicates a car). The ATIS performs both change detection (left) and absolute grayscale measurements (right), but only change detection is used for classification, the grayscale image is shown just for visualization. Static objects are not detected (car in the top left), and objects overlapping the scene border are ignored (top right). All three objects moving within the scene are accurately tracked and identified, even though two are partially obscured by a lamppost.

Table 2.1: Speed Invariance Accuracies

OBJECT	FIXED EVENTS	FIXED TIME
Bottle	100	100
Box	100	96.95
Tennis Ball	99.4	99.2
Overall	99.8	98.71

Raw data was collected by placing objects on a moveable platform with the camera observing from a distance of 80cm. The logged data was split into test and training sets using MATLAB and a CNN was trained and tested using the Neuflow architecture of the machine learning library of Lua. These CNNs were implemented in real-time on the Xilinx ML605 platform. The variation of training data with accuracy and classification time with number of objects (classes) has been shown for reference in Fig. 3.

Speed Invariance

To test how object speed affects recognition accuracy, data was collected from three different objects (Ball, Box and Bottle) while holding them at a constant orientation and moving them at speeds ranging from 0 to 420 pixels per second in the horizontal direction. 2150 examples of each object were extracted, with 1700 used for training and 450 used for testing. The test was repeated 5 times using each of the dynamic and constant spatial ROI methods. The constant event number method was used both times to determine the temporal ROI. The results are shown in Table I.

Motion Direction Invariance

To test the dependence of the system on the direction of motion, data was collected for 8 different objects by moving the objects in a circular manner in a plane roughly parallel to the image plane. The dynamic spatial ROI and fixed event number temporal ROI were used for this test. A background class was also created by walking around the lab with the camera hand-held and extracting random regions from the video acquired in this manner.

2000 examples of each object were extracted, with 1500 used for training and 500 used for testing. The test was repeated 8 times.

Orientation Invariance

The next test was designed to investigate the effect of object orientation on classification accuracy. For this test 8 objects and background were used, with each object appearing at random orientations. The objects were captured by shaking them slowly to generate events.

The dynamic spatial ROI and fixed event number temporal ROI was used. 2500 examples of each object were extracted, with 2000 used for training and 500 used for testing. The test was repeated 5 times. The results are shown in Table II.

Orientation Discrimination

Estimating an object's orientation was treated as a two step problem. In the first step the system determines which object is being viewed, using the classifier trained for orientation invariance (Section 2.1.3). In the second step, another classifier is loaded which has been trained only on different views (orientations) of the detected object. This second classifier then outputs the orientation of the object being viewed.

To test this approach, 8 different classifiers were trained, one for each of the objects used in testing. Each classifier was trained on views of the object ranging from -90 to 90 degrees in steps of 18 degrees, with each view being treated as a different class. The test used 200 training examples and 100 test examples for each output class (orientation), and the test was repeated 2 times. The dynamic spatial ROI and fixed event number temporal ROI was used. The results are shown in Table II. There is no orientation discrimination accuracy provided for the background class as the notion of background we use is rather a cluttered combination

Table 2.2: Orientation Invariance and Orientation Discrimination Accuracies

OBJECT	ORIENTATION INVARIANCE	ORIENTATION DISCRIMINATION
Table Tennis Bat	99.65	98.30
Purse	98.25	98.95
Mobile 2	99.70	98.35
Mobile 1	98.63	96.65
Pen	99.98	99.65
JoyStick	99.48	99.90
Bottle	99.63	92.50
Background	97.03	
Overall	99.10	97.76

of different objects separate from the ones we train our classifier with. Therefore, orientation is neither well defined nor useful for background.

2.1.4 Discussion

For an application such as developing an embedded system to visually assist an upper limb prosthesis in object grasping, one must ensure that its performance is robust to variations in appearance which can result from the relative position, orientation, and motion between the sensor and object.

Invariance to translation parallel to the image plane is the easiest to ensure and is obtained by tracking the object. Translation perpendicular to the image plane (along the z-axis) results in a change in the apparent scale of the object and we have presented the scale invariance in Section II.B. Invariance to rotation about the z-axis has been shown in Section III.C. Minor rotation about the x- and y-axes was encountered during recording of testing and training data, but we assume the user will approach the object from a consistent direction.

The necessity of a background class for the final recognition task arises as a result of there being multiple motion clusters recorded by the ATIS when the system is moved towards any object. These motion clusters will either pertain to an object of interest or to background distractors. The background-trained classifier filters out these distractors. The lowest recognition accuracies are obtained for the background class because it exhibits the highest intra-class variance.

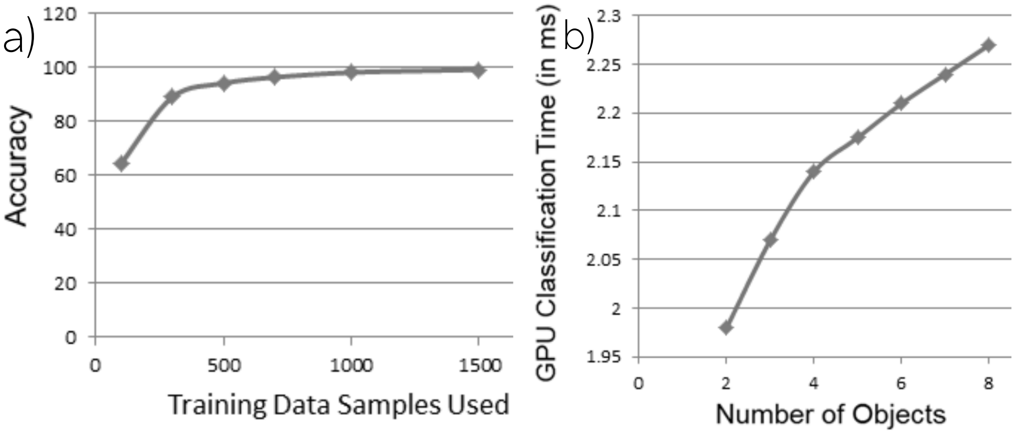


Figure 2.3: (a) A plot of Test Data Accuracy versus the number of training samples used. The accuracy is seen asymptotically reaching the value of 99.1% for large training datasets. (b) A plot of Classification time of one sample (in milliseconds) versus the number of Objects(Output nodes) for the Network. As expected the classification time increases with the network size, but not completely linearly, as the GPU implementation is partially parallel in nature.

To improve classification accuracies, we worked with distorted datasets as in [SSP03b], where the training data had been elastically deformed and the resulting classifier had higher test accuracies. We experimented with different time windows sizes as a measure of distortion to the training set, and as an example we found that a 60 fps based classifier had 1.25% higher accuracy on the 30 fps based testing data and the 30 fps based classifier itself. Further, the constant-time window classifier trained had 2% lower accuracy on constant-event number testing data than the constant-event number classifier. Therefore it can be seen that distortion mechanisms can provide good generalization for insufficient datasets or datasets having less intra-class variance, and having a large number of objects will require doing so. The training process takes around 4 hours for the 9-class convolutional network. The current implementation takes one object at a time for classification. Apart from speeding up computation, the FPGA implementation has been done to move towards a stand- alone real time system that could directly communicate with the camera input in the future.

Note that all the accuracies reported here are with a binary assignment of pixel value for the image which is presented to the CNN, based on the presence/absence of a spike-event within the corresponding spatiotemporal ROI (as described back in section 2.1.2. We also tested another way of accumulating the image presented to the CNN, where instead of binarizing the pixel value, we report the sum-total of the number of spike-events within each pixel ROI. Interestingly, this approach performed significantly worse (by 2%) than the former approach, especially for motion direction invariance. This naturally implies that binarizing the pixel values based on the presence of any spike-event, is somehow contributing to a motion direction invariance in the trained classifiers. This is easy to explain, as different motion directions tend to emphasize different edge parts of an object, where the most spike-events usually generate from the edges most perpendicular to the chosen direction of motion. As such, there would be more spike-event count registered at those edges. When binarizing the image, this motion-direction triggered difference in spike-event density, is less, as the binarizing of the pixel-values ensure that a pixel value is registered as one irrespective of the underlying spike-count.

2.1.5 Conclusion

In this study, we have presented a system for real time object recognition and orientation estimation using ATIS with a CNN. The system is capable of recognizing objects with 99.10% accuracy and discriminating orientation with accuracy 97.7%. A system is intended for real time grasp planning whilst performing robust object recognition and orientation estimation.

2.2 Pose-invariant Object Recognition with the DVS

Much of the following text has been adapted from ©Springer, ICANN, [pos16].

2.2.1 Introduction

Conventional frame-based sensors capture intensity values of the whole pixel array at fixed time intervals. In contrast, asynchronous imagers remove the notion of a frame by essentially being responsive to intensity changes at an almost continual time-scale. With their sparse, non-redundant input data stream only capturing salient moving edges, computational burden is reduced by only computing with the active events at any time as in [14]. For object recognition this points to faster and co-incidental processing as highlighted in [13], where classification happens more spontaneously to the onset of spikes from moving objects. The high temporal resolution also allows for accurate pose-estimation in real-time when the underlying edge-structure of the object is known as shown in [3].

This work proposes a method for pose-invariant object recognition with event-based visual data. Like in [2] each object class is subdivided into multiple pose-specific classes, similar to [2] where separate eigen-faces are found for each pose. Slow feature analysis (SFA [5],[7]) .SFA has been successfully applied to learning pose-invariant features in [6]. We extend the standard random projection based Extreme Learning Machine architecture through choosing informative and slowly changing non-linear projections using the principles of SFA. We propose a Slow-ELM architecture which learns from data having gradual change in 2D pose of objects.

In the earlier work, we primarily wished to incorporate speed, motion-direction and 2D orientation invariance to a CNN trained for event-based object recognition. In this work, we focus on a trickier one: pose-invariance. In the previous work, we incorporate those invariances by training data augmentation. Here we use a similar augmentation scheme, by recording object views across all of its poses, by placing the object on a rotating platform. But additionally, we impose a smoothly changing representation constraint to the hidden layer representation in a neural network, by using SFA.

As the DVS only responds to changes, one can only expect spikes generated by the object edges when either the object or the camera is in motion. Thereby, the invariance of our classifier performance to speed is demonstrated, along with quantifying the amount of multi-pose-view information needed to make reliable class estimates. Our Slow-ELM learner shows a considerable improvement in classification performance compared to the standard ELM, achieving 1% error with 2D pose views spanning 90 degrees. Compared to the principal components based projections, slow projections are found to give better recognition estimates. shows faster convergence to small error rates with accumulated events from successive views varying in 2D pose.

Such a method can aid any system which requires reliable object identity information in real-time. Given that the object's visual stamp could originate from any pose, one requires a learner unbiased towards a certain pose and having the ability to integrate information from minimal multiple-views so as to generate reliable estimates of object identity with less delay. Asynchronous imagers are the natural choice as they will only respond when the camera proceeds to a different view-point, thereby minimizing computational load.

2.2.2 Background

Each pixel of the DVS generates a spike-event only when it records a relative change in intensity above a threshold from its previous time of spike. This almost happens as an instantaneous reaction to the change as the temporal resolution of the spikes is of the order of a sec. Each spike-event is of the form $e_i = (x_i, y_i, p_i, t_i)$, where (x_i, y_i) and t_i indicate the location and the precise time of the change and p_i indicates whether the intensity increased $(+1)$ or decreased (-1) at that pixel. Such spontaneity ensures faster information transfer between the visual change event and the response, but also contributes to more noisy spikes. The noise is a result of both circuit noise and the noisy threshold of the pixels. Thus any features extracted with the sparse spike-events have to be robust to the outlier spikes and variations in the structure of the object induced by the partially fluctuating nature of the threshold for spiking at every pixel as described in [1]. This calls for a high-level feature representation robust to noise and transformations of the object, which is realized by:

- Recording event-data for objects smoothly changing in structure or pose

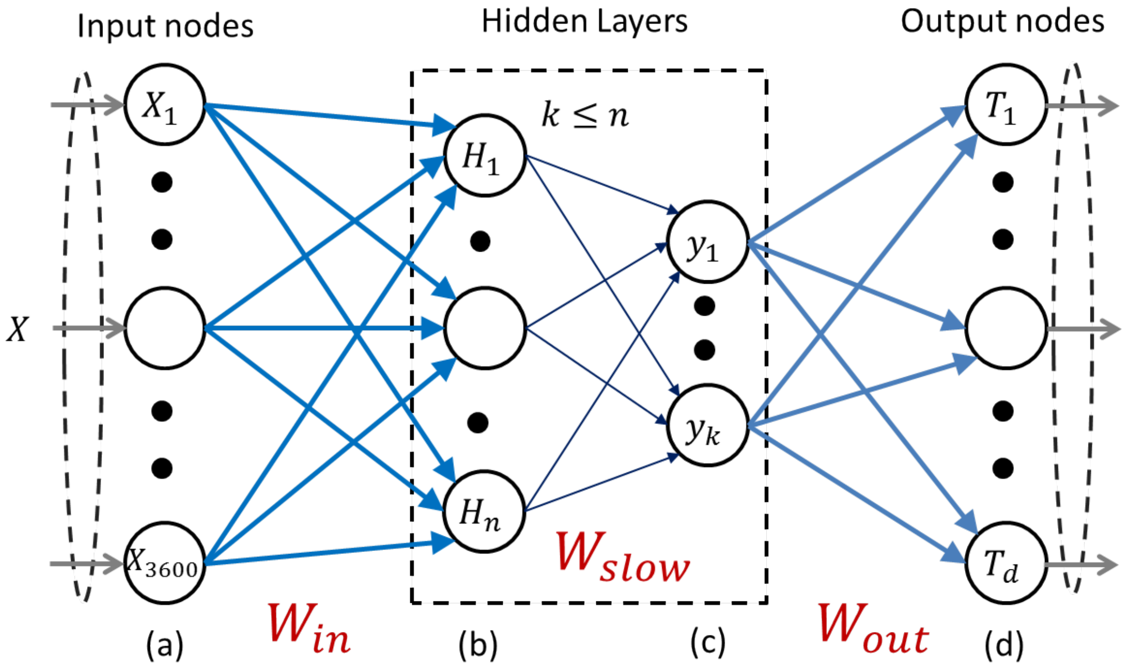


Figure 2.4: The Slow-ELM architecture. The transformation represented by the matrix W_{slow} only preserves the slowly changing projections of H. W_{in} correspond to the Gaussian randomized weights as in conventional ELM. W_{out} is learnt between the projected signal and the output vectors.

- Finding slowly changing and uncorrelated non-linear features using SFA

2.2.3 Methods

The algorithm consists of four steps: Spatiotemporal ROI estimation; slow-ELM; pose-specific labelling; multi-view object class estimation.

Spatiotemporal ROI Estimation

For collecting the spike-events to generate a 'frame', we employ the constant event approach used in our previous work in [4] which maintains event structure w.r.t change of speed. Before the data is input to the classifier, a rectangular spatial ROI is obtained by considering a certain fraction of the events on each side (up, down, left, and right) of the centroid of the extracted events in each 'frame'. Finally, we maintain a memory of the ROI in the previous frame to generate the one with the next set of events. This involves only processing the events near or inside the previous ROI for the estimation of the new ROI. The image formed by the pixels within the ROI was then resized to a square image of a fixed size before passing

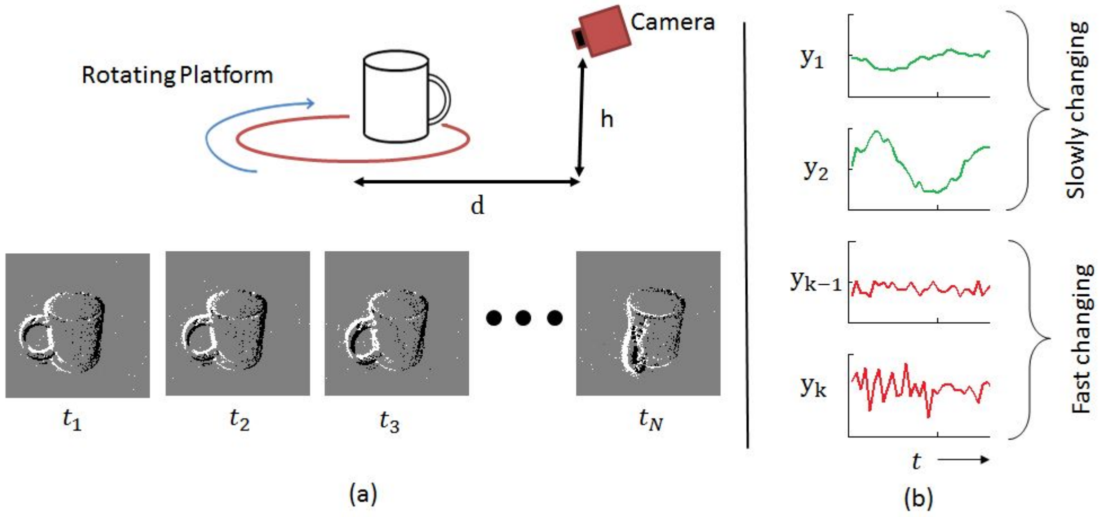


Figure 2.5: (a) shows the experimental setup along with sample framed event data to (I_1 to I_N) for the rotating cup object. The recording is repeated for 3 values of distance d and two different heights h of the camera, each time with 3 motor speeds of rotation ω. (b) shows the contrast between the slowly changing and fast changing projections in response to the rotating object.

onto the Slow-ELM.

Slow-ELM

We have training samples $\{(x_i, y_i)\}_{i=1}^{N}$, where each X_i is obtained from the image inside the ROI. Every dimension of X is scaled to the range [-1,1] before passing onto the ELM. The ELM is initialized with the entries of the input layer weights in being initialized randomnly according to the normal distribution $N(0, 1)$. The n hidden neuron values in H_i are computed via adding a sigmoidal non-linearity f onto the random projections as follows

$$H_i = f(W_{in}^T X_i) \tag{2.2}$$

Now the SFA algorithm elaborated in [5] is applied, which finds uncorrelated linear projections of as expressed by the projection matrix :

$$Y_i = W_{slow}^T H_i \tag{2.3}$$

The elements of W_{slow} are found according to the SFA optimization method. In particular SFA looks for projections which minimize:

$$\langle (\Delta y_j)^2 \rangle \tag{2.4}$$

Under the constraints:

$$\langle y_j \rangle = 0 \tag{2.5}$$

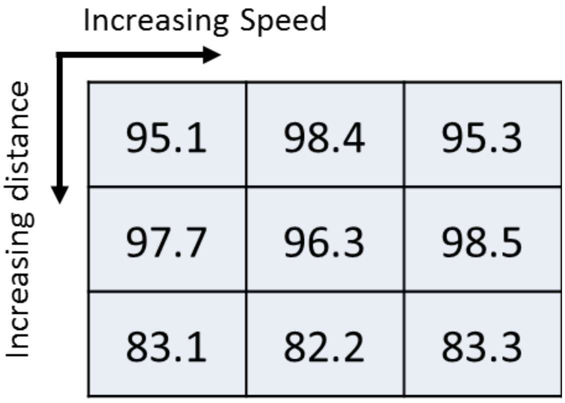
$$\langle y_j^2 \rangle = 1 \tag{2.6}$$

$$\langle y_i y_j \rangle = 0, i \neq j \tag{2.7}$$

$\langle y \rangle_t$ denotes the expectation of y over time, in our case being the average value of the projection across all classes. The unit variance condition ensures normalization of the projections. Similarly, $\langle (\Delta y_j)^2 \rangle$ is the squared energy of the difference of a projection over two consecutive instances (difference energy). In our experiments, two consecutive instances of frames only differ in the 2d-pose of the object. As noted in [5], these slow features can be obtained simply by sphering the data followed by finding the lowest eigenvalues of the difference data Δy. Thus, we seek to find projections that are slowly varying with change of pose, but overall give enough information to discriminate between classes. As the hidden neuron vector H is n-dimensional, W_{slow} will be an (nxn) matrix with each column being a projection found through SFA. Since SFA returns the projections in order of decreasing difference energies we keep only the first k columns of to generate W_{slow}. Different values of k are experimented with in our experiments.

Pose-specific Labelling

Every object data captured is categorized differently according to the 2D pose range it belongs in as we record from all viewpoints across 360 degrees (Fig. 3 (a)). In particular, we choose 8 partitions of the pose (0-45), (45-90), (315-360). So with N objects, we have 8N classes. The algorithm up to this point remains unsupervised as the only learning happens for finding the entries of W_{slow}. As shown in Fig. 1 the final layer is learnt through the regularized least squares algorithm shown in [15]. For each training sample X_i, we extract the slow projections Y_i through the aforementioned steps . Now the supervised RLS algorithm estimates the linear mapping between Y_i and t_i , in W_out as used in [8]:


Increasing Speed
Increasing distance

95.1	98.4	95.3
97.7	96.3	98.5
83.1	82.2	83.3

Figure 2.6: Recognition accuracy across 3 different speeds and distances.

$$W_{out} = (\frac{I}{C} + Y^T Y)^{-1} Y^T T \tag{2.8}$$

Here $Y = [Y_1, Y_2, ..., Y_N]$ and $T = [t_1, t_2, ..., t_N]^T$. The parameter C controls the tradeoff between the regularization and the error term. Higher the value of C, lesser the smoothness constraint on the weights and therefore higher the chance of over-fitting the data. Given the input to the final layer Y_i we then finally end up with the output vector:

$$t_i = W_{out}^T Y_i \tag{2.9}$$

The class estimate is then the object for which one of its pose-specific class has the maximum value across all 8N classes in t.

Multi-view object class estimation

This discusses the problem of object class estimation when multiple input data $(X_1, X_2, ..., X_N)$ derived from many view-points of a single object is presented to the classifier. This we do simply by looking at the class receiving the maximum number of votes, where the i^{th} vote cast is to the object category of X_i inferred by the Slow-ELM output. Since we record the event data with the object smoothly changing in pose, $(X_1, X_2, ..., X_N)$ are the successive instances of the event-structure as the object rotates.

2.2.4 Experimental Setup

As the DVS only responds to changes in the scene, the experimental setup consisted of a rotating platform on which an object was placed. Such a setup however makes the pixels near the centre of rotation generate lesser spike-events than the pixels near the edge. To avoid this motion intensity bias, the objects were placed near the edge of the platform (as shown in Fig. 3 (a)). For each object, the event data was captured as the platform was rotated over 6 radians, thus uniformly covering the range of 2d-pose. The experiment was repeated for two elevations of the camera similar to what was done in [16], and across 3 different distances from the platform centre. For each configuration, object data was recorded for 3 different angular velocities of the platform, with a total of 8 objects. The objects chosen were: camera, cup, computer mouse, pen, mobile phone, scissors, spectacle and bottle. The output weight matrix essentially learns a 64-class classification problem.

2.2.5 Testing and Results

Recordings varied in motor speeds of the circular rotating platform (3 values), distance to the platform centre (3 values) and height of the camera (2 values), hence giving a total of 18 recordings. Out of the 18 recordings, 9 were used for testing and the other 9 for training. Not every object had the same number of data, as they generated spikes at different event rates. Therefore for an unbiased estimate of performance, testing data for the classes having lesser examples were duplicated randomly to ensure equal instances of each class. After duplication, each class had approximately samples. The image extracted from the ROI is resized to a 60x60 image, which is then treated as a 3600 dimensional vector to be input to the ELM. W_{in} gives 3000 hidden nodes, and different number of projections in y are experimented with to show the progression of accuracy with varying input dimensionality to the final layer of learning.

Performance with varying speed and distance

Shown in Fig. 3 is the effect of changing speeds and distance of the platform on the accuracy. The accuracy remains high for distances $d =$ 30 and 45 cm, but drops abruptly for $d =$60 cm. This indicates that the classes become less separable quickly as the distance to the object is increased beyond a limit. The effect with varying speed of the motor of the platform however is not felt which indicates the invariance to speed changes.

Comparing slow-ELM with other selection criteria

Fig. 4. demonstrates how Slow-ELM compares in performance with traditional ELM and other variants, as a function of the number of projections used for learning . In particular,

- slow-ELM: This is the same as finding through the SFA criteria. The projections chosen here is in increasing order of difference energy.

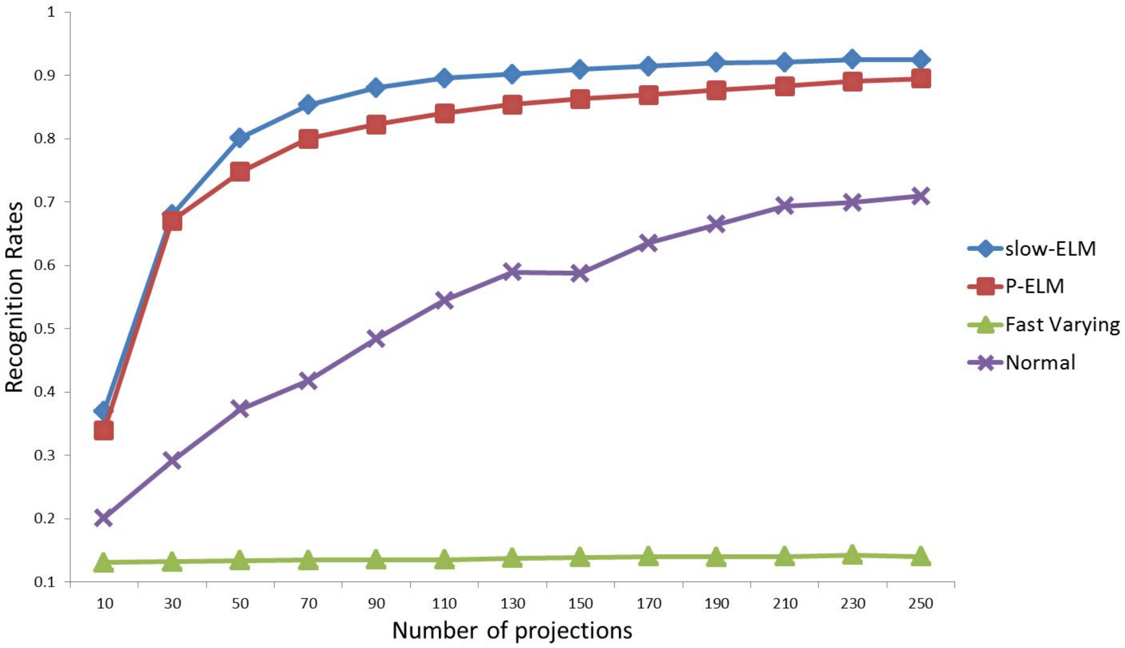


Figure 2.7: Recognition Accuracy for varying number of the k selected projections used for training the output weights in W_{out}, shown for the different selection criteria mentioned in section V(B).

- P-ELM: We choose the maximally informative projections of through Principal Components Analysis.
- ELM: This is the conventional ELM where the hidden neuron values in H are directly used to learn W_{out}. The number of projections simply is the number of hidden neurons.
- Fast varying: Here we sort the projections of W_{slow} in reverse order to slow-ELM, giving preference to projections having higher difference energies and thus exhibiting higher fluctuations with time.

The figure clearly demonstrates that SFA based projections give the best recognition accuracies ($\approx 93\%$). In contrast, the FAST features perform very near to chance itself (14%, chance is 100/64=15%). This suggests that fast, fluctuating features are noisy and do not provide abstract category information essential for classification.

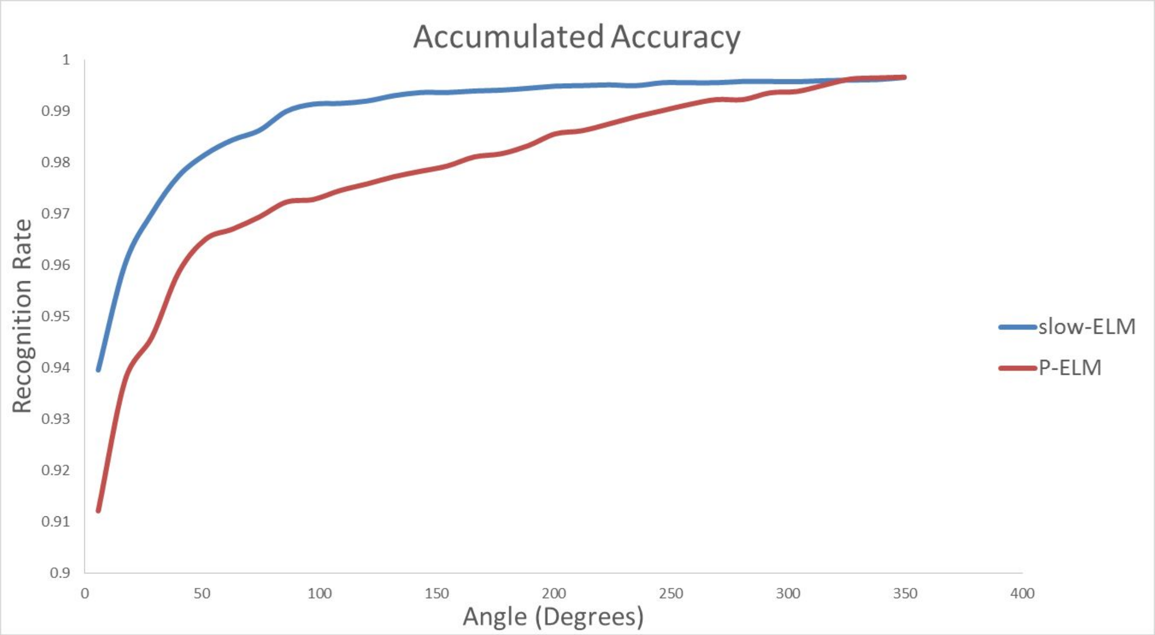


Figure 2.8: Recognition accuracy with slow-ELM and ELM-PCA on aggregated data from successive viewpoints spanning different range of 2D-pose

Multi-pose view object recognition

Here the method described in Section III.C is used to arrive at class estimates with event-data accumulated across changing pose as the objects rotate. Precisely, we quantify the recognition accuracy when event data spread out in different range of 2d pose is available. This is averaged across all starting poses of accumulation of event data of an object. Fig. 5. compares the recognition accuracy for both SFA and PCA based projections. It can be seen that SFA quickly reaches a low error rate (1%) in classification with only 90 degrees of pose information whereas PCA requires 280 degrees to achieve the same error.

2.2.6 Discussion

Unlike conventional SFA which uses a quadratic expansion of the input signal to a higher dimension, we employ random projections followed by a sigmoidal operation as the source of non-linearity. This allows the algorithm to directly operate on a high-dimensional feature space, choosing linear combinations of features which are most obedient to the slowness principle. The slowness criterion essentially eliminates all ?noisy? random projections, thereby accelerating learning and generalization. In our previous work the performance of a classifier working with asynchronous event-data was seen to be very sensitive to the way one handled the spatio-temporal ROI of events to be chosen as an input to the classifier. Building on the principles of constant event sampling, which ensured lesser variance in object structure due to partial speed invariance, slow-ELM is a step further as we move towards stable, less variant high level representations. These works in general highlight the importance of:

- Sampling the event-data in such as way as to suppress structural variability of the object
- Working with features which only are informative but slowly varying in nature

This architecture can be mapped onto it's asynchronous, spiking neural network equivalent by simply scaling the weights in a certain way as shown in [13]. The asynchronous equivalent network incrementally changes the spike potentials of each hidden neuron with each incoming spike. Using slowly changing features is essential to this mapping, as otherwise the noisy spikes will have a greater fluctuating effect on the neuronal spike-potentials. It will also ensure a smoother transition to the correct object estimate and overall a more stable representation in time.

2.2.7 Conclusion

We have presented a system capable of recognizing objects from a real-time feed of spike-events and capable of generating accurate class estimates by combining information from successive views varying in object pose. Apart from the low computation time which allows upto 10^4 classifications per second, the training time is also considerably lesser than the state-of-the-art Convolutional Neural Networks. The speed-invariance and the partial scale-invariance (object distance) of the classifier has been demonstrated.

References

[AMC$^+$15a] H. Akolkar, C. Meyer, Z. Clady, O. Marre, C. Bartolozzi, S. Panzeri, and R. Benosman. What can neuromorphic event-driven precise timing add to spike-based pattern recognition? *Neural Computation*, 27(3):561–593, March 2015.

[AMC$^+$15b] H Akolkar, C Meyer, Z Clady, O Marre, C Bartolozzi, S Panzeri, and R Benosman. What can neuromorphic event-driven precise timing add to spike-based pattern recognition? *Neural Computation*, 27(3):561–593, March 2015.

[ATB$^+$17] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey Mckinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A low power, fully event-based gesture recognition system. pages 7388–7397, 07 2017.

[BBD$^+$07] Daniel Bauer, Ahmed Nabil Belbachir, Nikolaus Donath, Gerhard Gritsch, Bernhard Kohn, Martin Litzenberger, Christoph Posch, Peter Schön, and Stephan Schraml. Embedded vehicle speed estimation system using an asynchronous temporal contrast vision sensor. *EURASIP Journal on Embedded Systems*, 2007(1):34–34, 2007.

[BBY$^+$14] C. Brandli, R. Berner, M. Yang, S. C. Liu, and T. Delbruck. A 240 x00d7; 180 130 db 3 x00b5;s latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, Oct 2014.

[BCL$^+$14a] R. Benosman, C. Clercq, X. Lagorce, S. H. Ieng, and C. Bartolozzi. Event-based visual flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):407–417, Feb 2014.

[BCL$^+$14b] R. Benosman, C. Clercq, X. Lagorce, Sio-Hoi Ieng, and C. Bartolozzi. Event-based visual flow. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(2):407–417, Feb 2014.

[BQT$^+$12] Olivier Bichler, Damien Querlioz, Simon J. Thorpe, Jean-Philippe Bourgoin, and Christian Gamrat. Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity. *Neural Networks*, 32:339 – 348, 2012.

[BTFA15] Francisco Barranco, Ching L. Teo, Cornelia Fermuller, and Yiannis Aloimonos. Contour detection and characterization for asynchronous event sensors. June 2015.

[CHL05] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, June 2005.

[CIB15a] Xavier Clady, Sio-Hoi Ieng, and Ryad Benosman. Asynchronous event-based corner detection and matching. *Neural Networks*, 66:91 – 106, 2015.

[CIB15b] Xavier Clady, Sio-Hoi Ieng, and Ryad Benosman. Asynchronous event-based corner detection and matching. *Neural Networks*, 66:91 – 106, 2015.

[CMBB17] Xavier Clady, Jean-Matthieu Maro, SÃľbastien BarrÃľ, and Ryad B. Benosman. A motion-based feature for event-based pattern recognition. *Frontiers in Neuroscience*, 10:594, 2017.

[CMGS11] D.C. Ciresan, U. Meier, L.M. Gambardella, and J. Schmidhuber. Convolutional neural network committees for handwritten character classification. pages 1135–1139, Sept 2011.

[CO16] D. Czech and G. Orchard. Evaluating noise filtering for event-based asynchronous change detection image sensors. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 19–24, June 2016.

[CS14] Andrea Censi and Davide Scaramuzza. Low-latency event-based visual odometry. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 703–710, 2014.

[CS16] Thusitha N. Chandrapala and Bertram E. Shi. Invariant feature extraction from event based stimuli. *CoRR*, 2016.

[DL07] T. Delbruck and P. Lichtsteiner. Fast sensory motor control based on event-based hybrid neuromorphic-procedural system. *Proc. IEEE Int. Symp. Circuits and Systems*, pages 845–848, May 2007.

[DRCB05] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proceedings of the 14th International Conference on Computer Communications and Networks*, ICCCN '05, pages 65–72, Washington, DC, USA, 2005. IEEE Computer Society.

[DT05] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

[EV09] Ehsan Elhamifar and RenÃľ Vidal. Sparse subspace clustering. In *CVPR*, pages 2790–2797. IEEE, 2009.

[FMA$^+$10] C. Farabet, B. Martini, P. Akselrod, S. Talay, Y. LeCun, and E. Culurciello. Hardware accelerated convolutional neural networks for synthetic vision systems. *Proc. IEEE Int. Symp. Circuits and Systems*, pages 257–260, Jun 2010.

[Fuk88] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1(2):119 – 130, 1988.

[GB17] Arren Glover and Chiara Bartolozzi. Robust visual tracking with a freely-moving event camera. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3769–3776, 2017. Exported from https://app.dimensions.ai on 2018/07/31.

[GLO$^+$16] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27 – 48, 2016. Recent Developments on Deep Big Vision.

[GMOT14a] R. Ghosh, A. Mishra, G. Orchard, and N.V. Thakor. Real-time object recognition and orientation estimation using an event-based camera and cnn. In *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE*, pages 544–547, Oct 2014.

[GMOT14b] R. Ghosh, A. Mishra, G. Orchard, and N.V. Thakor. Real-time object recognition and orientation estimation using an event-based camera and cnn. In *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE*, pages 544–547, Oct 2014.

[GS06] Robert Gütig and Haim Sompolinsky. The tempotron: a neuron that learns spike timing-based decisions. *Nat Neurosci*, 9(3):420–8, March 2006.

[HA89] W. Hoff and N. Ahuja. Surfaces from stereo: integrating feature matching, disparity estimation, and contour detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(2):121–136, Feb 1989.

[Hin02] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, August 2002.

[HOT06] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Comput.*, 18(7):1527–1554, July 2006.

[HS81] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[HW59] D. H. Hubel and T. N. Wiesel. Receptive fields of single neurons in the cat's striate cortex. *Journal of Physiology*, 148:574–591, 1959.

[HWC13] Raoul Hoffmann, David Weikersdorfer, and Jorg Conradt. Autonomous indoor exploration with an event-based visual SLAM system. In *2013 European Conference on Mobile Robots, ECMR 2013 - Conference Proceedings*, pages 38–43, 2013.

[HZT+09] Yaping Huang, Joali Zhao, Mei Tian, Qui Zou, and Siwei Luo. Slow Feature Discriminant Analysis and its application on handwritten digit recognition. *2009 International Joint Conference on Neural Networks*, (9):1294–1297, 2009.

[Jae01] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks. GMD Report 148, GMD - German National Research Institute for Computer Science, 2001.

[JXYY13] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):221–231, January 2013.

[KHB+14] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew Davison. Simultaneous mosaicing and tracking with an event camera. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[KMGS16] Beat Kueng, Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Low-latency visual odometry using event-based feature tracks. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2016-November, pages 16–23, 2016.

[KMM12] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, July 2012.

[KTS+14] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1725–1732, Washington, DC, USA, 2014. IEEE Computer Society.

[LGTB97] S. Lawrence, C.L. Giles, Ah Chung Tsoi, and A.D. Back. Face recognition: a convolutional neural-network approach. *Neural Networks, IEEE Transactions on*, 8(1):98–113, Jan 1997.

[LIC+15] Xavier Lagorce, Sio-Hoi Ieng, Xavier Clady, Michael Pfeiffer, and Ryad B. Benosman. Spatiotemporal features for asynchronous event-based data. *Frontiers in Neuroscience*, 9:46, 2015.

[LK81] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International*

Joint Conference on Artificial Intelligence - Volume 2, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[LMI$^+$15a] X. Lagorce, C. Meyer, S. H. Ieng, D. Filliat, and R. Benosman. Asynchronous event-based multikernel algorithm for high-speed visual features tracking. *IEEE Transactions on Neural Networks and Learning Systems*, 26(8):1710–1720, Aug 2015.

[LMI$^+$15b] X. Lagorce, C. Meyer, Sio-Hoi Ieng, D. Filliat, and R. Benosman. Asynchronous event-based multikernel algorithm for high-speed visual features tracking. *Neural Networks and Learning Systems, IEEE Transactions on*, 26(8):1710–1720, Aug 2015.

[LOG$^+$16] X. Lagorce, G. Orchard, F. Gallupi, B. E. Shi, and R. Benosman. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2016.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[LPD08a] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128 times; 128 120 db 15 956;s latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2):566–576, Feb 2008.

[LPD08b] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128 times; 128 120 db 15 956;s latency asynchronous temporal contrast vision sensor. *Solid-State Circuits, IEEE Journal of*, 43(2):566–576, Feb 2008.

[LSL$^+$16] Mengchen Liu, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, and Shixia Liu. Towards better analysis of deep convolutional neural networks. *CoRR*, 2016.

[LvSMD10] S. Liu, A. van Schaik, B. A. Mincti, and T. Delbruck. Event-based 64-channel binaural silicon cochlea with q enhancement mechanisms. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 2027–2030, May 2010.

[MAAI$^+$14] Paul A. Merolla, John V. Arthur, Rodrigo Alvarez-Icaza, Andrew S. Cassidy, Jun Sawada, Filipp Akopyan, Bryan L. Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, Bernard Brezzo, Ivan Vo, Steven K. Esser, Rathinakumar Appuswamy, Brian Taba, Arnon Amir, Myron D. Flickner, William P. Risk, Rajit Manohar, and Dharmendra S. Modha. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[MBS17] Elias Mueggler, Chiara Bartolozzi, and Davide Scaramuzza. Fast event-based corner detection. In *British Machine Vision Conference (BMVC)*, 2017.

[MFB+15] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza. Lifetime estimation of events from dynamic vision sensors. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4874–4881, May 2015.

[MGKK15] Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Jan Kautz. Hand gesture recognition with 3d convolutional neural networks. pages 1–7, 06 2015.

[MGP+17] Abhishek Mishra, Rohan Ghosh, Jose C. Principe, Nitish V. Thakor, and Sunil L. Kukreja. A saccade based framework for real-time motion segmentation using event based vision sensors. *Frontiers in Neuroscience*, 11(MAR), 2017.

[MKLD15] Michael Milford, Hanme Kim, Stefan Leutenegger, and Andrew Davison. Towards Visual SLAM with Event-based Cameras. *RSS Workshop*, (Figure 2):1–8, 2015.

[MLG+18] Ana I. Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso N. García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. *CoRR*, abs/1804.01310, 2018.

[MMT15] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *CoRR*, 2015.

[NIP+15] Zhenjiang Ni, Sio-Hoi Ieng, Christoph Posch, StÃľphane RÃľgnier, and Ryad Benosman. Visual tracking using neuromorphic asynchronous event-based cameras. *Neural Computation*, 27(4):925–953, 2015. PMID: 25710087.

[NP13] G. Nebehay and R. Pflugfelder. Tlm: Tracking-learning-matching of keypoints. In *2013 Seventh International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–6, Oct 2013.

[NPB+12] Z. NI, C. PACORET, R. BENOSMAN, S. IENG, and S. RÉGNIER*. Asynchronous event-based high speed vision for microparticle tracking. *Journal of Microscopy*, 245(3):236–244, 2012.

[OEC14] G. Orchard and R. Etienne-Cummings. Bioinspired visual motion estimation. *Proceedings of the IEEE*, 102(10):1520–1536, Oct 2014.

[OME+15] Garrick Orchard, Cedric Meyer, Ralph Etienne-Cummings, Christoph Posch, Nitish V. Thakor, and Ryad Benosman. Hfirst: A temporal approach to object recognition. *CoRR*, 2015.

[OMEC+15] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, and R. Benosman. Hfirst: A temporal approach to object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(10):2028–2040, Oct 2015.

[ONL$^+$13] Peter O'Connor, Daniel Neil, Shih-Chii Liu, Tobi Delbruck, and Michael Pfeiffer. Real-time classification and sensor fusion with a spiking deep belief network. *Frontiers in Neuroscience*, 7(178), 2013.

[PCP$^+$16] P K J Park, B H Cho, J M Park, K Lee, H Y Kim, H A Kang, H G Lee, J Woo, Y Roh, W J Lee, C W Shin, Q Wang, and H Ryu. Performance improvement of deep learning based gesture recognition using spatiotemporal demosaicing technique, 2016.

[PCZS$^+$13a] J.A. Perez-Carrasco, Bo Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, Shouchun Chen, and B. Linares-Barranco. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2706–2719, Nov 2013.

[PCZS$^+$13b] J.A. Perez-Carrasco, Bo Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, Shouchun Chen, and B. Linares-Barranco. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2706–2719, Nov 2013.

[PMW08] C. Posch, D. Matolin, and R. Wohlgenannt. An asynchronous time-based image sensor. In *2008 IEEE International Symposium on Circuits and Systems*, pages 2130–2133, May 2008.

[PMW10] C. Posch, D. Matolin, and R. Wohlgenannt. A qvga 143db dynamic range asynchronous address-event pwm dynamic image sensor with lossless pixel-level video compression. *IEEE Int. Solid-State Circuits Conf. Digest of Technical Papers*, pages 400–401, Feb 2010.

[PNBSG12] Ewa Piatkowska, Ahmed Nabil Belbachir, Stephan Schraml, and Margrit Gelautz. Spatiotemporal multiple persons tracking using dynamic vision sensor, 06 2012.

[pos16] Pose-invariant object recognition for event-based vision with slow-ELM. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9887 LNCS, pages 455–462, 2016.

[PPG$^+$13] E. Painkras, L.A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D.R. Lester, A.D. Brown, and S.B. Furber. SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation. *Solid-State Circuits, IEEE Journal of*, preprint, 2013.

[PZY$^+$17] Xi Peng, Bo Zhao, Rui Yan, Huajin Tang, and Zhang Yi. Bag of events: An efficient probability-based feature extraction method for AER image sensors. *IEEE Transactions on Neural Networks and Learning Systems*, 28(4):791–803, 2017.

[QYM17] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. *CoRR*, abs/1711.10305, 2017.

[Rei61] W. Reichardt. Autocorrelation: A principle for the evaluation of sensory information by the central nervous system. 1961.

[RVLC$^+$15] D. Reverter Valeiras, X. Lagorce, X. Clady, C. Bartolozzi, Sio-Hoi Ieng, and R. Benosman. An asynchronous neuromorphic event-driven visual part-based shape tracking. *Neural Networks and Learning Systems, IEEE Transactions on*, 26(12):3045–3059, Dec 2015.

[RVOIB15] David Reverter Valeiras, Garrick Orchard, Sio Hoi Ieng, and Ryad Benjamin Benosman. Neuromorphic event-based 3d pose estimation. *Frontiers in Neuroscience*, 9(522), 2015.

[SJC$^+$14] L. Sun, K. Jia, T. H. Chan, Y. Fang, G. Wang, and S. Yan. Dl-sfa: Deeply-learned slow feature analysis for action recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2632, June 2014.

[SLC04] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, ICPR '04, pages 32–36, Washington, DC, USA, 2004. IEEE Computer Society.

[SSP03a] Patrice Y. Simard, Dave Steinkraus, and John C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*, ICDAR '03, pages 958–, Washington, DC, USA, 2003. IEEE Computer Society.

[SSP03b] P.Y. Simard, D. Steinkraus, and John C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 958–963, Aug 2003.

[ST17] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810, 2017.

[SVI$^+$15] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[TBF$^+$15] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 4489–4497, Washington, DC, USA, 2015. IEEE Computer Society.

[TCGP09] D. N. Ta, W. C. Chen, N. Gelfand, and K. Pulli. Surftrac: Efficient tracking and continuous object recognition using local feature descriptors. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2937–2944, June 2009.

[TG08] Markus Meister Tim Gollisch. Rapid neural coding in the retina with relative spike latencies. *Science*, 319(5866):1108–1111, 2008.

[TM08] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Found. Trends. Comput. Graph. Vis.*, 3(3):177–280, July 2008.

[TZ15] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *CoRR*, abs/1503.02406, 2015.

[WC07] Shu-Fai Wong Shu-Fai Wong and Roberto Cipolla. Extracting Spatiotemporal Interest Points using Global Information. *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[WG15] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2794–2802, Dec 2015.

[WS02a] L Wiskott and T Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, April 2002.

[WS02b] Laurenz Wiskott and Terrence J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Comput.*, 14(4):715–770, April 2002.

[WUK$^+$09] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proc. BMVC*, pages 124.1–124.11, 2009.

[WXG$^+$16] Hanyu Wang, Jiangtao Xu, Zhiyuan Gao, Chengye Lu, Suying Yao, and Jianguo Ma. An event-based neurobiological recognition system with orientation detector for objects in multiple orientations. *Frontiers in Neuroscience*, 10:498, 2016.

[YCBL14] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3320–3328, Cambridge, MA, USA, 2014. MIT Press.

[ZAD17] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, volume 2017-January, pages 5816–5824, 2017.

[ZDC+15] Bo Zhao, Ruoxi Ding, Shoushun Chen, Bernabe Linares-Barranco, and Huajin Tang. Feedforward Categorization on AER Motion Events Using Cortex-Like Features in a Spiking Neural Network. *IEEE Transactions on Neural Networks and Learning Systems*, 26(9):1963–1978, 2015.

[ZYY+13] Bo Zhao, Qiang Yu, Hang Yu, Shoushun Chen, and Huajin Tang. A bio-inspired feedforward system for categorization of aer motion events. In *Biomedical Circuits and Systems Conference (BioCAS), 2013 IEEE*, pages 9–12, Oct 2013.