Real-Time Object Recognition and Orientation Estimation Using an Event-Based Camera and CNN

Rohan Ghosh, Abhishek Mishra, Garrick Orchard, and Nitish V. Thakor *Fellow, IEEE*
Singapore Institute for Neurotechnology (SINAPSE), National University of Singapore, Singapore
Email: rghosh92@gmail.com, abhishek.mishra@nus.edu.sg, garrickorchard@nus.edu.sg, eletnv@nus.edu.sg

Abstract—**Real-time visual identification and tracking of objects is a computationally intensive task, particularly in cluttered environments which contain many visual distracters. In this paper 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 event-based vision sensor detects only changes in the scene, naturally responding to moving objects and ignoring static distracters in the background. We present operation of the system 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%.**

I. 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. Pixels in such sensors individually and independently adjust to their lighting conditions, allowing operation over a wide intra-scene dynamic range. Furthermore, by only detecting changes in the scene, dependence on absolute levels of illumination is essentially removed. 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. Finally, despite operating over such a wide dynamic range, pixel outputs are restricted to a single binary bit, representing either an increase or decrease in intensity (or no output, representing no change in intensity). Using only three states per pixel (increase, decrease, or no change) significantly reduces computational requirements and also the time taken to train a Convolutional Neural Network (CNN) classifier [1].

CNNs are recognised as a powerful, bio-inspired tool for visual classification, providing high accuracy for tasks such as digit classification [2] [3]. They have been gaining popularity recently as interest has been increasing in using deep learning to handle large datasets.

Perez Carrasco *et al.* [1] 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 paper we combine an asynchronous event-based sensor, known as the Asynchronous Time-based Image Sensor (ATIS) [4], with a CNN implemented with the Neuflow architecture [5], achieving real-time tracking and identification of objects. Such a system can find many applications, two of which are addressed in this paper. Notably, the initial temporal binning method described later in this paper is along similar lines as [1], 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 [6] 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 paper 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 II, before describing the system testing in Section III. We then discuss results in Section IV.

II. Methods

The system consists of an asynchronous event-based vision sensor [4] for raw visual data acquisition, and Neuflow [5] 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 [7], before determining the temporal (Section II-A) and spatial (Section II-B) 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 II-C). The final images used for classification have been shown in Fig. 4.

A. 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.

$$\frac{dI(u,v,t)}{dt} = -\frac{dI(u,v,t)}{du}\frac{du}{dt} - \frac{dI(u,v,t)}{dv}\frac{dv}{dt} \tag{1}$$

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. 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.

B. 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 [7].

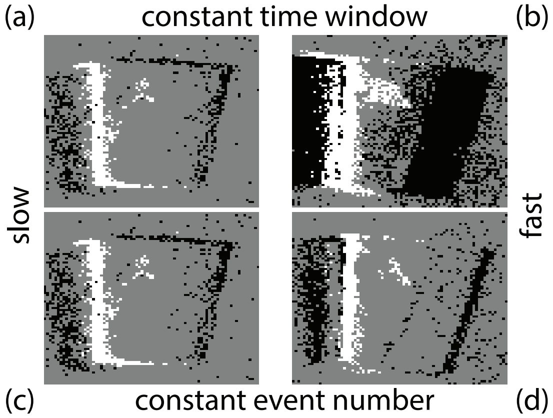

Fig. 1. Reducing speed dependence using a dynamic time window. The top row show views of the same object moving slow (a) and fast (b) extracted using a constant time window method (each image contains 33ms of data), while the bottom row shows views of the object moving slow (c) and fast (d) at the exact 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.

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.

C. 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.

III. Testing

The system was setup as a live demonstration for tracking and classifying vehicles and pedestrians passing by on a road (see Fig. 2) and subjectively appeared to provide accurate

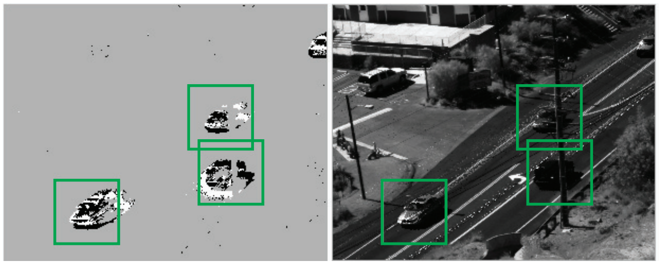

Fig. 2. Screenshot from live operation while tracking and identifying moving targets. Boxes indicate tracked objects, while colour indicates the object class (green indicates a car). The ATIS performs both change detection (left) and absolute grayscale measurements (right), but only change detection 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 I
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

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.

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.

A. 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.

TABLE II
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

B. 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.

C. 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.

D. 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 III-C). 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

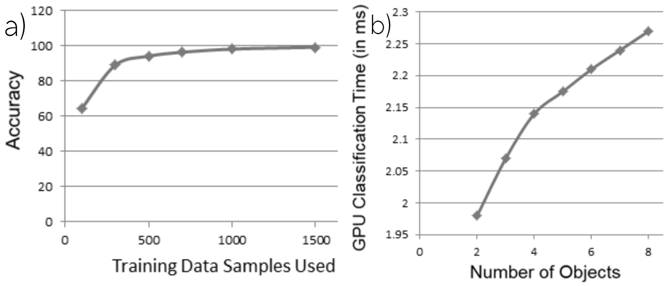

Fig. 3. (a) A plot of test data accuracy versus the number of training samples used. The accuracy is seen asymptotically reaching the value of 99.1% for large training datasets. (b) A plot of classification time of one sample (in milliseconds) versus the number of objects(output nodes) for the Network. As expected the classification time increases with the network size, but not completely linearly, as the GPU implementation is partially parallel in nature.

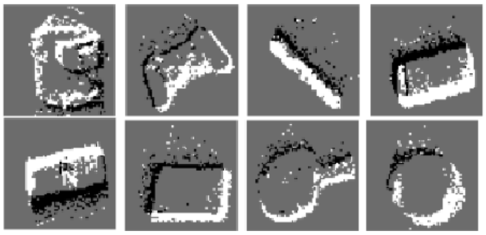

Fig. 4. Examples of object recordings used for testing and training. Top row from left to right: Bottle, Joystick, Pen, Mobiles, Box, Bat and Ball.

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

IV. Discussion

As we work towards developing an embedded system to visually assist an upper limb prosthesis in object grasping, we 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.

To improve classification accuracies, we worked with distorted datasets as in [8], 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.

V. 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.

References

[1] J. Perez-Carrasco, B. Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, S. 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*, vol. 35, no. 11, pp. 2706–2719, Nov 2013.

[2] S. Lawrence, C. Giles, A. C. Tsoi, and A. Back, "Face recognition: a convolutional neural-network approach," *Neural Networks, IEEE Transactions on*, vol. 8, no. 1, pp. 98–113, Jan 1997.

[3] D. Ciresan, U. Meier, L. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pp. 1135–1139, Sept 2011.

[4] 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*, pp. 400–401, Feb 2010.

[5] 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*, pp. 257–260, Jun 2010.

[6] D. Bauer, A. N. Belbachir, N. Donath, G. Gritsch, B. Kohn, M. Litzenberger, C. Posch, P. Schön, and S. Schraml, "Embedded vehicle speed estimation system using an asynchronous temporal contrast vision sensor," *EURASIP Journal on Embedded Systems*, vol. 2007, no. 1, pp. 34–34, 2007.

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

[8] P. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, Aug 2003, pp. 958–963.