

Event-Driven Ball Detection and Gaze Fixation in Clutter

Arren Glover, *Member, IEEE*, and Chiara Bartolozzi, *Member, IEEE*

Abstract—The fast temporal-dynamics and intrinsic motion segmentation of event-based cameras are beneficial for robotic tasks that require low-latency visual tracking and control, for example a robot catching a ball. When the event-driven iCub humanoid robot grasps an object its head and torso move, inducing camera motion, and tracked objects become no longer trivially segmented amongst the mass of background clutter. Current event-based tracking algorithms have mostly considered stationary cameras that have clean event-streams with minimal clutter. This paper introduces novel methods to extend the Hough-based circle detection algorithm using optical flow information that is readily extracted from the spatio-temporal event space. Results indicate the proposed *directed*-Hough algorithm is more robust to other moving objects and the background event-clutter. Finally, we demonstrate successful on-line robot control and gaze following on the iCub robot.

I. INTRODUCTION

Detection and tracking of moving objects is important for robots to operate in dynamic environments and many vision processing techniques already exist for estimating motion and tracking objects. Conventional cameras produce (compressed) streams of images that work well for humans to view and interpret. However, using such cameras in robotics requires the constant processing of large amounts of data, regardless of any change in the scene. Consequently, the camera frame-rate for on-line robotics is typically limited to 10-30 Hz and motion blur occurs when targets or cameras move quickly.

Biologically inspired event-based cameras (e.g. [1]) produce pixel events at a microsecond temporal resolution, “spiking” when the amount of light falling on a given pixel increases or decreases beyond a threshold. The cameras do not produce events for unchanging areas of the scene and data bandwidth is proportional to the actual scene motion. Event-based cameras are extremely interesting in robotics for accurate, real-time estimation of dynamic object motion and ego-motion (e.g. [2]–[4]) with low processing and power requirements. However, vision algorithms must be redeveloped to exploit the vastly different, asynchronous, event stream.

Dynamic vision sensors [1] (DVS) are used on the event-driven iCub robot [3] with the goal of exploiting the fast dynamics, towards low-latency, highly-accurate tracking of moving objects for grasping, catching, and avoidance during human-robot interaction. Event-based tracking methods have shown promising results for low-latency object tracking [5], [7]–[9] making them a strong candidate for use on the iCub. However, as the iCub moves its head to follow a

Arren Glover and Chiara Bartolozzi are with the iCub Facility, Istituto Italiano di Tecnologia, Italy. {arren.glover, chiara.bartolozzi}@iit.it

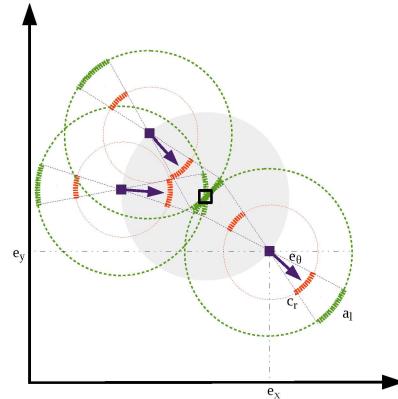


Fig. 1: The event-based Hough transform. A circular object (shaded) in view of the DVS produces events (solid squares) at edges. Each event’s Hough transform (green) will overlap at the object centre (hollow square) to produce a strong overall response for the correct radius, a smaller radius (orange) does not overlap at the centre. The proposed *directed* algorithm performs the Hough transform for a small arc perpendicular to the object’s edge (bold).

ball, the combination of camera motion and texture contrast in the environment produces many additional events that form background clutter from which the ball cannot be trivially segmented. Initial testing found that the state-of-the-art event-based circle detection algorithm [5] was detrimentally affected by background clutter or other moving objects; the robot was not able to reliably detect and specifically identify the ball when undergoing ego-motion.

A. Related Work

The current *frame-based* ball detection for grasping on the iCub involves a sliding window search to template images depicting the target object taken *a-priori*, and tracked with a particle filter [6]. The implementation is robust, with demonstrations successful in a wide variety of locations and conditions, but is tuned to a single object and required much developer time to create the template images. The speed of the ball is limited to slow movements if it is to remain detected, as any motion blur affects the template matching process. It is run at 30 Hz on a 320×240 pixel image.

Using event-based cameras, clusters have been tracked assuming a Gaussian distribution over the sensor space, approximately capturing object size and orientation [7]. Objects that do not fit the Gaussian model instead form multiple, separate clusters; tracking then requires modelling connectivity between clusters, for example when tracking a persons

face [8]. Tracking non-Gaussian objects has also been performed using filters hand-tuned to geometrical shapes (stars, crosses, etc.) [9].

An event-based Hough transform was used for tracking microparticles [5] that were all of a circular shape with an identical radius. The Hough-transform relies on the geometrical shape of a circle but has a simple, sequential implementation, making it ideal for event-driven data.

While many algorithms have been evaluated from a robotics perspective, less have been actually used to control a robotic platform on-line. Simple tracking was used to control a one-degree-of-freedom robotic ‘goalie’ that was able to block a rolling ball [7], and a DVS was used for localisation in a mobile robot navigation task [10]. However, more complex robots and tasks have only been approached through analysis of off-line data, for example quadcopter control [4] or stereo triangulation [11].

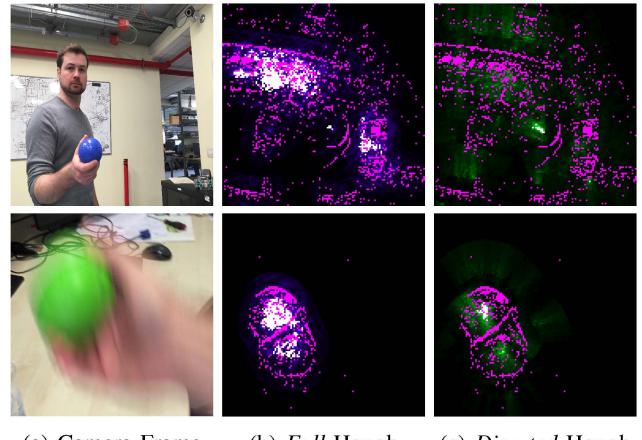
B. Contribution

The major contribution of this work addresses the problem of tracking a specific object during camera motion when many clutter events are perceived from the person’s hand and from the robot’s surrounding environment. Event-driven Gaussian tracking is not particularly sensitive to subtleties in object shape and cannot recognise a ball amongst clutter, and neither filter-based nor Hough-based approaches were previously used when clutter was present. Initial experimentation found that cluttered conditions adversely affected ball detections on the iCub robot using the Hough-transform, which is known to be sensitive to noise [12].

We propose to improve ball detection precision by augmenting the standard Hough transform with information readily extractable from the spatio-temporal event-based camera signal. As such, we introduce a *directed*-Hough transform. The proposed approach uses optical flow techniques to estimate the direction of the circle centre from any event that occurs on its perimeter. The voting mechanism of the Hough-transform is then only applied in the estimated direction of the centre, maintaining a strong response to circular shapes, while removing the large portion of votes that are redundant but contribute to erroneous detections in a cluttered scene. Edge orientation can easily be calculated using DVS cameras, as edges are extracted ‘for free’ and the high temporal resolution further distinguishes coherent edges amongst spatial clutter. We also introduce and demonstrate an event-driven Hough transform that functions over multiple radii, as the ball must be tracked at different distances from the camera (previously only single radius was used [5]).

The use of edge orientation based on dynamic movement of objects/cameras concisely captures important visual information and is a relatively simple task in the event-based data. We propose the use of such information could also improve performance in other event-driven problems (shape modelling, feature interest points, etc.) and therefore the presented technique also has some generalised applicability.

We first describe the proposed algorithm (Section II) and characterise its performance (Section III). The algorithm is



(a) Camera Frame (b) *Full*-Hough (c) *Directed*-Hough

Fig. 2: An example camera frame (30 Hz) re-enacting the hand and eye movement, *em*, dataset (top) and fast hand, *fh*, movement dataset (bottom). Events (purple) are produced due to any motion and the resulting Hough transform (combined across all radii for visualisation) is indicated by the coloured regions; white colour indicating a strong circle detection. Typical failure cases in the *full*-Hough transform (middle) occur due to clutter either from the hand or background. The proposed *directed*-Hough transform (right) produced a more focussed response in clutter.

then implemented on the event-driven iCub robot to perform DVS-in-the-loop gaze following (Section IV). In contrast, much prior event-based work is fully evaluated off-line [5], [8], [14], [15] or with non-moving cameras [2], [7]. The significance of the results and proposed future are finally discussed (Section V).

II. EVENT-BASED BALL DETECTION

A single event will project a *full* circle into the Hough space. The transforms of all events sitting on the perimeter of a circle overlap in its centre, resulting in a strong response that is used to extract the circle’s position and size (Fig. 1). However, the Hough transform is sensitive to noise [12], which was observed with event-based data, when many events occurred in tight clusters (Fig. 2). We propose the *directed*-Hough transform, in which only an arc of the full circle is projected into the Hough space (towards the circle centre) to reduce excess overlap in directions that are known to be incorrect.

The direction of the centre can be found if the orientation of the edge (or its tangent) at the event location is known. Edges can readily be extracted from event-driven data as events only occur due to contrast changes (i.e. on the edges of objects) and the micro-second temporal resolution of events helps to distinguish ridges in temporal space, even amongst spatial clutter. We take advantage of an established optical flow technique [14] that uses plane-fitting to find the direction of flow normal to the edge orientation (i.e. due to the aperture problem all flow will be normal to the edge [15], [16]). We therefore do not need to calculate the orientation of

an entire edge, but can find the perpendicular directly from the flow direction, which will always point directly towards (or away from) a circle's centre.

We first describe the event-based representation, and *full-Hough* transform before describing the optical flow and *directed-Hough* algorithm.

A. Event-based Representation

An event ($e_{x,y,p,t}$) occurs when a pixel on the DVS detects a contrast change. The event is defined by the spatial location of the pixel (x, y), the polarity (p) (increase/decrease in illumination), and the detection time (t). An event-based camera differs from a frame-based camera in that it provides asynchronous information at microsecond temporal resolution; instead of images, the data from an event-based camera forms manifolds in spatio-temporal space.

B. Multi-Radius Event-based Hough Transform

We follow the Hough transform as described in [5], and augment it using the *directed* approach while also adding multiple detection scales (as the distance between ball and camera changes). The parametrisation of the circle which defines the Hough transform coordinates, given the event position e_x and e_y , is:

$$(e_x - c_x)^2 + (e_y - c_y)^2 = c_r^2 \quad (1)$$

where the possible values of c_x , c_y and c_r that satisfy the above equation are used to vote for likely circle positions.

In addition to the original algorithm, the Hough transform must be normalised to account for variation in circle radius as larger circles have more pixels and therefore attract more votes. Simply normalising the final hough response, \hat{h} by the perimeter ($\hat{h} = \frac{\hat{h}}{2\pi c_r}$) is inaccurate for small circles due to pixel quantisation, hence normalisation is performed according to the discrete number of pixels that lie on a circle as defined in Eq. 1 given a half a pixel quantisation error. The normalisation value can be found before operation by sampling in pixel space according to this constraint.

The most recent 1500 events are transformed into the Hough space and the largest response indicates the most probable circle position and radius. The value of the response is used as a measure of confidence. We use a fixed number of events following [5], however this approach will be sensitive to the total amount of contrast in the scene; more robust event-representation methods are still under investigation (e.g. [15]).

C. Flow Angle for the Directed-Hough Transform

The calculation of optical flow using a plane fitting technique is used to determine the normal flow relative to the edge orientation [15], [16]. While the aperture problem is usually a cause of error, a measurement of the orientation of the object edge is still useful information for determining object shape. In the case of a circle, the normal flow direction around the perimeter always points to the centre of the circle.

Planes are fit to the surface of events as per [14]. Given the equation of a plane, $ax + by + ct + d = 0$, the components

of the direction of the slope of the plane, dt/dx and dt/dy , are extracted as a/c and b/c , respectively.

The *directed* Hough transform updates Eq. 1 given the measurement of normal flow angle, e_θ , only for positions x_c and y_c , for a given value of r_c , such that:

$$x_c = e_x \pm r_c \cos(e_\theta \pm a_l) \quad \text{and} \quad y_c = e_y \pm r_c \sin(e_\theta \pm a_l) \quad (2)$$

where a_l is the length of a small arc (10°) and e_θ is calculated as:

$$e_\theta = \arctan\left(\frac{dt/dy}{dt/dx}\right) \quad \text{or} \quad e_\theta = \arctan\left(\frac{b}{a}\right) \quad (3)$$

The *directed* transform is therefore not as dense as the full transform and aims to reduce excess overlap from clusters of events that do not have circular edges, especially due to tight but uncorrelated clusters of events. We use a voting bin size for the Hough transform equal to the size of one pixel.

III. ALGORITHM COMPARISON

The detection performance of the *directed-Hough* algorithm is first compared to the original *full-Hough* algorithm using two off-line datasets that contain clutter events. The algorithms were not compared without clutter as the solution was shown to be robust [5].

A. Experiments

The ball is held by a person and moved in front of the iCub robot as the event stream is recorded. In the *hand-move* dataset (*hm*) the iCub remains stationary and the ball is moved in a circular pattern with a frequency of approximately 2.5 Hz, with change in height, width and depth of over 0.5 m. Note that the robust frame-based ball detection did not track the ball at these speeds. Clutter events occur from the movement of the hand and arm and always appear close to the ball position. The *eye-move* dataset (*em*) is taken as the iCub moves its eyes with both vertical and horizontal motion, as if to gaze at the vertices of an imaginary square. The ball is moved slowly, but with similar changes in position as in *hm*. Background clutter occurs due to the movement of the camera within the laboratory environment (Fig. 2).

Background clutter is mostly dependent on the background edge configurations rather than the type of motion of the cameras (given two orthogonal directions were used). Adding further directions of camera motion only minimally changes the clutter, and the effect on ball detection performance is insignificant.

The ground truth ball position was annotated by visualising 'frames' of events over fixed temporal intervals. Ball position and radius was manually tagged at minimum 0.1 s intervals. We assume manual tagging of the circle state is accurate to less than two pixels.

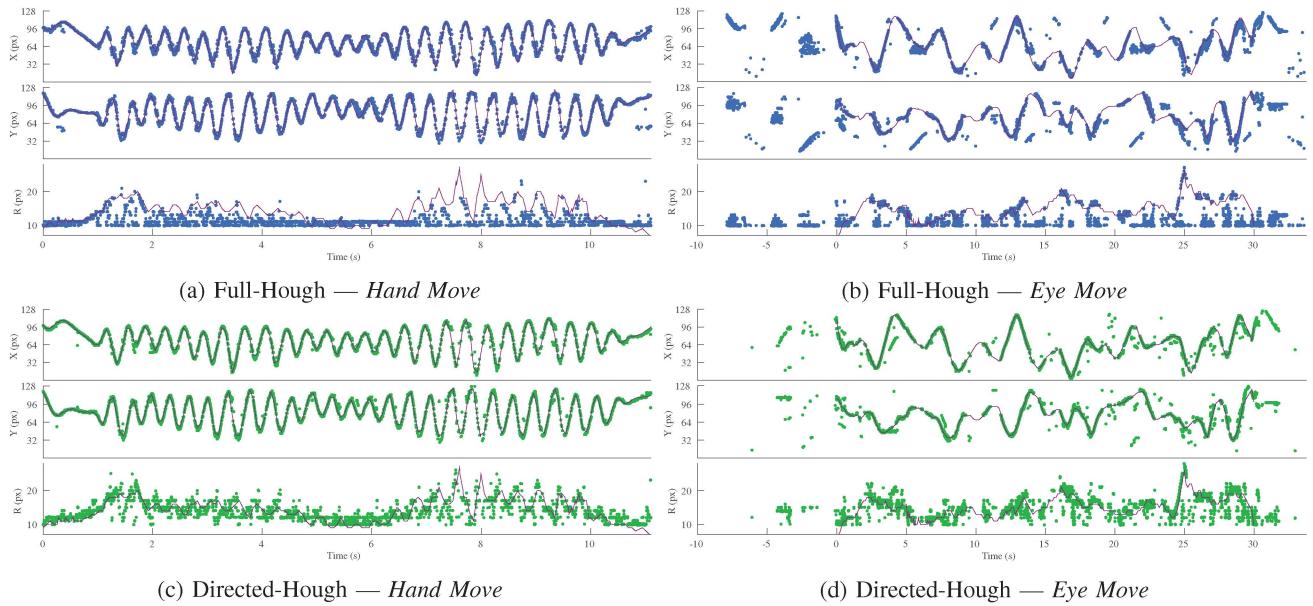


Fig. 3: Ball observation results. The ground truth is shown as a continuous line and only observations above the acceptance threshold are plotted.

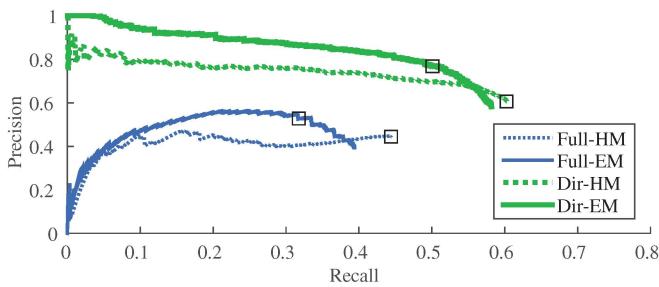


Fig. 4: Detection Precision and Recall. The percentage of observations that agree with the ground truth (precision) is plotted against the percentage of time in which the ball is detected (recall), given a threshold change. Inliers are defined as observations within 6 pixels from the ground truth and within 2 pixels radius error (approx. 5% error). Squares indicate appropriate thresholds to remove low strength observations.

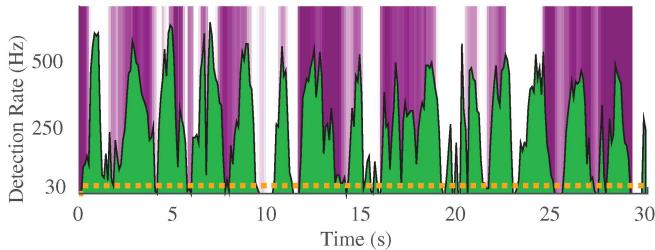


Fig. 5: Detection Rate. The DVS detection rate (green) is compared to a theoretical 30 Hz frame-based camera (brown) in the *em* dataset. Peaks correspond to increased ball motion (purple) but do not occur when the ball is stationary.

B. Results

A comparison of the detection precision at different recall rates is shown in Fig. 4, which indicates the *directed*-Hough algorithm achieved a significantly higher performance in both datasets. The similar curves indicate an overall consistent performance of the *directed*-Hough algorithm.

At high thresholds (i.e. low recall), the precision of the *full*-Hough approaches zero, which indicates the strongest circle response actually comes from clutter, as opposed to a true circle observation. Error in the *full*-Hough algorithm occurred due to clutter events, which were present in both datasets, but from different sources. Examples of which are shown in Fig. 2, and which are reflected in the observation plots in Fig. 3.

In the *hm* dataset clutter events came from the hand and arm; the cluster as a whole is tracked well (see the good positional accuracy in Fig. 3a), however the radius is often detected at the smallest value. The event clutter in a small area produces an imprecise Hough transform as overlapping occurs frequently, and is biased to the smallest radius as it has the largest normalisation factor. The position estimation is correct as no events occur outside the hand and ball. In comparison, the *directed*-Hough transform produces less overlap outside the circle centre and the radius is therefore better estimated (Fig. 3c).

In the *em* dataset clutter occurs in the background as the iCub itself moves. The background clutter results in incorrect position estimates which consistently occur in the same locations (Fig. 3b). These locations correspond with dense regions of events caused by structure in the environment. The estimate of radius is also somewhat erroneous due to hand clutter. Incorrect positional estimates are more randomly distributed for the *directed*-Hough transform (Fig. 3d) and

therefore are more likely to be rejected by the potential application of outlier removal.

The detection rate of the *directed*-Hough transform under background clutter reached over 500 Hz; somewhat correlated with periods of fast motion (Fig. 5). The ball cannot be detected when the relative velocity to the camera motion is zero, which can often occur, despite constant ball motion, when the direction of motion of the ball and camera align. The event-based camera resulted in more detections when the ball was moving (important for planning reactive motions in dynamic environments), and less/zero when the ball was visually in the same location (reducing processing requirements). Filtering could be used to improve the permanence of the estimate in periods of low motion, however, it would still be important that background clutter was rejected during these times, as occurs with the *directed*-Hough transform.

Two factors should be taken into account when interpreting the recall achieved in Fig. 4: 1. The expected recall should not be 100% as we cannot expect detections when there is no relative motion. 2. The recall values shown are sampled at 1 ms, i.e. 50% recall can mean that for every second the ball is still observed correctly 500 times. We therefore propose that the levels of recall achieved could be suitable for visual tracking on a robot.

IV. ON-LINE GAZE FIXATION

The *directed*-Hough transform is implemented on the iCub robot for on-line detection and gaze fixation.

A. The Event-driven iCub

Gaze following on the event-driven iCub involves integrating the event-based hardware within the YARP environment such that events can be extracted, detection algorithms performed on-line, and finally the gaze control of the neck and eyes of the iCub. The event-driven project¹ contains the open-source implementations of the described algorithms which is combined with the iCub gaze controller [17]. The gaze controller uses the inverse kinematics to move the neck and eye joints given a specified gaze target in the 3D robot reference frame under position control. The gaze controller's desired state is set from every ball detection (up to 1 kHz, only the left camera), and the iCub continues its previous gaze trajectory in periods without detections.

The transform from camera to robot reference frame requires calibration of the forward kinematics between the torso, neck joints and eye positions. The camera's optical calibration is performed before operation by viewing known fiducial markers (asymmetric-circle-grid as defined in OpenCV), applying filtering to events 'framed' in a temporal window, and finally using standard calibration techniques.

The 3D position of the ball is calculated using the full state (x, y, r) of the *directed*-Hough detections. In the future we plan to estimate depth using stereo vision, leveraging previous work [11], however, as an initial implementation, we assume the ball size is known *a-priori* and use the

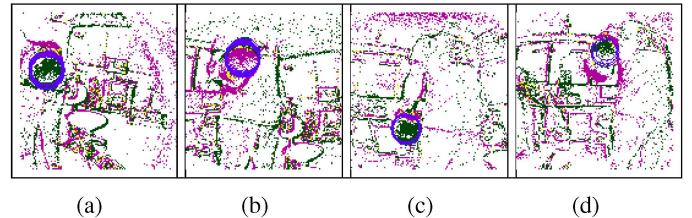


Fig. 6: Example gaze following on the iCub. The ball is (a) detected on the left hand side, and the robot (b) tracks to the left. Afterwards the ball is (c) moved downwards and the robot gaze follows (d). The colour of the event indicates its polarity.

detected radius to estimate depth. A linear mapping between radius size and depth was modelled for typical operating depths. In reality the relationship is not linear and errors will be present in extreme positions (too far, or too close); for these studies the focus is not on transform precision, rather the on-line control of the robot using the DVS and the effects on the visual stream due to robot motion.

B. Experiments

The robot is positioned into a neutral position, looking forward, and the ball is held by a person, initially in the lower-left portion of the field of view. The ball is moved in a square pattern (up, right, down and then left) at approximately 45 cm from the robot, and then again, approximately 30 cm from the robot. The gaze controller is used to move the head and eyes of the robot to focus on the ball position, given the circle detections. Examples of gaze following on the robot are shown in Fig. 6.

C. Results

Ball observations correctly occur in the square pattern in which the ball was moved (circles in Fig. 7, note the robot reference frame used). However, the detection rate during robot gaze following is much more sparse than expected from off-line results (we do not reach 500 Hz). We attribute the drop in detection rate to a drop in relative velocity between the ball and the camera, resulting in a drop in the number of events that contributed to ball observations. During a gaze following task, by definition, the relative velocity between the camera and target is controlled to approach zero. It is therefore only at changes in ball direction, or after the robot stops motion, that the ball detection signal becomes strong enough for subsequent detection. The resulting behaviour is a saccadic gaze following motion.

Importantly, when the ball is less visible and the background more prominent during a saccade, the clutter from the background is still rejected by the *directed*-Hough algorithm. Incorrect gaze motions are not performed.

The robot's gaze follows the sparse ball detections: the first square pattern (Fig. 7a) begins in the centre of the figure before moving to the bottom left; and then continues in a clock-wise direction. A perfect square is not formed as the Gaze Controller must solve the inverse kinematics to reach

¹<https://github.com/robotology-playground/event-driven>

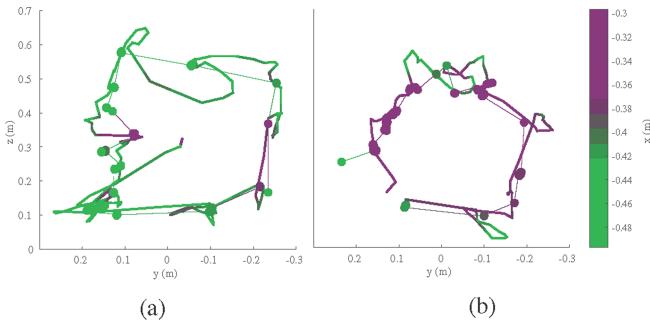


Fig. 7: Ball position in the robot reference frame. The ball more sparse observations (round markers) is transformed to the Cartesian robot reference frame and the robot's gaze (line) follows the detections. The ball is moved in a square pattern (a) appox. 45 cm and (b) approx. 30 cm from the robot's head.

the correct position; in particular the ‘dip’ at the top of the figure occurs as the robot reconfigures its neck position to avoid collision between head and torso when looking at a high z coordinate and moving from left to right. A similar trace occurs for the second square pattern (Fig. 7b) at a closer distance to the robot; seen as the average x coordinate is smaller (in the negative direction).

The transform of the ball position to the robot reference is not always perfect; errors are present in the kinematic chain and camera calibration, and some noise is present in the estimated radius.

V. CONCLUSIONS AND FUTURE WORK

Robots must move to complete tasks and a robot with a DVS camera will therefore always experience an event-stream including background clutter; detection and tracking amongst such clutter is an important problem that as of yet has had less attention. This paper introduced the *directed*-Hough transform that augmented the traditional circular Hough-transform with spatio-temporal directional information (from easily computed DVS optical flow) for ball detection during camera motion, and achieved a significantly higher precision than the standard algorithm. We show that even in cluttered conditions we can still achieve detection rates up to 500 Hz and achieve detection of a fast moving stimulus for which the frame-based method fails. The size of the ball was also found by applying the transform at multiple radii.

The *directed*-Hough algorithm achieved a high precision during in-the-loop gaze fixation; all background clutter was rejected using the algorithm. The detection rate, however, dropped significantly resulting in more of a saccadic, rather than smooth, gaze following behaviour. In this case, the act of actively following the ball reduced the relative velocity between ball and camera, which therefore reduced the number of events indicating the ball’s position. We do not believe this is an inherent problem with the DVS itself, but rather a limitation of using a fixed FIFO (following [5]) as it is biased towards large, fast moving objects. We propose that a more

informative underlying event-representation could improve the detection rate for gaze following. However, designing a robust replacement that is less sensitive to this issue is still an open problem (e.g. see [15] for one approach).

Probabilistic filtering could also be used to improve tracking consistency and perform outlier rejection. Kalman filters and particle filters used with event-based data have only used fixed update rates, which requires tuning based on data, task and computation power available. An elegant event-driven implementation of such a filter is still an open research question. Finally, we plan to implement stereo depth estimation to improve the robustness of the position estimation, such that the robot is able to reach and grasp the ball.

REFERENCES

- [1] P. Lichtsteiner, C. Posch, and T. Delbrück, “A 128x128 120 dB 15 us latency Asynchronous Temporal Contrast Vision Sensor,” *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.
- [2] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. J. Douglas, and T. Delbrück, “A pencil balancing robot using a pair of AER dynamic vision sensors,” in *IEEE International Symposium on Circuits and Systems*, 2009, pp. 781–784.
- [3] F. Rea, G. Metta, and C. Bartolozzi, “Event-driven visual attention for the humanoid robot iCub,” *Frontiers in Neuroscience*, vol. 7, no. December, p. 234, jan 2013.
- [4] E. Mueggler, B. Huber, and D. Scaramuzza, “Event-based , 6-DOF Pose Tracking for High-Speed Maneuvers,” in *The IEEE International Conference on Intelligent Robots and Systems*, Chicago, USA, 2014.
- [5] Z. Ni, C. Pacoret, R. Benosman, S. Ieng, and S. Régnier, “Asynchronous event-based high speed vision for microparticle tracking,” *Journal of Microscopy*, vol. 245, no. 3, pp. 236–244, 2012.
- [6] M. Taiana, J. Santos, J. Gaspar, J. Nascimento, a. Bernardino, and P. Lima, “Tracking objects with generic calibrated sensors: An algorithm based on color and 3D shape features,” *Robotics and Autonomous Systems*, vol. 58, no. 6, pp. 784–795, 2010.
- [7] T. Delbrück and M. Lang, “Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor,” *Frontiers in Neuroscience*, vol. 7, no. 7 NOV, p. 223, jan 2013.
- [8] D. R. Valeiras, X. Lagorce, X. Clady, C. Bartolozzi, S.-h. Ieng, and R. Benosman, “An Asynchronous Neuromorphic Event-Driven Visual Part-Based Shape Tracking,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2015.
- [9] 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*, vol. 26, no. 8, pp. 1710–1720, sep 2015.
- [10] D. Weikersdorfer and R. Hoffmann, “Simultaneous Localization and Mapping for event-based Vision Systems,” in *International Conference on Computer Vision Systems (ICVS)*. Springer, 2013, pp. 133–142.
- [11] R. Benosman, S. H. Ieng, P. Rogister, and C. Posch, “Asynchronous event-based Hebbian epipolar geometry,” *IEEE Transactions on Neural Networks*, vol. 22, no. 11, pp. 1723–1734, nov 2011.
- [12] J. Illingworth and J. Kittler, “The Adaptive Hough Transform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 690–698, 1987.
- [13] P. Fitzpatrick, G. Metta, and L. Natale, “Towards long-lived robot genes,” *Robotics and Autonomous Systems*, vol. 56, no. 1, pp. 29–45, 2008.
- [14] R. Benosman, C. Clercq, X. Lagorce, S. H. Ieng, and C. Bartolozzi, “Event-based visual flow,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 407–417, feb 2014.
- [15] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza, “Lifetime Estimation of Events from Dynamic Vision Sensors,” in *The IEEE International Conference on Robotics and Automation*, 2015.
- [16] T. Brosch, S. Tschechne, and H. Neumann, “On event-based optical flow detection,” *Frontiers in Neuroscience*, 2015.
- [17] U. Pattacini, “Modular Cartesian Controllers for Humanoid Robots: Design and Implementation on the iCub,” Ph.D. dissertation, Istituto Italiano di Tecnologia, 2011.