6th IEEE RAS/EMBS International Conference on
Biomedical Robotics and Biomechatronics (BioRob)
June 26-29, 2016. UTown, Singapore

Real-Time Robot Tracking and Following with Neuromorphic Vision Sensor

Abhishek Mishra[1], Rohan Ghosh[1], Ashish Goyal[2],
Nitish V. Thakor[3], *Fellow, IEEE* and Sunil L. Kukreja[4], *Senior Member, IEEE*

***Abstract*— In this paper, we consider the problem of robotic motion tracking and following with neuromorphic vision sensors. We formulate the problem in a leader-follower paradigm. The objective of the follower robot is to perform real-time motion segmentation of a scene and follow the leader robot. Motion segmentation using a neuromorphic vision sensor mounted on a mobile robot is a challenging task due to events created by movements of the platform (self-movement). Current approaches for tracking do not perform well during sensor ego-motion or need *a priori* knowledge about the object being tracked. To address these limitations, we designed an algorithm based on clustering space-time events induced by a neuromorphic sensor followed by a classification procedure. This technique is based on a distance transformation of existing sets. After clustering, a binary class label is assigned to each: (1) background or (2) moving object. The classifier uses event rates of clusters to determine proper class labels. The proposed technique forms an important module for the creation of collectively intelligent multi-pedal robots that utilize neuromorphic vision sensors. The utility and robustness of our algorithm is demonstrated as a real-time online system implemented on two hexapod robots.**

I. Introduction

A critical task for autonomous robots is the ability to track and follow a targeted moving object. These objects may have a variety of movement patterns and sizes (e.g. person, robot, vehicle, etc.). Prior tracking-following techniques have utilized RGB-D cameras [1]–[5], lasers [6], [7], stereo camera pairs [8] and Microsoft Kinect^{TM} [9] based sensors. However, some approaches are specific to a particular object to be tracked and, therefore, require a greedy data search and classification approach. Clustering and classification using data from a frame rate camera demands significant computational power and time due to high information redundancy.

*This Research is funded by the Ministry of Defence, Singapore, Grant Number: R-719-004-102-232..

[1]Abhishek Mishra and Rohan Ghosh are with the Singapore Institute for Neurotechnology (SINAPSE), National University of Singapore (NUS), Singapore. abhishek.mishra@nus.edu.sg, rghosh92@gmail.com

[2]Ashish Goyal is with the Department of Electrical Engineering, Indian Institute of Technology, Bombay, India. goyal26@outlook.com

[3]Nitish V. Thakor is the Director of the Singapore Institute of Neurotechnology (SINAPSE) and Professor of Electrical and Bioengineering at National University of Singapore, and Professor of Biomedical Engineering at the Johns Hopkins University sinapsedirector@gmail.com

[4]Sunil Kukreja is the Head of Neuromorphic Engineering and Robotics at the Singapore Institute for Neurotechnology (SINAPSE), National University of Singapore (NUS), Singapore sunilkukreja.sinapse@gmail.com

Alternatively, event based vision sensors such as the dynamic vision sensor (DVS) [10] or asynchronous time-based image sensor (ATIS) [11] provide asynchronous data at a temporal resolution on the order of μs. Since neuromorphic sensors only respond to dynamic intensity changes, the computational overhead is minimal. In addition, due to their analog design, they have low power requirements. In this paper, we implement an event based dynamic vision sensor on a six legged robot (hexapod) to develop a motion segmentation technique that identifies moving objects against a static background scene.

Previous tracking methods have used neuromorphic vision sensors to implement pose estimation of known objects [12]. This method is robust to sensor ego-motion and performs well even in cluttered scenes. However, a limitation of their algorithm is that it needs a prior knowledge about the object to be tracked. Convolution AER vision architecture for real-time systems (CAVIAR) is a multichip multilayer hardware, capable of real-time tracking and object recognition using a large scale spiking neural network (SNN) [13]. Although this biologically inspired network performs well for tracking simple geometric shapes, its use is limited by the complexity of setup and dependency on object recognition for motion segmentation. Another technique tracks a generic class of objects using a flexible part-based description [14], [15]. This approach performs best in situations where the scene has low or no clutter and when a 3-dimensional object can be described accurately as a combination of smaller parts. Other procedures using spatiotemporal clustering with neuromorphic sensor data have been presented for tracking people [16], grippers for haptic feedback [17], particles in a fluid [18] and traffic monitoring [19]. The problem of tracking has also been formulated as one of finding salient objects in a scene. This event-driven visual saliency approach to tracking was implemented on the iCub platform [20], [21]. Clearly, existing methods for tracking have two main limitations related to: (1) ego-motion and (2) prior knowledge about an object.

In this paper, we use hexapods as our robotic platforms because they provide the advantage of being able to manoeuvre in a large variety of terrains. They also have the ability to rapidly adapt to different forms of injury [22]. Hexapods have a complex, non-linear and periodic motion profile, which poses a challenge for ego-motion compensation algorithms. For this reason, we use the temporal variation and periodicity of motion magnitude for object tracking instead of an approach involving motion estimation [23].

Our algorithm consists of two major steps as follows. (1) Clustering of space-time events based on distance transform maps. (2) Classification of clusters based on their temporal variance. The emphasis of this paper is primarily on the motion segmentation aspect of object tracking and following. Our main contribution is the development of a framework for performing motion segmentation using multipedal robots as the platform.

The paper is organized as follows. We describe the functionality of event-based vision sensors and details of our algorithm based on these sensors in Section II. The results of our experiments to assess the effect of various parameters on tracking performance are presented in Section III. In Sections IV we discuss the unique features of our algorithm and its shortcomings. We provide concluding remarks and future directions in Section V.

II. Methods

The segmentation approach implemented in this paper uses clustering of spatio-temporal data followed by classification of each cluster. The classification methodology is based on knowledge that when a hexapod moves in a tripod gait configuration, the speed varies sinusoidally. This change of speed results in a corresponding modulation of event rates from the neuromorphic sensor. Hence, we hypothesize that event rates of background clusters will correlate to the total event rate and be distinct from event rates associated with the leader hexapod. This clustering approach is described in Section II-C and event rate classification technique in Section II-D.

A. Event Based Vision Sensors

Conventional frame-based cameras acquire pixel-intensity information from an entire image at regular time intervals. However, any movement of the camera frame of reference causes only incremental changes in the visual scene. This results in data redundancy, leading to unnecessary computational overhead. In contrast, frame-free asynchronous imagers like the DVS capture intensity information at a pixel only when there is a change in its visual field. This allows for a natural correlation of incoming dynamic scene information and motion magnitude of the robot. This feature allows a system to increase the computational load only when there is a change in the environment. A motion event, $E_i(x,y,p,t)$, is defined as an intensity change of a pixel located at (x,y) at time t with polarity p, indicating an intensity increase or decrease on a logarithmic scale. The subscript i denotes the i^{th} motion event in the spatio-temporal data. The pixel array is 128×128 with a time resolution of μs. Hence, the data, S, can be represented as a collection of motion events, E_i, expressed as $S = \{E_i,\ i = 1,....,N\}$. The high temporal resolution of the sensor allows for class characterization using statistics derived from event rates, which is not possible with frame based approaches.

B. Hexapod robots used

Two hexapod robots were used for experimental application. They were controlled by an Intel Edison compute module in addition to a custom controller used to send servo commands to the six legs. Each leg of the hexapod consists of three servo motors in a daisy chain configuration. This provides three degrees of freedom for each leg. The robot is powered by a lithium polymer battery and controlled through the Intel Edison's onboard Wi-Fi. The gait control algorithm for the hexapod is based on inverse kinematics and supports movements in different directions. The leader hexapod was controlled by a user while the follower hexapod's goal was to accurately perform the task of motion segmentation using the neuromorphic vision sensor and follow the leader. The hexapod used in this study is shown in Fig. 1.

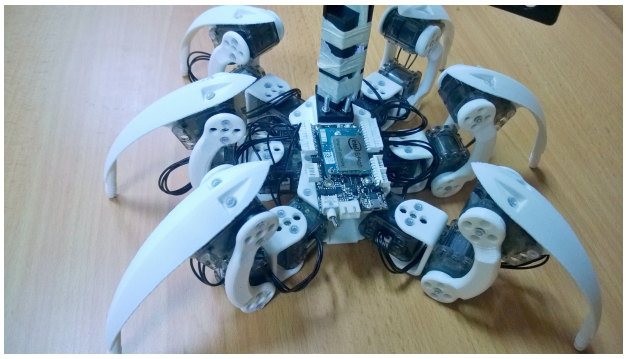

Fig. 1. 3D printed hexapod used in experiments. The locomotion algorithm is based on inverse kinematics and embedded in the Intel Edison. Communication between robots is achieved through sockets using the onboard Wi-Fi.

C. Clustering using distance transform metric

We use a matching approach based on the work of Ni, et al. [24]. However, instead of matching against a pre-defined object template, we perform matching with respect to the last classified model of the leader hexapod. The algorithm is initialized by forming clusters, C_j, using a k-means approach [25]. Each event cluster is represented as a binary image of size 128×128 and a distance transform map [26], D_j, of each cluster is computed. For each cluster, let the set P denote the motion events locations (or the cluster points, such that $P \cup \bar{P} = C_j$). For all pixel locations $n = (x,y)$ in the distance transform map its value is calculated as

$$D_j(n) = \min_{l \in P} ||l - n||_2 \tag{1}$$

where $||.||_2$ represents the Euclidean distance between two points l and n. Let E_{i+1} denote a new event. Each E_{i+1} is assigned to a cluster, C_k, where the index k is calculated by the following rule

$$k = \underset{\forall j}{\operatorname{argmin}} D_j(E_{i+1}). \tag{2}$$

The expression in Eqn. 2 assigns a new motion event to a cluster C_k which has the minimum distance transform value for that location across all maps, D_j. Distance transform maps are updated when new events occur in a cluster. To reduce computational load, we update the distance transform maps after p number of events and maintain a master distance transform map. Each pixel in the master map represents the

minimum value of distance transform across the individual maps. The pseudo code for our real-time clustering and map update is given in Algorithm 1.

Algorithm 1: Clustering events

Data: Events E_i, dist_thresh, N
Result: Clusters

```
while Events are valid do
    read events
    if ∃ cluster C_j :dist(E_i,C_j) <dist_thresh then
        assign E_i to C_j
        update distance transform
    else
        pass E_i to buffer
    if buffer length >N then
        Form new clusters using k-means
    else
        go back to read events
```

Fig. 2A shows a sample data frame, and its master distance transform map Fig. 2B calculated from the individual distance transform maps shown in the right column Fig. 2C.

D. Classification of clusters

To classify clusters as part of the dynamic object or background, we exploit the information present in the temporal variation of event-rates. We assume that velocity of the sensor varies with time. This is a reasonable assumption for gait profiles of multipedal robots. We use the existing variability in velocity of motion profile of the hexapod, to classify objects as dynamic or static.

For each i^{th} cluster, C_i, we define λ_i as the instantaneous rate of events. This rate, λ_i, is a complex function of the size of an object (S_i), depth (d_i), velocity (V_i) and ego-motion (V_{self}); i.e. $\lambda_i = f(S_i, d_i, V_i, V_{self})$. For this reason, we approximate λ_i as

$$f(S_i, d_i, V_i, V_{self}) \approx g(S_i, d_i)|V_i - V_{self}| \tag{3}$$

where $g(S_i, d_i)$ and V_i depend on the scene and V_{self} depends on ego motion of the DVS. $g(S_i, d_i)$ varies smoothly with time since d_i can not vary instantaneously. Our objective is to find all $i : V_i \gg 0$; i.e. find the moving clusters among all clusters. Simplifying notation for $g(S_i, d_i)$ as g_i for L clusters and during the observation interval $(t, t+\delta t)$, we define the total event rate $R(t)$ as

$$R(t) \equiv \sum_{i=1}^{L} \lambda_i(t) = \sum_{i=1}^{L} g_i |V_i(t) - V_{self}(t)|. \tag{4}$$

In Eqn. 4 $\lambda_i(t)$ is the event rate of the ith cluster (C_i) which is calculated by counting the number of events E assigned to that cluster in time interval δt as

$$\lambda_i(t) = \frac{|E \in C_i|}{\delta t} \tag{5}$$

where the operation $|.|$ counts the number of events belonging to a particular cluster. The parameter δt is an important parameter and its effect on final classification performance is discussed in Section III-C.

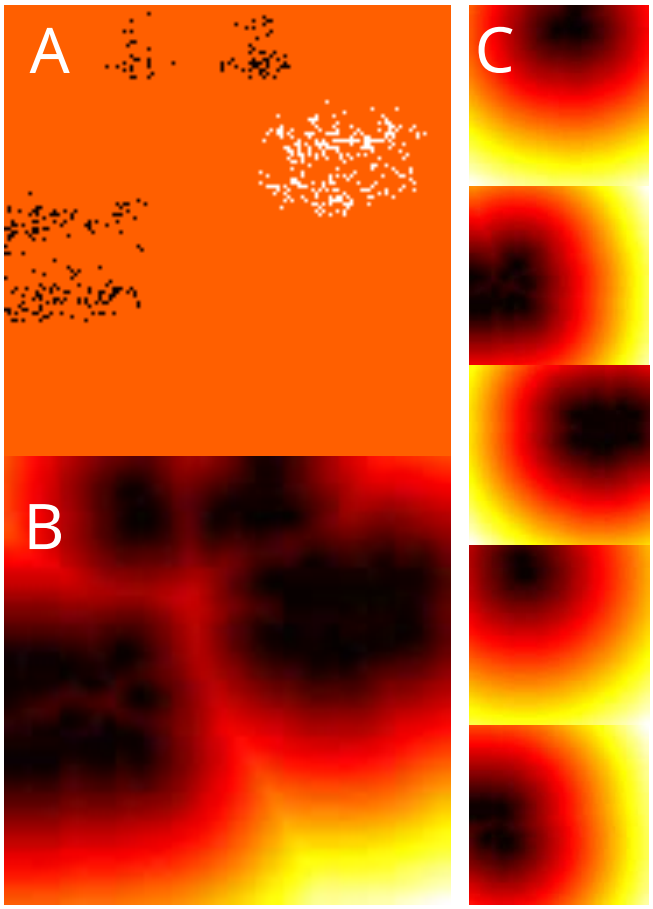

Fig. 2. (A) Snapshot of a scene containing clusters relevant to the leader hexapod (labelled white) where the background scene is also shown (labelled black). (B) A master distance transform map created from this configuration of clusters (increasing values from black to yellow. (C) Individual distance transform maps created for each cluster in the scene. Minimization of these arrays is performed as shown in Eqn. 2, resulting in the map depicted in (B).

Let $m(t)$ denote the moving average of total event rates described by Eqn. 4 for $1s$ windows. For each cluster, we define its fractional event rate $P_i(t)$ as

$$P_i(t) = \frac{\lambda_i(t)}{R(t)}. \tag{6}$$

The operation defined by Eqn. 6 normalizes individual cluster rates by the total event rate, $R(t)$. A plot of this metric for background and leader hexapod is illustrated in Fig. 3. As described earlier, the gait profile of the hexapod is a sinusoid, alternating between low and high phases. Notice that when the velocity of the follower hexapod is in low phase. (shaded bars), the total event rate $R(t)$ is smaller than the moving average event rate $m(t)$. In this interval, we observed that fractional cluster rates associated with the background shows a strong correlation with the total event rate. This is because background events are completely governed by the motion profile of the robot. However, clusters associated with the leader hexapod show a negative correlation with the total

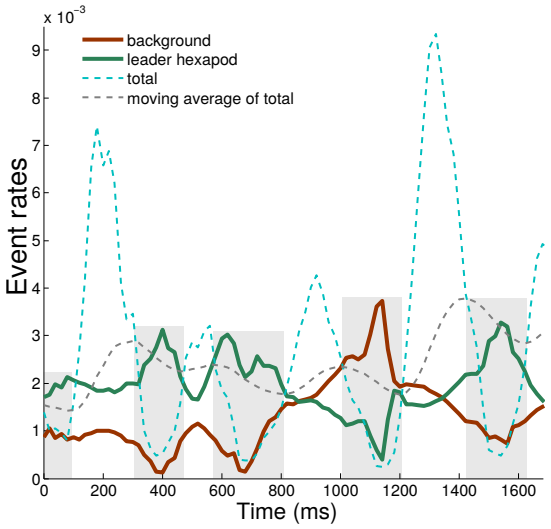

Fig. 3. Rate plots of clusters associated with the leader hexapod (green) and background (red). Total and average event rates associated with the entire scene are shown as dotted lines. Shaded area represents time zones whose data is used for classification.

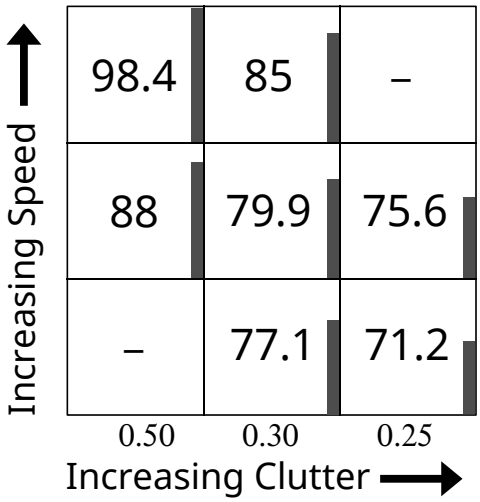

Fig. 4. Each grid represents an experiments under specific environmental conditions. Proportion of foreground objects with respect to background decreases along the horizontal axis. Velocity of the follower hexapod increases along the vertical axis. Dash represents no experiments. The percentage average accuracy is shown in each grid.

rate. Therefore, we classify each cluster as belonging to the leader hexapod or background based on their correlation ρ_i as

$$\rho = \frac{\mathbb{E}[(\lambda_i(t) - \mathbb{E}(\lambda_i(t))(R(t) - \mathbb{E}(R(t))]}{\sigma(\lambda)\sigma(R(t))} \tag{7}$$

where $\sigma(.)$ represents the standard deviation and $\mathbb{E}(\cdot)$ is the expected value. Using this correlation value, the final classification is performed.

$$class_i = \begin{cases} leader\ hexapod & \text{if } \rho_i > 0 \\ background & \text{if } \rho_i < 0 \end{cases} \tag{8}$$

This classification is performed each time the total event rate falls below the moving average rate.

III. Results

Data were collected for a variety of settings: (1) number of background objects present were increased gradually from 2, 3 to 4, (2) hexapod's speed was increased heuristically to achieve maximum event rates for the leader hexapod and (3) distance between the hexapods was changed from 1m to 2m in increments 0.3m to produce different cluster sizes. For each dataset, the follower hexapod moved in a straight line while the leader hexapod moved at a fixed velocity across the sensor's field of view.

A. Accuracy of classification

For each experimental condition, spatiotemporal data were binned at constant time intervals of $20,000\ \mu s$ [27] and the clusters classified as the leader hexapod's were stored for further use. Note that the beginning operation is performed for data analysis only, while the algorithm is event based. The number of frames with correct classification results were compared with the ground truth knowledge of correct cluster for each frame. The classification rate was normalized by the total number of frames for each experimental paradigm to compute the final accuracies (see Fig. 4). In Fig. 4, the relative proportion of foreground with respect to background decreases along the x-axis and the leader hexapod's velocity increases linearly along the y-axis. Each grid point represents the average accuracy for 4 experiments each performed under those conditions.

These results demonstrate that the accuracy of classification is independent of cluster size associated with the two classes: leader hexapod and background. This is because we have assessed the relative normalized correlation between event rates of each cluster, which is independent of spatial dimensions. The findings also suggest that higher speeds of the leader hexapod produces a higher accuracies. This is because with increased velocity of the moving object, the event rates of the leader hexapod's clusters increase in value, while the event rates of background clusters remain unchanged.

B. Success plot

We used a *success plot* performance metric to evaluate our tracking accuracy [28]. At each experimental condition, the data was converted to frames of $20,000\ \mu s$ time intervals. This was followed by ground truth labelling using a bounding box r_g for each frame. If r_t denotes a bounding box computed by our tracking algorithm, the overlap score S is defined as

$$S = \frac{|r_t \bigcap r_g|}{|r_t \bigcup r_g|} \tag{9}$$

where $|\cdot|$ is the number of pixels. This metric calculates a ratio of the number of correctly classified pixels of each frame normalized by total number of pixels associated with the ground truth and algorithm labels. The success plot was calculated by counting the number of frames f within a threshold number of overlapping pixels τ. This result is displayed in Fig. 5, where the horizontal axis represents an

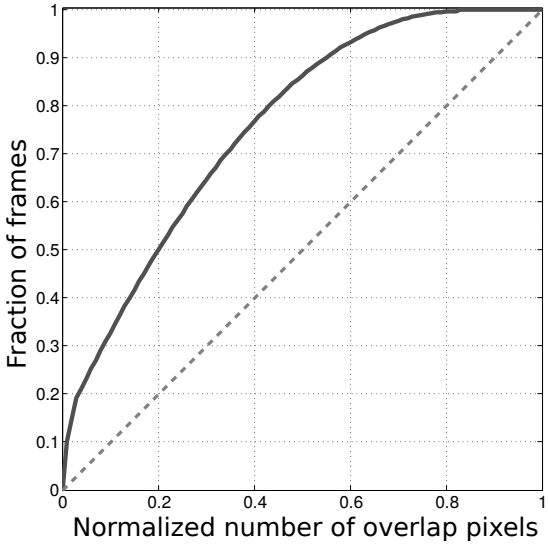

Fig. 5. Success plot of tracking results. Each point on the curve represents the fraction of frames f whose computed bounding boxes lie within a threshold τ along the horizontal axis.

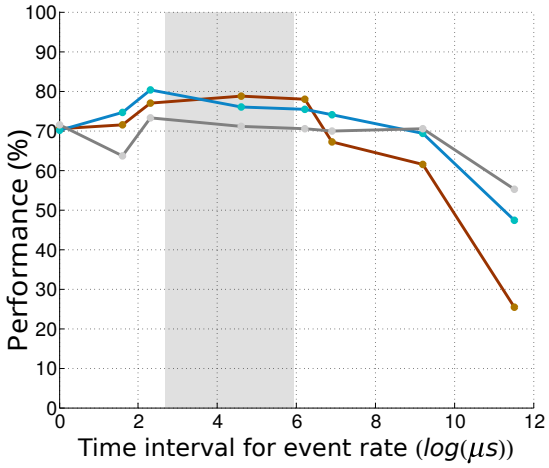

Fig. 6. Variation of tracking accuracy plotted against classification parameter δt, for three different datasets chosen from extreme grid points of Fig. 4. Horizontal axis represents a log (base 2) values of δt parameter. Optimal values for this parameter are found to lie within $[5.6, 64]\mu s$, (shaded region) which is the plateau of maximum performance in classification.

increasing normalized threshold and the vertical axis is the normalized fraction of frames. The area under curve (AUC) is representative of the overall performance of our algorithm with respect to the truth.

C. Performance dependency on classification parameter-δt

The classification parameter δt is the amount of time (μs) needed to compute the event rates for a cluster. Since our classification strategy relies on the inherent motion profile of the hexapod, there is an optimal value for this parameter. This value should be small enough to capture the multipedal robotic locomotion profile and large enough to not cause random rate fluctuations. The effect on classification performance while varying this parameter is shown in Fig. 6 for three datasets under different experimental conditions. The optimal range of values for this parameter deemed as the plateau of maximum accuracy in classification, is shown in the shaded region of the graph.

IV. Discussion

An important feature of our algorithm is the utilization of a robot's motion profile to calculate optimal statistics for classification. This strategy of motor perturbation sensory perception is not uncommon in biology. For example, saccades in vision facilitate fixation and feature enhancement [29]–[31], tympana micro-motions in ears aid with precise source localization [32]. An algorithm utlizing platform motion for classification is an important development since it removes the need for compensation of sensor ego-motion. This is possible because of the spatio-temporal data provided by neuromorphic imagers.

We used event rates of clusters to compute classes. During the low velocity phase of the hexapod's tripod sinusoidal motion profile, event rates are most distinctive. This is inferred from Eqn. 3, since it is known that for the leader hexapod $V_i > V_{self}$ and decreasing V_{self} consequently increases the value of $f(.)$. This phase of motion is shown in Fig. 3 by the shaded regions. During this phase, moving objects elicit motion events whose rates are anti-correlated with the total event rate. However, the static background scene produces event rates in positive correlation to the hexapod's profile. The misclassification of the fourth shaded region in Fig. 3 is due to incorrect clustering due to scene occlusion.

In practical scenarios, clustering using distance transform as described in this study may not produce good results due to the large amount of spatio-temporal data that must be processed in real-time. A distance transform based clustering approach leads to proper assignment of motion events, when the number of clusters formed is low (≤ 10). Other techniques are needed to for applications where the background activity is high.

We approximated the slow velocity phase of the follower hexapod's tripod gait based on intervals where the total event rate fell below the moving average rate. This approach works well for multipedal robots. However, the same concept can be applied to wheeled or aerial robots by utilizing vibrations of their platforms. This would require a more suitable statistical metric, replacing the correlation measure.

A distance transform map of any cluster is an estimate of the probability that a new spike-event belongs to a given cluster. It requires a map to describe the structure of events originating from the cluster at a given timeframe. Using a fixed temporal lag to accumulate events often results in large structural variation. Therefore, using a constant number of events may lead to a more robust representation.

V. Conclusions and Future Study

In this study, we presented an online implementation for robot tracking and following. We have introduced the concept of using motor perturbations for classification and to compute accurate temporal statistics from spatio-temporal

data recorded by a neuromorphic vision sensor. This work contributes to the utilization of temporal statistics to classify motion events and may be applicable to many types of moving objects.

A general model for motion segmentation utilizing this strategy, will be presented in another future study. We will present a real-time algorithm that can utilize vibrations in a robot to classify moving objects from static background scene.

VI. Acknowledgements

We would like the acknowledge Dr. Suresh Golwalkar, Intel Corporation, for providing us the hexapods used in this study.

References

[1] M. Munaro, F. Basso, and E. Menegatti, "Tracking people within groups with rgb-d data," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 2101–2107.

[2] C. Choi and H. Christensen, "Rgb-d object tracking: A particle filter approach on gpu," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, Nov 2013, pp. 1084–1091.

[3] O. Jafari, D. Mitzel, and B. Leibe, "Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 5636–5643.

[4] L. Spinello and K. O. Arras, "People detection in rgb-d data," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3838–3843.

[5] E. Herbst, X. Ren, and D. Fox, "Rgb-d flow: Dense 3-d motion estimation using color and depth," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 2276–2282.

[6] A. Carballo, A. Ohya, and S. Yuta, "People detection using range and intensity data from multi-layered laser range finders." in *IROS*, 2010, pp. 5849–5854.

[7] A. Cherubini and F. Chaumette, "Visual navigation of a mobile robot with laser-based collision avoidance," *The International Journal of Robotics Research*, vol. 32, no. 2, pp. 189–205, 2013.

[8] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "A mobile vision system for robust multi-person tracking," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[9] J. Fabian, T. Young, J. Peyton Jones, and G. Clayton, "Integrating the microsoft kinect with simulink: Real-time object tracking example," *Mechatronics, IEEE/ASME Transactions on*, vol. 19, no. 1, pp. 249–257, Feb 2014.

[10] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 $\times$ 128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 2, pp. 566–576, Feb 2008.

[11] C. Posch, D. Matolin, and R. Wohlgenannt, "A qvga 143 dB dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 1, pp. 259–275, Jan 2011.

[12] D. Reverter Valeiras, G. Orchard, S. H. Ieng, and R. B. Benosman, "Neuromorphic event-based 3d pose estimation," *Frontiers in Neuroscience*, vol. 9, no. 522, 2015.

[13] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S.-C. Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. Ballcels, T. Serrano-Gotarredona, A. Acosta-Jimenez, and B. Linares-Barranco, "Caviar: A 45k neuron, 5m synapse, 12g connects aer hardware sensory-processing-learning-actuating system for high-speed visual object recognition and tracking," *Neural Networks, IEEE Transactions on*, vol. 20, no. 9, pp. 1417–1438, Sept 2009.

[14] D. Crandall, P. Felzenszwalb, and D. Hutten, "Spatial priors for part-based recognition using statistical models," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 10–17.

[15] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on computers*, no. 1, pp. 67–92, 1973.

[16] E. Piatkowska, A. Belbachir, S. Schraml, and M. Gelautz, "Spatiotemporal multiple persons tracking using dynamic vision sensor," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, June 2012, pp. 35–40.

[17] Z. Ni, A. Bolopion, J. Agnus, R. Benosman, and S. Regnier, "Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics," *Robotics, IEEE Transactions on*, vol. 28, no. 5, pp. 1081–1089, Oct 2012.

[18] D. Drazen, P. Lichtsteiner, P. Hfliger, T. Delbrck, and A. Jensen, "Toward real-time particle tracking using an event-based dynamic vision sensor," *Experiments in Fluids*, vol. 51, no. 5, pp. 1465–1469, 2011.

[19] M. Litzenberger, C. Posch, D. Bauer, A. Belbachir, P. Schon, B. Kohn, and H. Garn, "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *Digital Signal Processing Workshop, 12th - Signal Processing Education Workshop, 4th*, Sept 2006, pp. 173–178.

[20] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*. ACM, 2008, pp. 50–56.

[21] F. Rea, G. Metta, and C. Bartolozzi, "Event-driven visual attention for the humanoid robot icub," *Frontiers in neuroscience*, vol. 7, 2013.

[22] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, no. 7553, pp. 503–507, 2015.

[23] T. Delbruck, V. Villanueva, and L. Longinotti, "Integration of dynamic vision sensor with inertial measurement unit for electronically stabilized event-based vision," in *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 2636–2639.

[24] Z. Ni, S.-H. Ieng, C. Posch, S. Régnier, and R. Benosman, "Visual tracking using neuromorphic asynchronous event-based cameras," *Neural computation*, 2015.

[25] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[26] S. F. Frisken, R. N. Perry, A. P. Rockwood, and T. R. Jones, "Adaptively sampled distance fields: a general representation of shape for computer graphics," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 249–254.

[27] 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*. IEEE, 2014, pp. 544–547.

[28] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 2411–2418.

[29] S. Martinez-Conde, S. L. Macknik, and D. H. Hubel, "The role of fixational eye movements in visual perception," *Nature Reviews Neuroscience*, vol. 5, no. 3, pp. 229–240, 2004.

[30] M.-O. Hongler, Y. de Meneses, A. Beyeler, and J. Jacot, "The resonant retina: exploiting vibration noise to optimally detect edges in an image," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 9, pp. 1051–1062, Sept 2003.

[31] M. Rucci, R. Iovin, M. Poletti, and F. Santini, "Miniature eye movements enhance fine spatial detail," *Nature*, vol. 447, no. 7146, pp. 852–855, 2007.

[32] R. N. Miles, D. Robert, and R. R. Hoy, "Mechanically coupled ears for directional hearing in the parasitoid fly ormiaochracea," *The Journal of the Acoustical Society of America*, vol. 98, no. 6, pp. 3059–3070, 1995.