A Spatio-temporal Clustering Method Using Real-time Motion Analysis on Event-based 3D Vision

Stephan Schraml
Neuroinformatics, Safety & Security Department,
AIT Austrian Institute of Technology
Donau-City Strasse 1/5, A1220, Vienna Austria.
stephan.schraml@ait.ac.at

Ahmed Nabil Belbachir, *Member IEEE*
Neuroinformatics, Safety & Security Department,
AIT Austrian Institute of Technology
Donau-City Strasse 1/5, A1220, Vienna Austria.
nabil.belbachir@ait.ac.at

Abstract

This paper proposes a method for clustering asynchronous events generated upon scene activities by a dynamic 3D vision system. The inherent detection of moving objects offered by the dynamic stereo vision system comprising a pair of dynamic vision sensors allows event-based stereo vision in real-time and a 3D representation of moving objects. The clustering method exploits the sparse spatio-temporal representation of sensor's events for real-time detection and separation between moving objects. The method makes use of density and distance metrics for clustering asynchronous events generated by scene dynamics (changes in the scene). It has been evaluated on clustering the events of moving persons across the sensor field of view. Tests on real scenarios with more than 100 persons show that the resulting asynchronous events can be successfully clustered and the persons can be detected.

1. Introduction

Event-based stereo vision [2] aims to duplicate the human vision system in reacting to scene dynamics by generating events including the depth information, using a pair of vision sensor. An event-based 2D Dynamic Vision Sensor (DVS) was introduced in [9] including a set of autonomous self-spiking pixels reacting to relative light intensity changes. Its advantages include high temporal resolution, extremely wide dynamic range and complete redundancy suppression due to included on-chip preprocessing. It exploits very efficient asynchronous, event-driven information encoding, Address-Event Representation (AER) [4], for capturing scene dynamics (e.g. moving objects).

Based on [9], an event-based 3D vision has been realized using a stereo sensor, with a pair of DVSs and a stereo matching algorithm for calculating depth information, which is reported in [2] and [11]. Such a system exploits the on-chip pre-processing offered by the DVS for efficient and real-time scene vision in 3D with regards to two aspects: Firstly, the data volume is reduced as compared to conventional image frame-based stereo systems due to the efficient representation of scene dynamics using on-chip pre-processing of the visual information. Indeed, real-time stereo vision is computationally demanding, implying the allocation of large and costly processing and memory resources. The dynamic vision sensors inherently support on-chip edge detection with a low data volume by means of massively parallel focal plane processing, to allow real-time 3D representation. Secondly, the sensor sensitivity to the relative light-intensity changes allows robustness against illumination conditions. Furthermore, since it is not necessary to integrate light as in frame-based sensors, the sensor is also highly sensitive to scene dynamics in weak illuminations with high temporal response.

Spatio-temporal data processing has been introduced by Fahle [5] and Adelson [1] in the early 80's. However, methodologies for representing low-level spatio-temporal cues and high-level models suitable to explain spatio-temporal evidence are still scarce. The main reasons why joint spatio-temporal processing has not been addressed in detail originates from different factors: (i) digital computers operate using "atomistic" principles, where operations are broken down into sequence of steps and processing is performed independently for each step on discrete data; (ii) common vision sensors provide temporal data sequences in form of distinct images (frames) and (iii) the computational burden imposed by the large amount of data in the space-time volume has been a limitation for efficient operation.

The space-time processing approach is an appropriate strategy for the robust analysis of visual data encompassing dynamic processes such as motion, variable shape, and appearance, whereas traditional frame-based approaches require additional modeling tools (e.g. Markov chains) for dynamical processes. In the development of methodologies for the space-time domain over the last two decades, the research focus has mostly remained on the development of low-level cues, which have incrementally become more descriptive (e. g. transition from simple motion cues to space-time shape).

Those efforts have been invested for automated extraction of relevant information (in space and in time) from image

sequences using frame-based image sensors. Mainly due to the temporally (rate) and spatially (frame) discrete nature of digital image sequences provided by these standard sensing devices, a constant data volume is continuously produced. Such frame-based sensors are not well suited for space-time processing as (i) the data contain substantial temporally redundant information within each frame, and (ii) temporally discrete with coarse resolution (typically 25 frames per second), and (iii) increasing the temporal resolution (thus the amount of visual data) leads to prohibitive computational complexity.

This paper proposes a clustering method for the DVS' events capable of clustering large amounts of continuosly streaming asynchronous data, represented in a spatio-temporal domain and its application for real-time object detection in real surveillance scenarios towards a compact remote stand-alone system. Besides the stereo sensor and the processing unit, the developed system also includes this event-based clustering algorithm, which is demonstrated for surveillance applications. The paper is structured as follows: Section 2 provides a brief review of the architecture of the event-based 3D vision system. The clustering method using the sensor data is presented in Section 3. Section 4 describes evaluation results on synthetic data as well as real-world recordings. A summary is provided in Section 5 to conclude the paper.

2. Dynamic Stereo Vision Sensor

This section briefly describes the existing dynamic stereo vision sensor reported in [2] and [11] including data examples generated by the system. The system, including the sensor board, DVS chip and DSP board, is depicted in Figure 1. It includes two DVSs as sensing elements [9], a buffer unit consisting of a multiplexer (MUX) and First-In First-Out (FIFO) memory, and a digital signal processor (DSP) as processing unit.

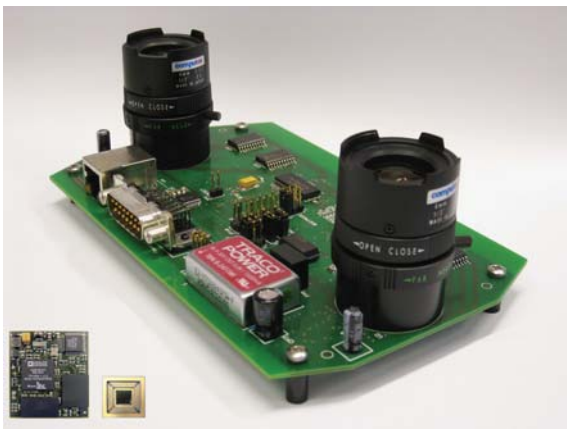

Figure 1: Dynamic stereo vision system device. In the lower left corner the DSP Bf537 and the sensor chip are shown. The DSP is mounted on the back of the board.

The DVS consists of an array of 128x128 pixels, built in a standard 0.35µm CMOS-technology. The array elements (pixels) respond to relative light intensity changes by instantaneously sending their address, i.e. their position in the pixel matrix, asynchronously over a shared 15 bit bus to a receiver using a "request-acknowledge" 2-phase handshake.

Such address-events (AEs) generated by the sensors arrive first at the multiplexer unit. Subsequently, they are forwarded to the DSP over a FIFO. The DSP attaches to each AE a timestamp at a resolution of 1ms. The combined data (AEs and timestamps) are used as input stream for 3D map generation and subsequent processing.

Figure 2 depicts a space-time representation of one DVS' data, resulting from a two persons crossing the sensor field of view. The events are represented in a 3 D volume with the coordinates x (0:127), y (0:127) and t (last elapsed ms), the so-called space-time representation. The bold colored dots represents the events generated in the recent 16 ms. The blue and red dots represent spike activity generated by a sensed light-intensity increase (ON-event) and decrease (OFF-event) resulting from the person motions, respectively. The small gray dots are the events generated in the elapsed 1.733 seconds prior to the recent 16ms. These highlight the event path in the past 1.733 sec of the moving persons, which is an ideal basis for clustering and tracking in space and time.

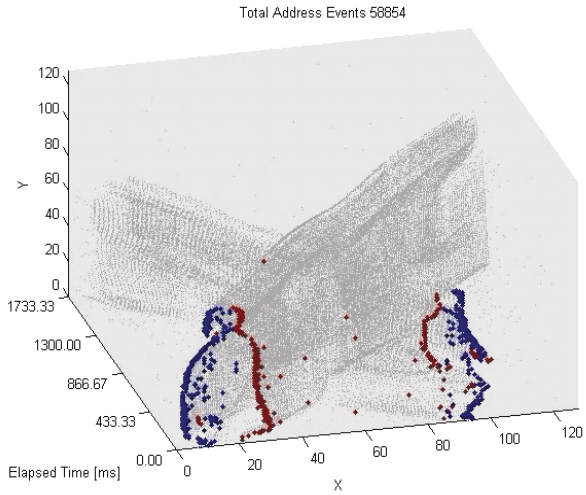

Figure 2: Event representation of scene dynamics (2 persons crossing the field of view) in a space-time domain using 1 DVS.

A description of the algorithm for real-time depth estimation is given in [2] and [11]. Figure 3 shows an example of a visual scene imaged by a conventional video camera (top left) and its corresponding AEs using a pair of

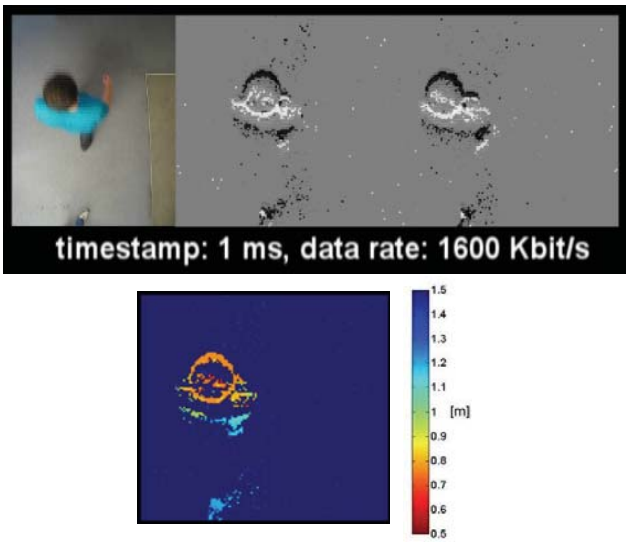

Figure 3: Still image of a person from a conventional video camera (top left); the corresponding AE a pair of dynamic vision sensors (top right); resulting event "sparse" depth map (bottom).

DVSs (top middle and top right) rendered in an image-like representation. The white and black pixels represent spike activity generated by a sensed light-intensity increase (ON-event) and decrease (OFF-event) resulting from one persons motions, respectively. The gray background represents regions with no activity in the scene. The non-moving parts in the scene do not generate any data. The processing unit (DSP) embeds event-based stereo vision algorithms, including the depth generation or the so-called *sparse depth map*. The resulting sparse color-coded depth map of the scene depicted in Figure 3(left) is provided at the bottom in Figure 3.

3. Real-time Spatio-temporal Clustering algorithm

The 3D DVS continuously and asynchronously generates events as reaction to moving objects crossing the sensor field of view. The objective of the proposed clustering method is to group together events belonging to the same moving object. It is therefore assumed that objects are characterized by following statements: (i) an object is determined by a set of address-events generated by an individual real-world object, a person for instance. (ii) Objects can be of arbitrary but limited size in x-y at any time. (iii) The distribution of AEs within an object may be sparse, as AEs are mainly generated by the edges of an object. (iv) Objects evolve in time, as they are moving through the sensors field of view.
It was of great interest for us having a clustering method which can deal with large amounts of asynchronously streaming data in real time at the same time demanding little memory and processing power, since using the DVS

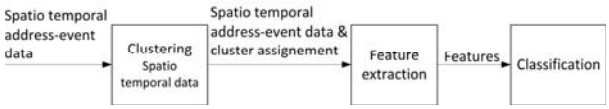

Figure 4: Overview of the spatio-temporal processing steps and data flow for demonstration use in a recognition application.

a single object can result in generation of starting from 10 thousands AEs per second. For analysis the clustered objects' events can be further used for high-level computer vision tasks like recognition and classification. Figure 4 provides an exemplary overview of processing steps for use of the clustering algorithm in a recognition application. The feature extraction and recognition application are out of the scope of this work such that only the spatio-temporal clustering is described.
We use a combination between density-based [10] and distance based clustering for robustness. Similarity between AEs is given by a distance function f(Cluster,AE) calculating the distance of the AE to the cluster center and expressed in the assignment of the AEs to the same cluster.
The metric used is the projected Manhattan distance in space-time (x,y,t) between the pixel coordinates of the AE and the cluster center to a 1-dimensional vector. The cluster center is defined as the moving average of (x,y) coordinates of the assigned AE's. The clustering input data is a stream consisting of the temporal sequence of AEs having (x,y) coordinates, their polarity "p" (OFF or ON), the timestamp "t" and the reconstructed depth "z". The event stream is neither stored for iterative processing nor grouped in frames. For each AE, a cluster assignment will be evaluated once; afterwards, the AE will be discarded. The actuality of a cluster is given by the timestamp of the latest assigned AE. The clustering method is comprised by following steps, which are performed for every AE:

1. Update the density matrix of AEs
2. Update the radial dilation of all existing cluster
3. Assign AE to a cluster
4. Update the properties of the selected cluster

The following definitions characterize the proposed clustering method:

- **Density matrix:** it provides the frequently generated AEs in the x,y,z coordinate system. In figure 5 (right image) an example of a density matrix is shown as color coded image, where the light blue shows low AE density and the red shows high AE density.

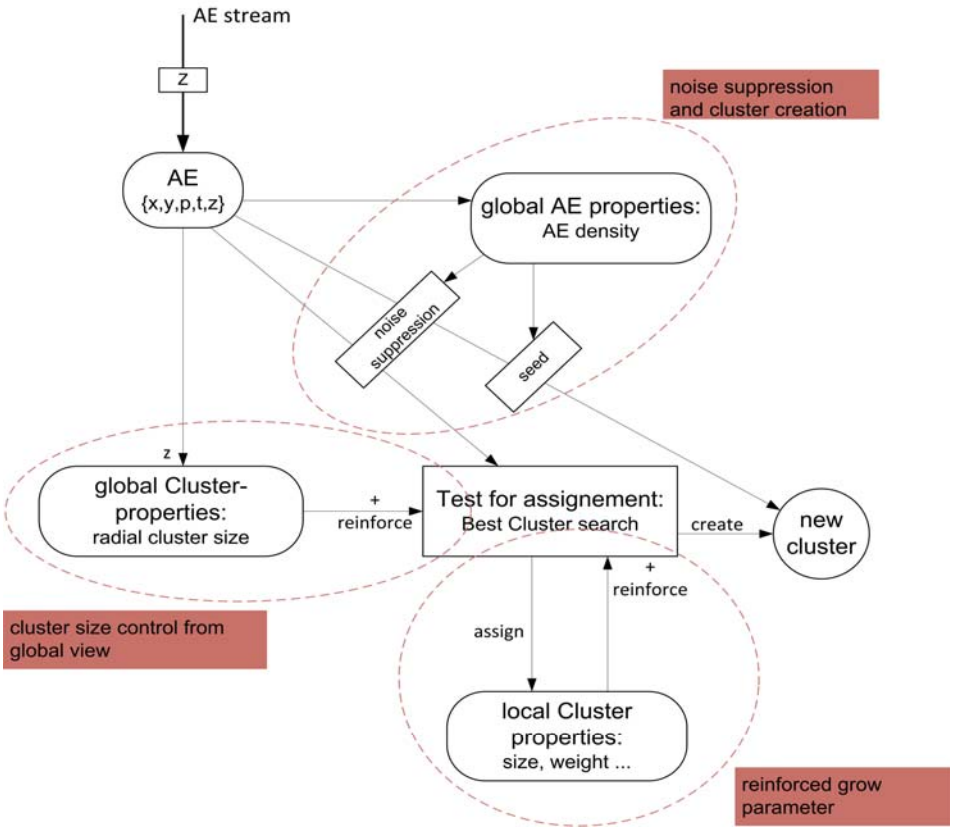

Figure 6: Overview of the clustering method.

- **Radial cluster dilation**: it provides a measure of the object dimension. It is determined by the distance between the cluster center and the location of a point where the radial cluster density drops beyond a certain threshold. The radial cluster density is the projected AE density related to a cluster center.

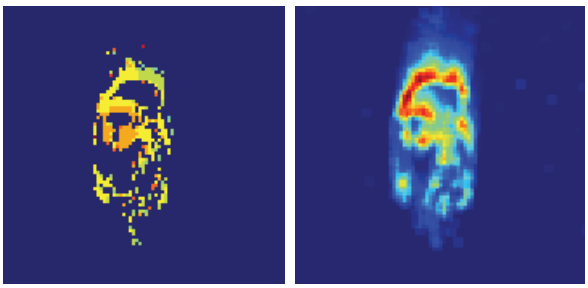

Figure 5: Rendered image with events accumulated for 20 ms including the depth information (left image) and density matrix (right image).

For every point in the sensor array/ every AE, we can calculate the distance $\Delta \boldsymbol{P}$ to every cluster center in the direction of x and y.

$$\Delta \boldsymbol{P}\text{: } \{dx; dy\} = (x_{cluster} - x_{ae}; y_{cluster} - y_{ae}) \qquad (1)$$

A direction-independent distance value R will be evaluated regarding the expected form of the cluster. This value is the radial distance value. The form of the cluster is defined through the dilation boundaries DX and DY forming a hexagon, DX and DY can be given as parameters.

$$R_{(\Delta P)} = \frac{|dx|}{DX} + \frac{|dy|}{DY} + \max\left(\frac{|dx|}{DX}, \frac{|dy|}{DY}\right) \qquad (2)$$

The AE distances are used to calculate the AE frequencies in relation to the radial distances

$$\Phi_{(R,C)} = \sum_{AE} \frac{AE_z}{R_{(AE,C)}} \qquad (3)$$

Where Φ is the radial density of the cluster C and distance R as a sum of density distances of all AEs at distance $R_{(AE,C)}$. The radial density can grow within the object and drop outside the object as long as the cluster center is located in the object center. In this case, the cluster radial dilation (object size) can be determined (using a certain threshold).

3.1. Assignment Policy

An overview of the clustering method is provided in figure 6. The cluster assignment policy is based on information derived (i) from global AE and cluster properties (with all AEs included) consisting of global cluster properties for cluster size calculation and global AE properties for noise suppression and (ii) from local cluster properties (only own cluster's AEs included) like size, weight and number of assigned AEs.
AEs may be filtered first by their calculated distance z in order to remove non relevant information (application-dependent) like e.g. shadows. Every generated AE will then imply the update of the density matrix and of the radial density of every cluster. Notice that processing one AE results in strengthens the size of every existing cluster. While clusters far away from the AE will be slightly affected due to the consequently large radial distance, clusters near to the AE will get more influenced. The AE assignment is calculated according to local clusters properties, which are detailed as follows:

- If local density is low, the AE is considered as noise event and therefore discarded.
- The radial distance to every cluster is calculated according to formula (1) and (2).
- The propagated strength (influence) of each cluster on the AE is evaluated. The evaluation function depends on the AE's distance, the radial dilation of the cluster and the weight of the cluster. This latter is calculated from the sum of all so far assigned AEs (number of AEs in a cluster). A cluster will be neglected when the radial distance exceeds the maximum object size.
- The AE will be finally assigned to the most influencing cluster updating its properties. The global cluster dilation and selected local properties affect the cluster assignment, i.e. higher values (in size, weight ...) increase the probability that an AE is assigned to a cluster. In case the AE does not fit to any existing cluster, a new cluster can be created when the local density at the AE's location exceeds a dedicated threshold.

3.2. Description of algorithm characteristics

The individual steps of the clustering method (cluster creation and AE assignment) are further detailed in this subsection as follows:

- ***Cluster definition:*** the cluster represents a bulk of frequent AEs, which have density-based interrelationship around a center. By that, the stability of a cluster increases when the density of AEs in the object center is the maximum.
- ***Pair-wise similarity:*** depends on the cluster strength within the AEs locations. The similarity is not explicitly calculated but derived from the assignment to a common cluster.
- ***New cluster creation:*** a cluster is created whenever an AE could not assigned to a cluster, because it lies outside the maximal size of all existing clusters, and the local density at the AEs location exceeds a threshold.
- ***Terminating of a cluster:*** A cluster can be removed whenever it is not timely actual anymore i.e. no new AEs were assigned to it for a dedicated time period.
- ***Temporal continuity:*** The temporal continuity of a cluster is ensured when continuously actual AEs are assigned to it, that is the case of moving objects.
- ***Stability:*** a cluster is stable whenever its center is close to the barycenter of the assigned AEs. This is the case of AEs generated from moving persons, but not valid in case of e.g. umbrellas such that only events on the umbrella contours "boundaries" are generated.
- ***Parameters:*** there are four parameters used for the clustering. Two thresholds for the clustering creation and noise suppression and two parameters for the dilation in x and y axes. The dilation parameters define the size of the cluster outside its center. These latter have to be chosen with respect to the expected size of the observed objects (like person, vehicle…etc) and should not be greater than twice the size of the smallest object. The cluster creation threshold has to be chosen to allow clustering of objects with a low AEs density.
- ***Parameter sensitivity:*** The clustering algorithm is not sensitive to the two parameters related to the noise suppression and the cluster creation. However, the dilation parameters have to be adequately chosen with regard to the object size, which is also depending of the sensor mounting position and the distance between the sensor and the objects.

As the events clusters are computed by a single-pass-method, the processing demands of the algorithm can be kept low. There is no reassignment or rearranging of AEs or clusters. By that we achieved that the complexity is proportional to the number of events n and the number of existing clusters k, "O(n*k)". Therefore, this method ensures fast calculation and assignment of events to clusters and to be suitable for large data sets and embedded systems. We could prove that this clustering approach is able to run in real time. In a live demonstration we achieved a performance of clustering of about 100kAE/s using the DVS device. While running on a PC the algorithm can handle much more AEs.

4. Experimental Results

We conducted a number of experiments with synthetic stimulus and with real-world data for the evaluation of the event-based spatio-temporal clustering method.

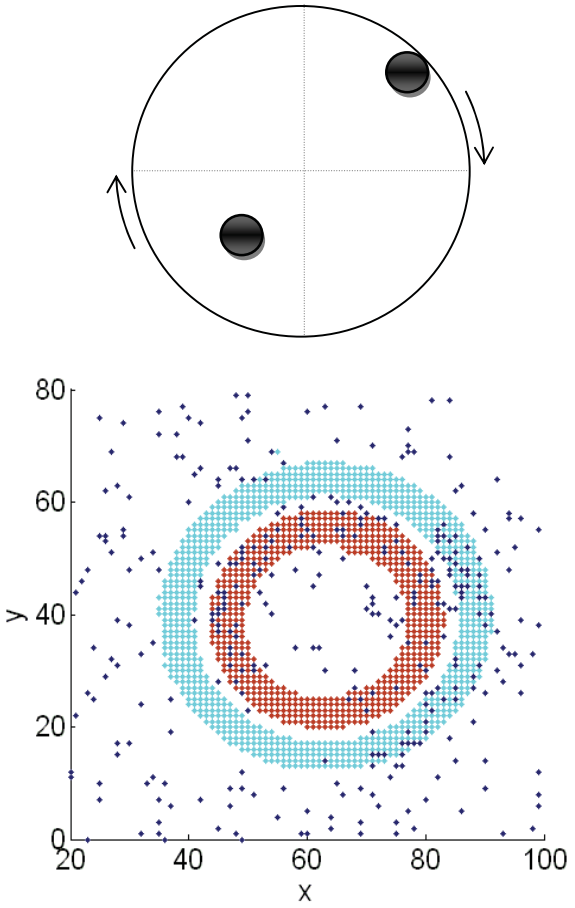

Figure 7: Synthetic stimulus. Test pattern (top) and generated AEs (bottom).

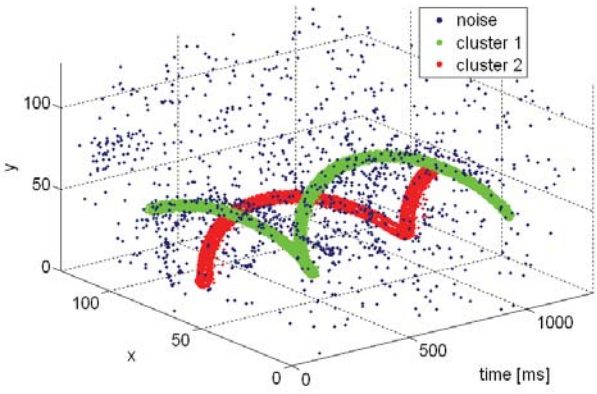

Figure 8: Clustering result of synthetic input for about 1.2s, containing 10.000 events.

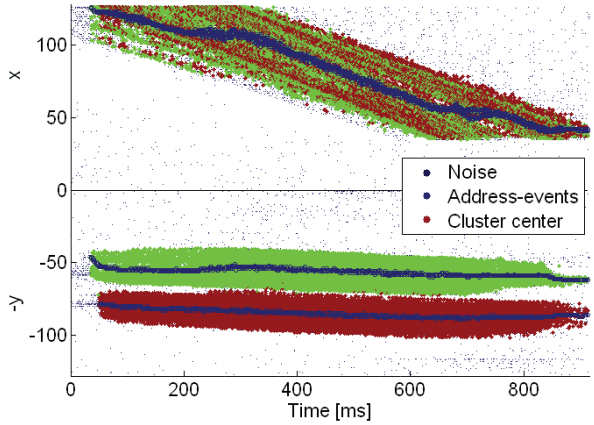

Figure 9: Illustration of tracking two riding cyclists during their passage across the sensor FOV (x; y; time)

Figure 7 shows a test pattern consisting of a rotating disc with two black 2D discs, rotating at 1rps for the synthetic stimulus (top) and the corresponding generated AEs, colored according to ground truth (with two clusters, outer and inner ring, and noise), shown in x-y plot (bottom). The result of our clustering method using this input is shown in an x-y-t plot in figure 8. The AEs generated from the two black circles on a rotating disc were correctly clustered as separate objects (shown in different color) and noise has been successfully identified.

Real-world test scenarios were collected with a total of 128 passages (82 riding cyclists; 26 pedestrians, 13 walking cyclists and 7 pedestrians with umbrellas). Figure 9 shows generated AEs from two cyclists crossing the sensor field of view. The image from a conventional camera is shown in the top where the bottom image depicts the generated AEs, represented according to their x-coordinate (top) and y-coordinate (bottom) in function of time. The depth information was mainly used to remove outliers and cast shadow of the object. It can be noticed that both object has been separated and tracked along their passage duration, especially on the y-axis.

Figure 10 illustrates another example of a real-world scene with two cyclists and one person crossing the sensor field of view. In Figure 11 the corresponding generated events and intermediate clustering results (corresponding to figure 10, middle) are shown. The outputs are: collected raw data accumulated for about 40ms and rendered in an image-like representation (top left), the corresponding

density matrix (top right), the calculated track of each cluster, derived from continuing observation of the cluster center position (bottom left) and residual cluster assignment (bottom right). The light colored circles on the bottom-right image indicate the clusters size.

Figure 10: Images of a scene showing two cyclists crossing field of view from up to down while a person is walking in the opposite direction (from left to right).

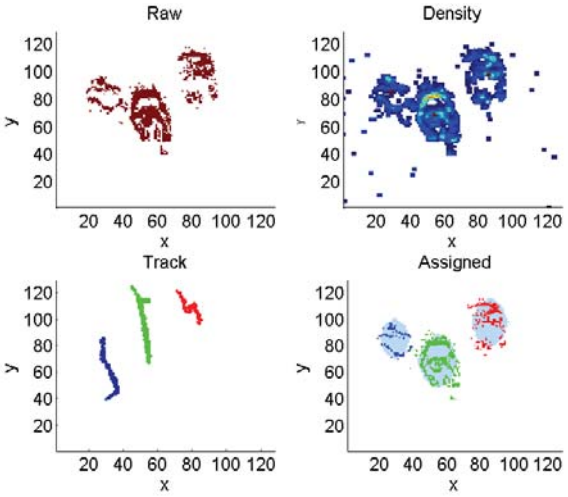

Figure 11: Output generated from the clustering method, clustering AEs generated by the scene shown in Figure 10 (middle image).

5. Conclusions and Outlook

This paper presents a real-time spatio-temporal clustering method for asynchronously generated object events towards stand-alone and compact (biologically-inspired) event-based 3D vision system. By combining a density approach and a distance approach, the clustering method was able to assign individual events to the object generating it. The results on synthetic stimulus (rotating disc) and real-world scenarios (moving person, riding cyclists) with more than 100 objects have shown that the method can be useful for real-time detection of moving objects. A validation on a larger data set in challenging (crowded) scenarios is the next investigation step.

6. References

[1] E.H. Adelson and J.R. Bergen, "Spatiotemporal Energy Models for the Perception of Motion," Journal of the Optical Society of America A,2, pp. 284-299, 1984.

[2] A.N. Belbachir, "Smart Cameras", Springer New York, November 2009.

[3] V. Chan, C. Jin and A. van Schaik, "An Address-Event Vision Sensor for Multiple Transient Object Detection," in IEEE Transactions on Biomedical Circuits and Systems, vol. 1, issue 4, pp. 278 – 288, Dec. 2007.

[4] E. Culurciello, R. Etienne-Cummings, K. Boahen, "Arbitrated address event representation digital image sensor," IEEE Elect. Letters, vol. 37, pp. 1443–1445, 2001.

[5] M. Fahle and T. Poggio, "Visual Hyperacuity: Spatio-temporal Interpolation in Human Vision," Proceedings of the Royal Society of London B, 213, pp.451-477, 1981

[6] A. Fusiello, E. Trucco and A. Verri, "Rectification with Unconstrained Stereo Geometry", in Proc. of the British Machine Vision Conf., pp. 400-409, BMVA Press, 1997.

[7] R.Greene-Roesel, M.C. Diógenes, D.R Ragland and L.A. Lindau, "Effectiveness of a Commercially Available Automated Pedestrian Counting Device in Urban Environments: Comparison with Manual Counts", Transport Research Board Annual 2008 Meeting, 2008

[8] G. Grubb, A. Zelinsky, L. Nilsson and M. Rilbe, "3D Vision Sensing for Improved Pedestrian Safety," in Proceeding of the IEEE IVS, pp. 19-24, 2004

[9] P. Lichtsteiner, C. Posch and T. Delbrück, "A 128×128 120dB 15us Latency Asynchronous Temporal Contrast Vision Sensor," IEEE JSSC, vol. 43, pp. 566 - 576, 2008.

[10] J. Sander et al., "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications," Journal of Data Mining & Knowledge Discovery, Springer, vol. 2, pp. 169-194, 1998.

[11] S. Schraml, A.N. Belbachir, N. Milosevic and P. Schoen, "Dynamic Stereo Vision for Real-time Tracking," in Proc. of IEEE ISCAS, June 2010.

[12] S. Schraml, N. Milosevic and P. Schön, "Smartcam for Real-Time Stereo Vision - Address-Event Based Stereo Vision," in P. of Computer Vision Theory and Applications, INSTICC Press, pp. 466 – 471, 2007.