Density-based rare event detection from streams of neuromorphic sensor data

Csaba Beleznai, Ahmed Nabil Belbachir
AIT Austrian Institute of Technology GmbH
Donau-City-Straße 1
1220 Vienna, Austria
Email: csaba.beleznai@ait.ac.at

Peter M. Roth
Institute for Computer Graphics and Vision
Graz University of Technology, Graz, Austria
Email: pmroth@icg.tugraz.at

***Abstract*—Discovering frequent and rare spatio-temporal patterns in large amounts of streaming visual data is of great practical interest since it allows for automated applications of activity and surveillance analysis. In this paper we present a computationally efficient and memory preserving clustering scheme which uses streaming input from a stationary-mounted neuromorphic camera and performs density-based clustering in a high-dimensional feature space. The clustering scheme can treat arbitrarily shaped complex distributions and employs an intuitive density-based criterion to assign previously unseen samples to categories of *frequently observed* and *rare*. The spatio-temporal structure of neuromorphic video is encoded into sparse binary features, which allow for fast Hamming distance based neighborhood analysis in the feature space. Moreover, data sparsity brings advantages with respect to memory-efficient transmission and storage of the learned statistical model when used within a camera network. We present rare event detection results in a multiple-day neuromorphic data sequence and discuss strengths, failure modes and possible extensions of the proposed method.**

I. Introduction

Analyzing and recognizing spatio-temporal patterns in streaming visual data is of great practical interest given the recent explosive growth in the quantities of networked digital video. Many applied domains such as visual surveillance, ambient assisted living and activity-oriented video analysis seek to learn levels of normality and distinguish between frequently and rarely observed spatio-temporal patterns. However, the analysis task encompasses several challenges. Video streams exhibit a vast richness of information due to the inherent variability in the data thus large data amounts are required to obtain meaningful statistical models. Large data quantities are associated with a substantial computational cost and large memory footprint. Furthermore, structuring and modelling the data distribution easily become nontrivial since data typically reside in high-dimensional feature spaces. The temporal characteristics of streaming data represents another challenge calling for computational techniques capable to build statistical models in an incremental and adaptive manner.

According to a data-oriented view the goal of rare or unusual event detection is to discover the principal modes in the distribution of video data and label incoming data samples falling into the categories of (i) learned modes (inliers) or (ii) not explained by any existing model (unusual event). Core scientific questions involve how to represent the complex spatio-temporal structure of videos, how to build informative models of the multi-modal, multi-scale distributon of aggregated features and how to unambigiously assign labels to previously unseen data samples, while meeting computational and memory requirements.

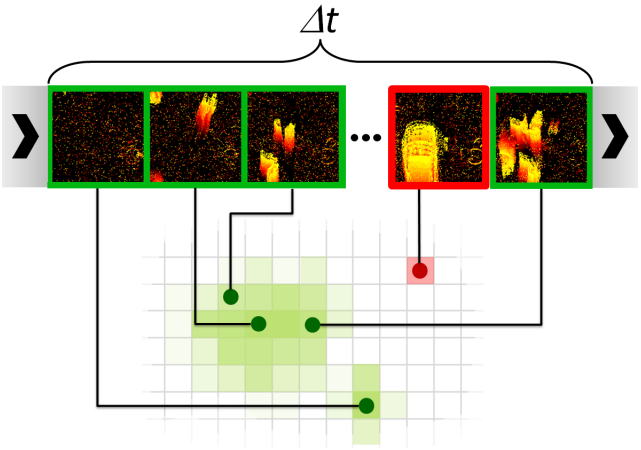

Fig. 1. Schematic illustration of our proposed approach: Frequently observed (green) and rare (red) events are discovered within a moving temporal window of Δt length in neuromorphic data streams (best viewed in color). The top part shows selected spatio-temporal patterns generated by moving object gradients for a top-view camera geometry. The individual motion patterns are shown as color-coded motion history [1] images for easy interpretation. The bottom illustration shows a density-based clustering scheme delineating clusters of normality (empty scene, walking pedestrians) and an unusual event (car in a pedestrian zone).

There is a large body of work focusing on unusual event detection and touching on the above core topics. When considering the employed representations, strategies range from representing objects, entire frames to describing local spatio-temporal volumes. Explicit object tracking often forms the basis for representing events, such as in [2] and [3] where static models of normality are learned. Time-aggregated histogram-of-oriented-gradient (HOG) descriptors representing individual frames are demonstrated in [4], [5] to detect previously unseen static states (e.g. an object at an unusual location) in low-frame rate videos. Single frame representations are aggregated while preserving temporal ordering in [6]. Dynamic event encoding local features of Haar wavelets [7], oriented gradients [8], optical flow histograms [8] and local binary patterns [9] are popular choices of representation. Salient dynamic patterns are

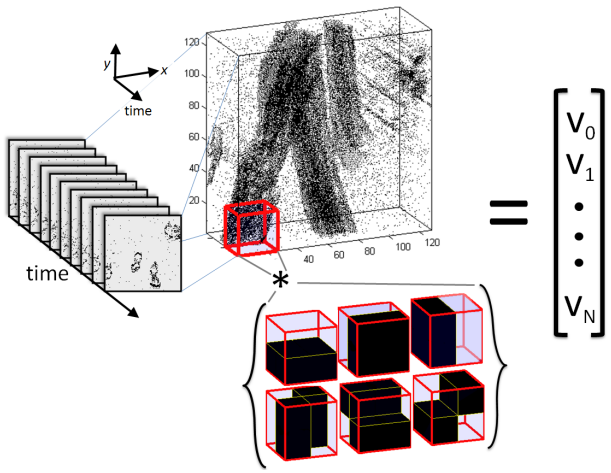

Fig. 2. Computation scheme of the employed sparse spatio-temporal features $\mathbf{v} = (\mathbf{v_0}, \ldots, \mathbf{v_N})$. Convolution of the input space-time data with a set of simple Haar wavelet features yields a local structural representation which is used in an aggregated form to represent a temporal slice.

learned by convolutional learning in [10]. Video segmentation resulting in space-time segments is used in [11] in a graph structure learning framework to represent single and multiple associated events from video. Clustering in high-dimensional feature spaces also represents a challenge. Parametric schemes such as k-means clustering and Gaussian Mixture Models generate spherical and ellipsoidal cluster shapes and they require the number of clusters *a priori*, thus typically resulting in under- or oversegmented partitioning of the feature space. Iterative schemes such as k-means and mean shift [12] are computationally intensive or an incremental computation is not straightforward.

We see two primary challenges in the context of rare dynamic event detection in a camera network. The vast amount of data of the spatio-temporal image space imposes a challenge since its transmission and computational analysis call for high network bandwith and substantial computational power. Another challenge is represented by the computationally efficient incremental clustering task in high-dimensional feature spaces. Existing methods [4], [5] capturing the global structure of image content employ agglomerative schemes for the incremental clustering task. The stopping criterion used by such hierarchical techniques is a very sensitive parameter governing the granularity of the resulting solution.

To address these challenges we propose two contributions. We present the use of neuromorphic dynamic vision sensors (DVS) [13] to efficiently capture scene dynamics in a networked setting, since compared to image-based sensors, they involve a substantial reduction of the data volume having an on-chip background subtraction. DVS chips [14], [15] feature massive parallel pre-processing of the visual information in on-chip analog circuits and stand out for their excellent temporal resolution, high dynamic range and low power consumption. Our second contribution is a weakly parametric density-based clustering method following the adaptive DBSCAN scheme [16] and using a hash-table representation for performing batch and incremental clustering steps in a computationally efficient manner. Clustering is performed on the DVS data which is a spatio-temporal sequence of digital pulses (events) generated by pixels independently responding to contrast variations.

The paper is organized as follows: Section 2 describes the overall concept of the proposed rare event detection method and provides details on the individual algorithmic components. Section 3 presents and discusses experimental results. Finally the paper is concluded in Section 4.

II. Method

A. Outline of the method

The proposed approach extracts large quantities of binary space-time descriptors in form of local Haar filter responses from data slices of time-aggregated neuromorphic data. Each descriptor vector is mapped to a point in a high-dimensional Hamming space and in this manner feature densities are estimated on a discrete grid. We adopt a density-based clustering scheme based on the adaptive DBSCAN algorithm [16], which allows for computationally efficient batch and incremental clustering and outlining densely and sparsely populated regions in the feature space. All employed parameters of the individual algorithmic stages are listed in Table I.

B. Algorithmic details

Input data: The neuromorphic data input consists of sparse binary data where pixels are set at image locations where a change is detected. First, these sparsely populated frames are aggregated in time (see Figure 2, left). A spatio-temporal volume with temporal thickness N_t is sampled at discrete $N_x \times N_y$ locations along the $x-y$ plane by non-overlapping rectangular analysis cuboids (a single cuboid is shown in red in Figure 2).

Spatio-temporal features: Data within the individual analysis cuboids is convolved with a set of simple space-time Haar-wavelet filters (shown in Figure 2). The local set of Haar filter responses $D_z(x_i, y_j, t_k)$ at a given location $\{x_i, y_j, t_k\}$ for $z = 1, \ldots, N_f$ different filters is used to compute a 3-valued feature vector entry v:

$$v^z_{i,j,k} = \begin{cases} 1, & \text{if } D_z(x_i, y_j, t_k) \geq \theta \\ 0, & \text{if } |D_z(x_i, y_j, t_k)| < \theta \\ -1, & \text{if } D_z(x_i, y_j, t_k) \leq -\theta \end{cases} \quad (1)$$

The raw data contains much noise therefore to render the features more resistant with respect to noise, we applied a

TABLE I
Parameters of our approach

Parameters	Value	Description
N_x, N_y	5	# of sampled locations
N_t	20	# of frames in a time slice
N_f	6	# of Haar filters
θ	102	noise threshold
γ	50%	temporal slice overlap rate

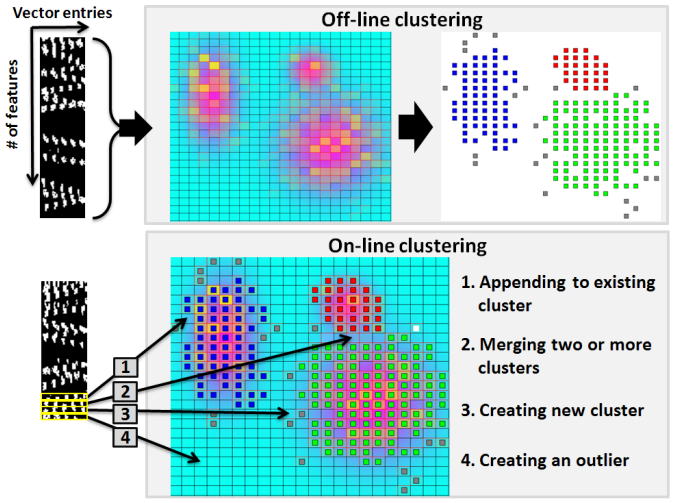

Fig. 3. Illustration explaining the clustering step. Top: all feature vectors are mapped onto the Hamming space in a batch manner and densities of individual Hamming space entries (cells) are updated. By performing density-driven clustering each cell with a non-zero density is assigned to a cluster or becomes an outlier (gray rectangles). Bottom: Given an existing cluster structure in the feature space incremental updates by single incoming features can generate four assignment cases.

threshold θ. The sensor noise level was estimated using a set of frames equivalent to several hours containing no objects and the threshold was set accordingly (see Table I). The overlap rate between two neighboring temporal slice is γ.

We apply max-pooling [17] to transform the joint feature representation into a new one which preserves important information while discarding irrelevant detail. Max pooling is particularly well suited to the discrimination of features that are very sparse [17] and achieves better robustness with respect to noise and clutter. Accordingly, we encode a local feature on two bits by setting the first bit if any of the local features $\left\{v^z_{i,j,k}\right\}_{z=1..N_f}$ is set to 1, and the second bit if any of the local features is set to -1. The resulting binary feature vector $\mathbf{v}$ aggregated for the entire time slice has a dimensionality of $N_x \times N_y \times 2 = 50$ (see Figure 2). Structured indexing techniques, such as Locality Sensitive Hashing [18] and compression-based indexing [19] allow for efficient approximate neighborhood queries within the Hamming space.

Density-driven clustering: We seek to cluster an evolving data stream, therefore, the clustering algorithm must meet following requirements: (i) no assumption on the number of clusters, since in a continuously changing data distribution the number of cluster will likely to vary; (ii) ability to discover clusters with arbitrary shapes and (iii) ability to model and discover outliers not meeting constraints of a given cluster. The adaptive variant of the DBSCAN algorithm [16] and its hash table based variant (D-Stream [20]) describe simple clustering schemes meeting these constraints. At the same time they are also computationally efficient since they need to examine each input feature entry only once. Both techniques perform a density-based region growing in arbitrary dimensions resulting in "connected components" where connectedness is given by the underlying density.

Algorithm 1 Streaming data clustering

```
t_i = 0;
initialize an empty hash table cell_list;
while data from stream is available do
    read incoming feature v = (v_1, v_2, ..., v_N);
    determine hash key from v addressing a given cell c;
    if (c is not in cell_list) insert c into cell_list;
    update values stored in c;
    if t_i == Δt then
        call offline_clustering(cell_list);
    else
        if t_i mod Δt == 0 then
            call online_clustering(cell_list);
        end if
    end if
    t_i = t_i + 1;
end while
```

By adopting the main ideas from these density-based clustering schemes we perform the following steps. We introduce two parameters, ϵ and N_{min}. When performing *off-line (batch) clustering* for each data point $\mathbf{v}_i$ the neighborhood of a given radius ϵ is examined and the number of data points, i.e. the cardinality of the neighborhood is determined. If the cardinality (local density) exceeds N_{min} then a new cluster is generated and all density-connected data points are assigned to this cluster label. If the cardinality does not exceed the density threshold N_{min}, the examined data point is labeled as an outlier. The *on-line clustering* scheme proceeds analogously, as illustrated in Figure 3. When a new feature is inserted, four possible assignment cases can be distinguished: (i) The entry is close to an existing cluster, thus obtains its label. (ii) The entry is between two or more clusters. In such cases, label collision is treated by noting label equivalences using the Union-Find algorithm and performing cluster relabeling. (iii) The data insertion is within the ϵ-neighborhood of one or more outliers (but no valid clusters are present), and the critical local density N_{min} is reached to create a new cluster. (iv) The data entry is inserted at a location where no or few previous entries exist, the local density is less than N_{min}, and the data point is labeled as an outlier.

After describing the off-line and on-line clustering steps, we provide details on our clustering scheme combining the off-line and on-line components. The overall algorithmic steps are described in Algorithm 1. Each feature vector $\mathbf{v}$ is inserted into a hash table, and upon insertion the following set of values is inserted or updated: $(N_{pts}, d_{loc}, flag_p, ClustLabel, t_{ins})$. N_{pts} denotes the number of insertions up to the current time instance, d_{loc} is the cardinality of the local ϵ-radius neighborhood and $flag_p$ is a flag indicating a binary processing status. $ClustLabel$ is the cluster index, which in case of an outlier takes the value -1. t_{ins} denotes the time (frame index) of insertion. The information given by t_{ins} can be used to retrieve the given feature or image frame representing the

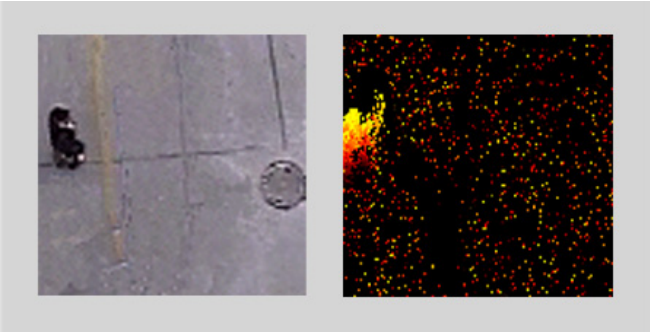

Fig. 4. Left: Sample view (RGB-image) of the observed scene. Right: Corresponding motion history image computed from neuromorphic data.

entry. Furthermore, by checking t_{ins} at each data insertion a removal (forgetting) mechanism can be easily implemented, for example by lowering the significance of old entries. In our implementation there is no removal mechanism.

The two clustering parameters ϵ and N_{min} are easy to interpret since they relate to the scale and minimum occurrence frequency. Furthermore, for sparse binary features the measurement of cardinality can be performed very efficiently. We employ Locality Sensitive Hashing from the FLANN library [21] and use its *RadiusSearch* function to retrieve the data entries within ϵ.

III. Results and Discussion

The presented unusual event detection framework was evaluated on a 7-day sequence of neuromorphic vision data recorded in a pedestrian zone using a top-view setup. The sequence exhibits a 128×128 pixel spatial resolution and a temporal sampling equivalent to 24 *fps*, resulting in nearly 15 million frames in total. Figure 4 left shows a sample RGB-image of the scene with a corresponding motion-history [1] image displaying the evolution of motion gradients in the neuromorphic data over a short time span.

Generating ground-truth for rare events in the dataset is difficult, since the frequency of events typically follows a heavy-tailed power-law or Zipf's law distribution [22]. Nevertheless, we annotated 17 instances of highly unusual events where a vehicle enters the pedestrian zone from an arbitrary direction. We used these annotations to examine whether these events are truly detected among the rarest events.

In order to verify that our representation and discrete hash table based indexing concept can cope with the *curse of dimensionality*, we performed a simple experiment. All computed dynamic features for the entire dataset were inserted into a hash table and the hash occupancy ratio was computed. Figure 5 shows how the number of hash table entries changes with increasing number of inserted features. The dashed line shows the linear increase limit, which would be caused by completely decorrelated, random feature input. As it can be seen from the figure the hash table occupancy is sublinearly converging and yields an approximately 17% occupancy ratio after the 7-days input. Indexing and searching the resulting

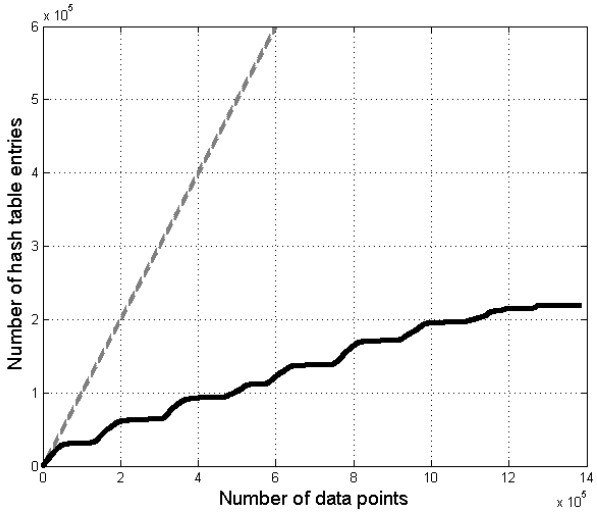

Fig. 5. The hash occupancy ratio measured on the entire 7-day dataset. Dashed line shows the linear increase limit for random data, and the dark continuous curve shows the strongly sub-linear and converging behavior for our computed features. Note that the small plateaus in the latter curve are caused by observations at night due to the slighter variability in the data.

232000 hash table entries in a Hamming space still remains computationally lightweight and memory efficient due to the efficient indexability by Locality Sensitive Hashing, fast Hamming distance computations and vector sparsity.

We performed three experiments to evaluate the clustering quality and rare event detection capability of the proposed framework. As shown in Table II we partitioned the input data using three partitioning schemes: batch-only, 50% offline - 50% online, and 50% offline with three (daily) online updates. All experiments used the identical parameter set of neighborhood radius and minimum required density (ϵ=5, N_{min}=5). All the three experiments yield very similar results, where all annotated unusual events are detected as they appear at the end of event list sorted according to observation frequency. In addition in Figure 6 we show the seven most usual and most unusual event classes discovered in our one-week dataset, as obtained in Experiment 1. As it can be seen the proposed method detects objects with unusual spatial (shape, size) and motion (velocity magnitude and direction) characteristics.

TABLE II
Rare event detection experiments

Nr.	Offline/Online	Detected rare events
1	fully batch	17 out of 17
2	50% / 50%	17 out of 17
3	50% / (3 updates)	17 out of 17

In order to further investigate the implicit structure of recovered usual and unusual features within the Hamming space we employ the recent t-SNE [23] dimensionality reduction technique on the selected frequent and rare features. Figure 7 shows the two-dimensional projection of the most relevant usual event clusters and unusual events along with some image examples characterizing individual data points.

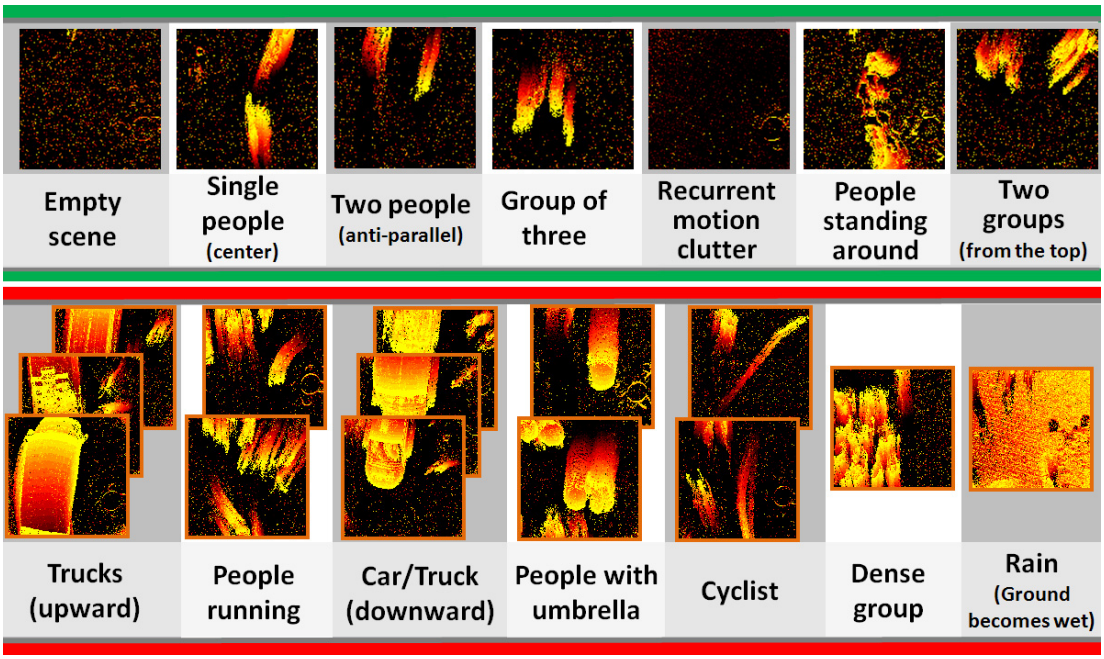

Fig. 6. The seven most usual (top row) and most unusual (bottom) row event classes discovered in our one-week neuromorphic dataset.

As it can be seen the rare event data exhibits strongly drifting characteristics with high variance, leading to a different scaling than for the distribution of the usual events. Since the employed clustering scheme uses a single constant scale and it does not consider cluster scale variations, it leads to an increased granularity for rare events. In addition, the recovered distribution of detected rare events shows that due to the employed high-dimensional representation (i.e. variability in the data) an even larger amount of observations is necessary to build a meaningful statistics. The method is, however, capable to collect and analyse an order-of-magnitude more data when performing batch clustering and significantly more with regular online updates.

Implementation details: the clustering method was implemented in Matlab with functions of hash table generation, indexing and nearest neighbor search coded in C++. The full batch processing (see Table II: Experiment 1) step takes approximately 10 minutes on a modern PC using the combined Matlab - C++ code. Embedded implementation and networked operation appear to be viable options, such as sparse feature computation performed in a smart camera, while indexing and clustering performed on a remote server.

The large amount of noise in the input data calls for possible improvements: principal component analysis or sparse random projections could be applied to further reduce the dimensionality of the input and to smooth the data at the same time.

IV. Conclusions and Outlook

In this paper we presented a density-driven computational framework capable to perform clustering in case of streaming data resulting in clusters well approximating the implicit structure of the underlying high-dimensional distribution. The employed sparse binary representation of the extracted spatio-temporal patterns well preserves the most significant spatial and dynamic attributes such as the location, dimensions and motion path of multiple moving objects, and despite the employed joint representation resulting in a great variability, it can succesfully capture observed usual and rare events without suffering the curse of dimensionality. Furthermore, given the compact representation of the visual input, the learned statistical representation and the efficient computation, the proposed scheme appears to be well applicable in practically relevant settings.

Acknowledgment

The authors would like to thank Stephan Schraml for valuable hints and discussion. This work was supported by the Embedded Computer Vision (ECV) project under the COMET program and the SHARE project in the IV2Splus program, both projects of the Austrian Research Promotion Agency (FFG). Furthermore, support from the SECRET Interactive project of the KIRAS Security Research Promotion Programme of the Austrian Federal Ministry of Transport, Innovation and Technology is acknowledged.

References

[1] G. R. Bradski and J. W. Davis, "Motion segmentation and pose recognition with motion history gradients," *Mach. Vis. Appl.*, vol. 13, no. 3, pp. 174–184, 2002.

[2] X. Wang, K. T. Ma, G.-W. Ng, and W. E. Grimson, "Trajectory analysis and semantic region modeling using nonparametric hierarchical bayesian models," *Int. J. Comput. Vision*, vol. 95, no. 3, pp. 287–312, Dec. 2011.

[3] D. Makris and T. Ellis, "Learning semantic scene models from observing activity in visual surveillance," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 3, pp. 397–408, 2005.

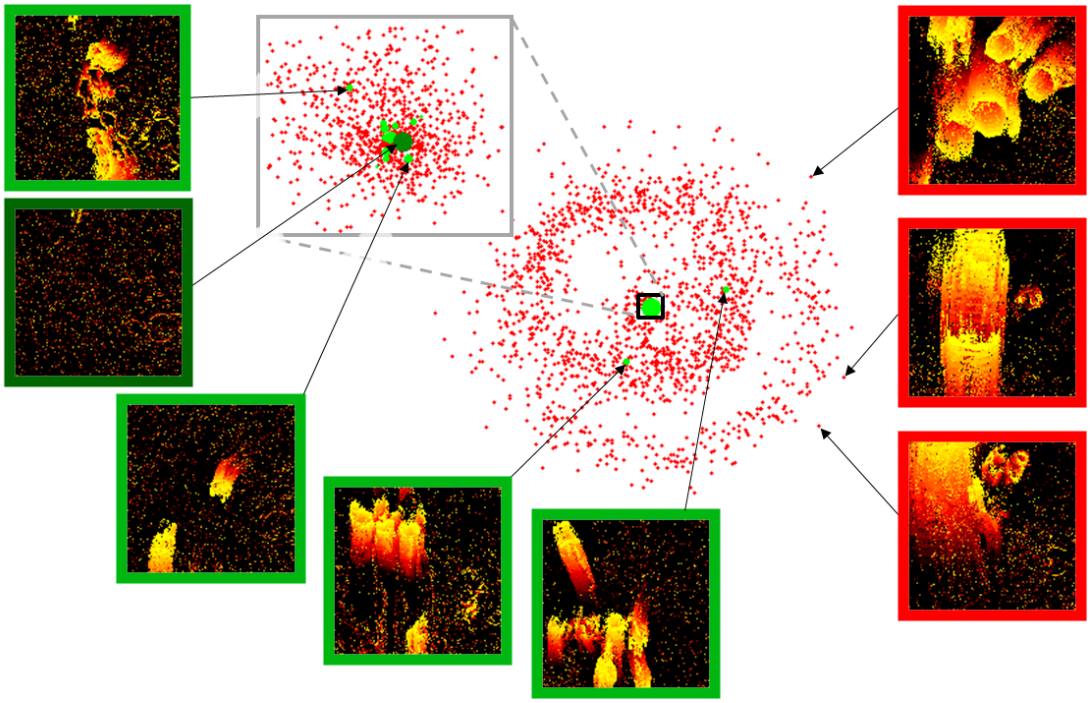

Fig. 7. Low-dimensional representation of usual and unusual event data points recovered in Experiment 1 (see Table II). The size (significance) of usual event clusters (in green) is heavily downscaled to preserve visibility. The presented data points capture in terms of density (observation frequency) more than 99% of all datapoints in the entire dataset.

[4] M. D. Breitenstein, H. Grabner, and L. Van Gool, "Hunting nessie – real-time abnormality detection from webcams," 2009, pp. 1243–1250.

[5] R. Schuster, R. Mörzinger, W. Haas, H. Grabner, and L. Van Gool, "Real-time detection of unusual regions in image streams," in *Proc. of the International Conference on Multimedia*, 2010, pp. 1307–1310.

[6] M. Douze, H. Jégou, C. Schmid, and P. Pérez, "Compact video description with precise temporal alignment," in *European Conference on Computer Vision*, 2010, pp. 522–535.

[7] G. Willems, T. Tuytelaars, and L. V. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. of the European Conference on Computer Vision*, 2008, pp. 650–663.

[8] I. Laptev and P. Perez, "Retrieving actions in movies," in *IEEE International Conference on Computer Vision*, 2007.

[9] G. Zhao, T. Ahonen, J. Matas, and M. Pietikäinen, "Rotation-invariant image and video description with local binary pattern features," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1465–1477, 2012.

[10] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. European Conf. on Computer Vision*, 2010, pp. 140–153.

[11] J. Kwon and K. M. Lee, "A unified framework for event summarization and rare event detection," in *CVPR*, 2012.

[12] B. Georgescu, I. Shimshoni, and P. Meer, "Mean shift based clustering in high dimensions: A texture classification example," in *ICCV*, 2003, pp. 456–463.

[13] C. Mead, "Neuromorphic Electronic Systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–36, Oct. 1990.

[14] T. Delbrck, "Frame-free dynamic digital vision," in *Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, 2008, pp. 21–26.

[15] C. P. P. Lichtsteiner and T. Delbrck, "A 128128 120db 15us latency asynchronous temporal contrast vision sensor," in *IEEE Journal of SSC*, vol. 43, 2008, pp. 566 – 576.

[16] M. Ester, H.-P. Kriegel, J. Sander, M. Wimmer, and X. Xu, "Incremental clustering for mining in a data warehousing environment," in *VLDB*, 1998, pp. 323–333.

[17] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *ICML*, 2010, pp. 111–118.

[18] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Thirtieth Annual ACM Symposium on the Theory of Computing*, 1998, pp. 604–613.

[19] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV (1)*, 2008, pp. 304–317.

[20] Y. Chen and L. Tu, "Density-based clustering for real-time stream data," in *International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 133–142.

[21] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Application VISSAPP'09)*. INSTICC Press, 2009, pp. 331–340.

[22] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2368–2382, 2011.

[23] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, Nov. 2008.