Event-Based Motion Segmentation by Motion Compensation —Supplementary Material—

Timo Stoffregen[1,2], Guillermo Gallego[3], Tom Drummond[1,2], Lindsay Kleeman[1], Davide Scaramuzza[3]

[1]Dept. Electrical and Computer Systems Engineering, Monash University, Australia.
[2]Australian Centre of Excellence for Robotic Vision, Australia.
[3]Dept. Informatics (Univ. Zurich) and Dept. Neuroinformatics (Univ. Zurich & ETH Zurich), Switzerland.

A. Multimedia Material

The video accompanying this work is available at https://youtu.be/0q6ap_OSBAk

B. Two Additional Motion-Compensation Segmentation Methods

In this section, we describe how two classical clustering methods (mixture densities and fuzzy k-means) can be modified to tackle the task of event-based motion segmentation, by leveraging the idea of motion-compensation [20, 21] (Sections B.1 and B.2). Examples comparing the three per-event segmentation models developed (Algorithms 1 to 3) are given in Section B.3; they are called *proposed* (or *Layered*), *Mixture Densities* and *Fuzzy k-Means*, respectively.

B.1. Mixture Densities

The mixture models framework [40, 41] can be adapted to solve the segmentation problem addressed. The idea is to fit a mixture density to the events $\mathcal{E}$, with each mode representing a cluster of events with a coherent motion.

Problem Formulation. Specifically, following the notation in [40, Ch.10], we identify the elements of the problem: the data points are the events $\mathcal{E}$ without taking into account polarity; thus, feature space is the volume V, and, consequently, the clusters are comprised of events (i.e., they are not clusters of optic flow vectors in velocity space).

The mixture model states that events $e_k \in V$ are distributed according to a sum of several distributions ("clusters"), with mixing weights ("cluster probabilities") $\boldsymbol{\pi} \doteq \{P(\omega_j)\}_{j=1}^{N_\ell}$:

$$p(e_k|\boldsymbol{\theta}) = \sum_{j=1}^{N_\ell} p(e_k|\omega_j, \boldsymbol{\theta}_j) P(\omega_j), \tag{8}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\theta}_j\}_{j=1}^{N_\ell}$ are the parameters of the distributions of each component of the mixture model and we assumed that the parameters of each cluster are independent of each other: $p(e_k|\omega_j, \boldsymbol{\theta}) = p(e_k|\omega_j, \boldsymbol{\theta}_j)$. The function $p(z|\boldsymbol{\theta})$ in (8), with $z \in V$, is a scalar field in V representing the density of events in V as a sum of several densities, each of them corresponding to a different cluster, and each cluster describing a coherent motion.

To measure how well the j-th cluster explains an event (8), we propose to use the unweighted IWE (10):

$$p(z \,|\, \omega_j, \boldsymbol{\theta}_j) \;\propto\; H_j\left(\mathbf{x}'(z; \boldsymbol{\theta}_j)\right) \tag{9}$$

$$H_j\left(\mathbf{x}\right) \;\doteq\; \sum_{m=1}^{N_e} \delta\left(\mathbf{x} - \mathbf{x}'_{mj}\right) \tag{10}$$

with warped event location $\mathbf{x}'_{mj} = \mathbf{W}(\mathbf{x}_m, t_m; \boldsymbol{\theta}_j)$. The image point $\mathbf{x}'(z; \boldsymbol{\theta}_j)$ corresponds to the warped location of point $z \in V$ using the motion parameters of the j-th cluster.

Hence, the goodness of fit between a point $z \in V$ and the j-th cluster is measured by the amount of event alignment (i.e., "sharpness"): the larger the IWE of the cluster at the warped point location, the larger the probability that z belongs to the cluster.

Notice that the choice (9) makes the distribution of each component in the mixture $p(z \,|\, \omega_j, \boldsymbol{\theta}_j)$ be constant along the point trajectories defined by the warping model of the cluster, which agrees with the "tubular" shape mentioned in the problem statement (Section 3). The mixture model (8) may not be constant along point trajectories since it is a weighted sum of several distributions, each with its own point trajectories.

Iterative Solver: EM Algorithm. With the above definitions, we may apply the EM algorithm in [40, Ch.10] to compute the parameters of the mixture model, by maximizing the (log-)likelihood of the mixture density:

$$(\boldsymbol{\theta}^*, \boldsymbol{\pi}^*) = \arg\max_{(\boldsymbol{\theta}, \boldsymbol{\pi})} \sum_{k=1}^{N_e} \log\, p(e_k|\boldsymbol{\theta}) \tag{11}$$

Algorithm 2 Event-based Motion Segmentation using Mixture Density Model

Input: events $\mathcal{E} = \{e_k\}_{k=1}^{N_e}$ in a space-time volume V of the image plane, and number of clusters N_ℓ.
Output: cluster parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_j\}_{j=1}^{N_\ell}$ and mixing weights $\boldsymbol{\pi} \doteq \{P(\omega_j)\}_{j=1}^{N_\ell}$.
Procedure:
Initialize $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$.
Iterate until convergence:
• Update the mixing weights (12), using the current motion parameters $\boldsymbol{\theta}$ and the mixing weights from the previous iteration in (13).
• Update motion parameters $\boldsymbol{\theta}$ by ascending on (11).

In the E-step, the mixing weights $\boldsymbol{\pi}$ are updated using

$$P(\omega_j) = \frac{1}{N_e} \sum_{k=1}^{N_e} p(\omega_j | e_k, \boldsymbol{\theta}_j) \tag{12}$$

with membership probabilities given by the Bayes formula

$$p(\omega_j | e_k, \boldsymbol{\theta}_j) = \frac{p(e_k | \omega_j, \boldsymbol{\theta}_j) P(\omega_j)}{\sum_{i=1}^{N_\ell} p(e_k | \omega_i, \boldsymbol{\theta}_i) P(\omega_i)}. \tag{13}$$

In the M-step, gradient ascent or conjugate gradient [42] of the log-likelihood (11) with respect to the warp parameters $\boldsymbol{\theta}$ is used to update $\boldsymbol{\theta}$, in preparation for the next iteration.

The pseudo-code of this mixture model method is given in Algorithm 2. From the mixing weights and the motion parameters, it is straightforward to compute the event-cluster assignment probabilities using (13). To initialize the iteration, we use the procedure described in Section 3.5.

Notice that, during the EM iterations, the above method not only estimates the cluster parameters $\boldsymbol{\theta}$ and the mixing weights $\boldsymbol{\pi}$ but also the distributions $p(z|\omega_i, \boldsymbol{\theta}_i)$ themselves, i.e., the "shape" of the components of the mixture model. These distributions get sharper (more peaky or "in focus") around the segmented objects as iterations proceed, and blurred around the non-segmented objects corresponding to that cluster. An example is given in Section B.3.

B.2. Fuzzy k-Means

Event-based motion segmentation can also be achieved by designing an objective function similar to the one used in the fuzzy k-means algorithm [40, Ch.10].

Problem Formulation. This approach seeks to maximize

$$(\boldsymbol{\theta}^*, \mathbf{P}^*) = \arg\max_{\boldsymbol{\theta}, \mathbf{P}} \sum_{j=1}^{N_\ell} \sum_{k=1}^{N_e} p_{kj}^b d_{kj}, \tag{14}$$

where $b > 1$ (e.g., $b = 2$) adjusts the blending of the different clusters, and the goodness of fit between an event e_k and a cluster j in V is given in terms of event alignment (i.e., "sharpness"):

$$d_{kj} \doteq \log H_j(\mathbf{x}'_{kj}), \tag{15}$$

the value of the unweighted IWE (10) at the warped event location using the motion parameters of the cluster. We use the logarithm of the IWE, as in (11), to decrease the influence of large values of the IWE, since these are counted multiple times if the events are warped to the same pixel location. Notice that (14) differs from (2)-(5): the responsibilities p_{kj} appear multiplying the IWE (i.e., they are not included in a weighted IWE), and the sum is over the events (as opposed to over the pixels (4)).

Notice also that this proposal is different from clustering in optical flow space (Fig. 16). As mentioned in Section B.1, here the feature space is the space-time volume $V \in \mathbb{R}^3$ (i.e., event location), rather than the optical flow space ($\mathbb{R}^2$) (i.e., event velocity).

Algorithm 3 Event-based Motion Segmentation using the Fuzzy k-Means Method

Input: events $\mathcal{E} = \{e_k\}_{k=1}^{N_e}$ in a space-time volume V of the image plane, and number of clusters N_ℓ.
Output: cluster parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_j\}_{j=1}^{N_\ell}$ and event-cluster assignments $\mathbf{P} \equiv p_{kj} \doteq P(e_k \in \ell_j)$.
Procedure:
Initialization (as in Section 3.5).
Iterate until convergence:
• Update the event-cluster assignments p_{kj} using (16).
• Update motion parameters $\boldsymbol{\theta}$ by ascending on (14).

Iterative Solver: EM Algorithm. The EM algorithm may also be used to solve (14). In the E-step (fixed warp parameters $\boldsymbol{\theta}$) the responsibilities are updated using the closed-form partitioning formula

$$p_{kj} = d_{kj}^{\frac{1}{b-1}} \Big/ \sum_{i=1}^{N_\ell} d_{ki}^{\frac{1}{b-1}}. \tag{16}$$

In the M-step (fixed responsibilities) the warp parameters of the clusters $\boldsymbol{\theta}$ are updated using gradient ascent or conjugate gradient. The pseudo-code of the event-based fuzzy k-means segmentation method is given in Algorithm 3.

B.3. Comparison of Three Motion-Compensation Segmentation Methods

We compare our method with the two above-mentioned methods (Sections B.1 and B.2) that we also designed to leverage motion compensation.

Fig. 8 shows the comparison of the three methods on a toy example with three objects (a filled pentagon, a star

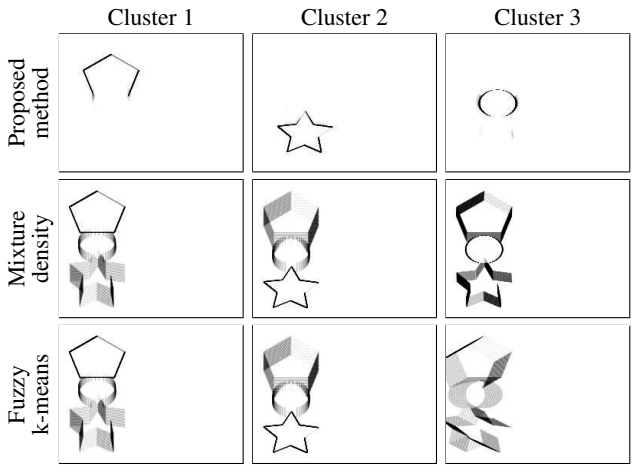


Figure 8: Comparison of three methods for event-based Motion Segmentation: Algorithms 1 to 3 (one per row).

and a circle) moving in different directions on the image plane. In the mixture density and fuzzy k-means methods, the motion-compensated IWEs do not include the event-cluster associations $\mathbf{P}$, and so, all objects appear in all IWEs, sharper in one IWE than in the others. In contrast, in the proposed method (Algorithm 1), the associations are included in the motion-compensated image of the cluster (weighted IWE), as per equation (2), and so, the objects are better split into the clusters (with minor "ghost" effects, as illustrated in Fig. 2), thus yielding the best results.

It is worth mentioning that the three per-event segmentation methods are novel: they have not been previously proposed in the literature. We decided to focus on Algorithm 1 in the main part of the paper and thus leave the adaptation of classical methods (mixture model and fuzzy k-means) for the supplementary material.

B.4. Computational Complexity

Next, we analyze the complexity of the three segmentation methods considered (Algorithms 1 to 3), defined by objective functions (5), (11) and (14), respectively. The core of the segmentation methods is the computation of the images of warped events (IWEs (2) or (10); one per cluster), which has complexity $O(N_e N_\ell)$.

Proposed (Layered) Model. The complexity of updating the event assignments using (7) is essentially that of computing the (weighted) IWEs of all clusters, i.e., $O(N_e N_\ell)$. The complexity of computing the contrast (4) of a generic image is linear in the number of pixels, $O(N_p)$, and so, the complexity of computing the contrast of one IWE is $O(N_e + N_p)$. The computation of the contrast is negligible compared to the effort required by the warp. Computing the contrast of N_ℓ clusters (corresponding to a set of candidate parameters) has complexity $O\big((N_e + N_p)N_\ell\big)$. Since multiple iterations N_{it} may be required to find the optimal parameters, the total complexity of the iterative algorithm used is $O\big((N_e + N_p)N_\ell N_{\text{it}}\big)$.

Mixture Density Model. The complexity of updating the mixture weights is that of computing the posterior probabilities $p(\omega_j|e_k, \boldsymbol{\theta}_j)$, which require to compute the IWEs of all clusters, i.e., complexity $O(N_e N_\ell)$. The complexity of updating the motion parameters is also that of computing the contrasts of the IWEs of all clusters, through multiple ascent iterations. In total, the complexity is $O\big((N_e + N_p)N_\ell N_{\text{it}}\big)$.

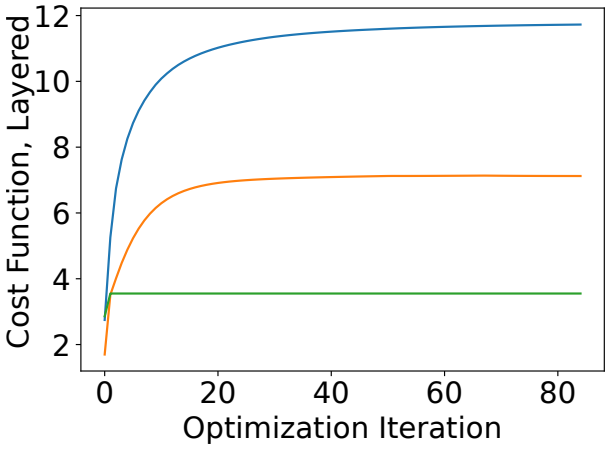
Fuzzy k-means Model. The complexity of computing the responsibilities (16) is that of computing N_ℓ IWEs (values d_{kj}), i.e., $O(N_e N_\ell)$. The complexity of updating the motion parameters is that of computing the objective function (14), $O(N_e N_\ell)$, through multiple iterations. In total, the complexity is $O(N_e N_\ell N_{\text{it}})$.

Plots of Computational Effort and Convergence. Fig. 9 shows the convergence of the three above methods on real data from a traffic sequence that is segmented into four clusters (Fig. 10 and third column of Fig. 4): three cars and the background due to ego-motion. The top plot, Fig. 9a, shows the evolution of the sum-of-contrasts objective function (5) vs the iterations. All methods stagnate after ≈ 20 iterations, and, as expected, the proposed method provides the highest score among all three methods (since it is designed to maximize this objective function). The Mixture model and Fuzzy k-means methods do not provide such a large score mostly due to the event-cluster associations, since they are not as confident to belonging to one cluster as in the proposed method. Fig. 9b displays the number of warps (i.e., number of IWEs) that each method computes as the optimization iterations proceed; as it can be shown, the relationship is approximately linear, with the proposed method performing the least warps for a considerable number of iterations, before stagnation (Fig. 9a).

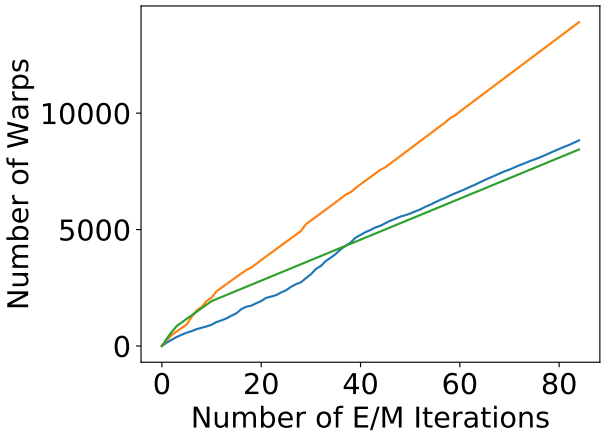
C. Additional Experiments

C.1. Non-rigid Moving Objects

In the following experiments we show how our method deals with non-rigid objects. Our algorithm warps events $\mathcal{E}$ according to point-trajectories described by parametric motion models whose parameters are assumed constant over the (small) time Δt spanned by $\mathcal{E}$. Low-dimensional parametric warp models, such as the patch-based optic flow (2-DOF, linear trajectories), rotational motion in the plane (1-DOF) or in space (3-DOF) are simple and produce robust results by constraining the dimensionality of the solution space in the search for optimal point-trajectories. However,



(a) Value of the sum-of-contrasts objective function (5) for each of the three methods per iteration of optimization.



(b) Number of warps performed by each method per iteration of optimization.

Figure 9: *Comparison of three Methods.* We compare the convergence properties of three motion-compensated event-segmentation methods (Algorithms 1 to 3): proposed (layered) method (*blue*), Mixture Density Model (*orange*) and Fuzzy k-Means (*green*). Data used is from the Traffic Sequence (third column of Fig. 4), the warped events at each iteration are visualized in Fig. 10.

simple warp models (both in event-based vision or in traditional frame-to-frame vision) have limited expressiveness: they are good for representing rigidly moving objects, but do not have enough degrees of freedom to represent more complex motions, such as deformations (e.g., pedestrian, birds, jelly fish, etc.). One could consider using warps able to describe more complex motions, such as part-based warp models [43] or infinite-dimensional models [44]. But this would make the segmentation problem considerably harder, not only due to the increased dimensionality of the search space, but also because it would be possibly filled with multiple local minima.

Pedestrian. Fig. 11 shows a pedestrian walking past the camera while it is panning. In spite of using simple warp models, our method does a good job at segmentation: the background (due to camera motion), the torso of the person and the swinging arms are segmented in separate clusters. This is so, because during the short time span of $\mathcal{E}$ (in the order of milliseconds), the objects move approximately rigidly.

Popping Balloon. In order to test the limits of this assumption, we filmed the popping of a balloon with the event camera (see Fig. 12). While the segmentation struggles to give a clear result in the initial moments of puncturing (12b), it still manages to give reasonable results for the fast moving, contracting fragments of rubber flung away by the explosion (12c, 12d).

C.2. Additional to Section 4.3 - Continuum Depth Variation

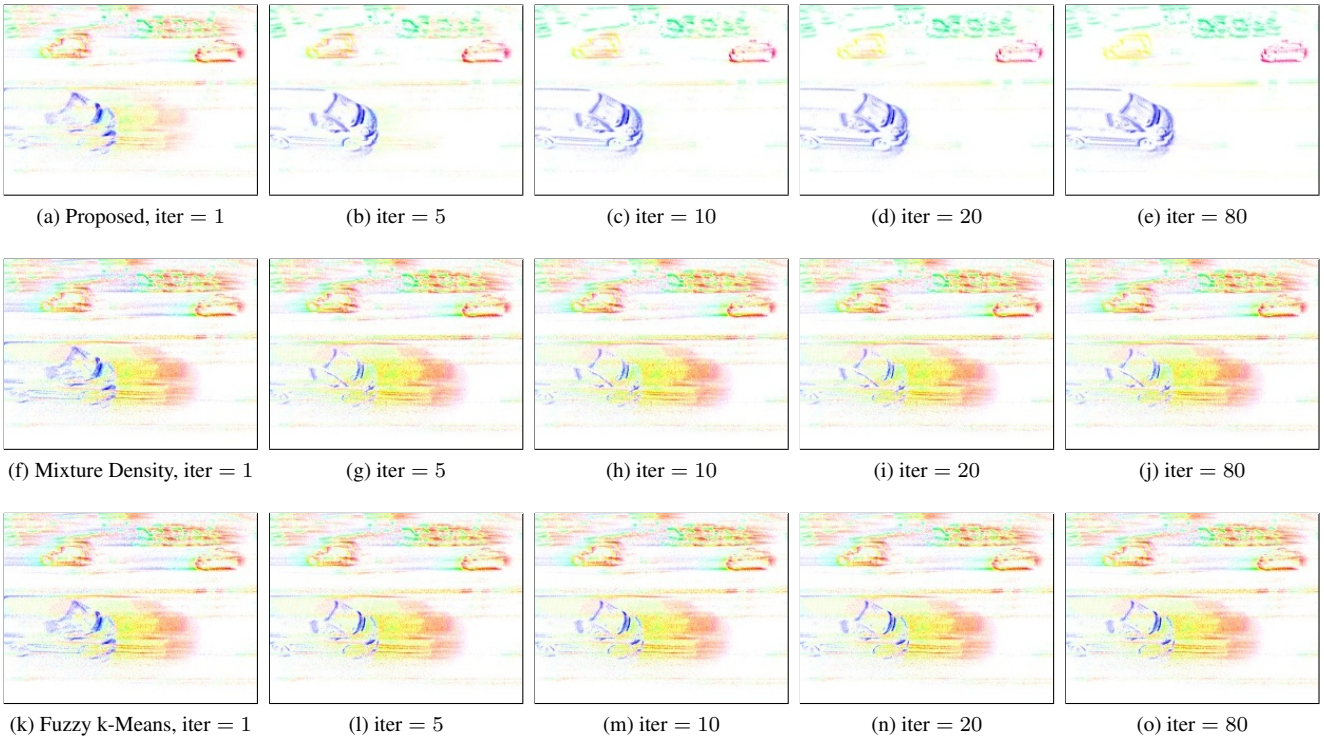
In this experiment we essentially show a scene similar to that in Fig. 7. The difference is that the scene in Fig. 13 shows a truly continuous depth variation. As can be seen in the results (using $N_\ell = 15$), our method discretizes the segmentation, although it is noteworthy that each "slice" of depth appears to fade toward foreground and background. This is because the method becomes less certain of the likelihood of events that sit between clusters, the darkness of a region reflecting the likelihood of a given event belonging to that cluster.

C.3. Continuum Depth Variation with High-Resolution Event-based Camera

Due to the recent development of new, high resolution event-based cameras [37], we show the results of our method on the output of a Samsung DVS Gen3 sensor, with a spatial resolution of 640×480 pixels. In this experiment we show the segmentation of several scenes (a textured carpet, some leaves and a temple poster) as the camera moves. Due to ego-motion induced parallax, there is a continuous gradient in the motion in the scene, i.e., the scenes present a continuum of depths. As can be seen in Fig. 14, our method works the same on high-resolution data.

C.4. Comparison to k-means Optic Flow Clustering

Finally, the following experiments shows the comparison of our method against k-means clustering of optic flow. We first illustrate the difference with a qualitative example and then quantitatively show the ability of our method to resolve small differences in velocities compared to k-means. To this end, we use an event-based camera mounted on a motorized linear slider, which provides accurate ground truth position of the camera. Since the camera moves at constant speed in a 1-D trajectory, the differences in optical flow values



(a) Proposed, iter = 1 (b) iter = 5 (c) iter = 10 (d) iter = 20 (e) iter = 80

(f) Mixture Density, iter = 1 (g) iter = 5 (h) iter = 10 (i) iter = 20 (j) iter = 80

(k) Fuzzy k-Means, iter = 1 (l) iter = 5 (m) iter = 10 (n) iter = 20 (o) iter = 80

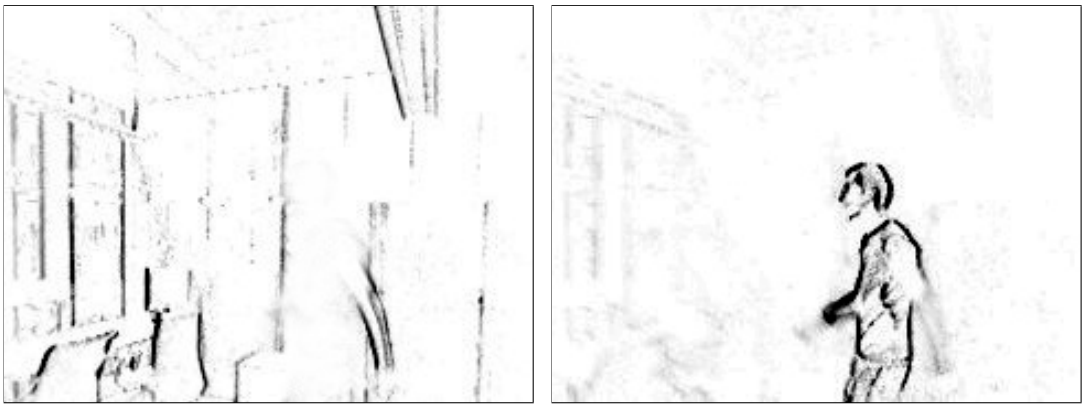
Figure 10: Images of the motion-corrected events for three segmentation methods. From left to right the images show the state after 1, 5, 10, 20 and 80 iterations respectively. *Top Row*: Algorithm 1, *Middle Row*: Algorithm 2, *Bottom Row*: Algorithm 3.

observed when viewing a static scene are due to parallax from the different depths of the objects causing the events.

Numbers Sequence. In this experiment, we placed six printed numbers at different, known depths with respect to the linear slider. The event-based camera moved back and forth on the slider at approximately constant speed. Due to parallax, the objects at different depths appear to be moving at different speeds; faster the closer the object is to the camera. Thus we expect the scene to be segmented into six clusters, each corresponding to a different apparent velocity.

Fig. 15 compares the results of k-means clustering optic flow and Algorithm 1. To compute optical flow we use conventional methods on reconstructed images at a high frame rate [45], with the optical flow method in [46] producing better results on such event-reconstructed images than state-of-the-art learning methods [47]. The results show that the velocities corresponding to the six numbers are too similar to be resolved correctly by the two-step approach (flow plus clustering), as evidenced by the bad segmentation of the scene (numbers 3, 4 and 5 are clustered together, whereas three clusters are used to represent the events of the fastest moving number–the zero, closest to the camera). In contrast, our method accurately clusters the events according to the motion of the objects causing them, in this case, according to velocities, since we used an optical flow warp (linear motion on the image plane). The higher accuracy of our method is easily seen in the sharpness of the motion-compensated images (cf. Fig. 15d and Fig. 15f).

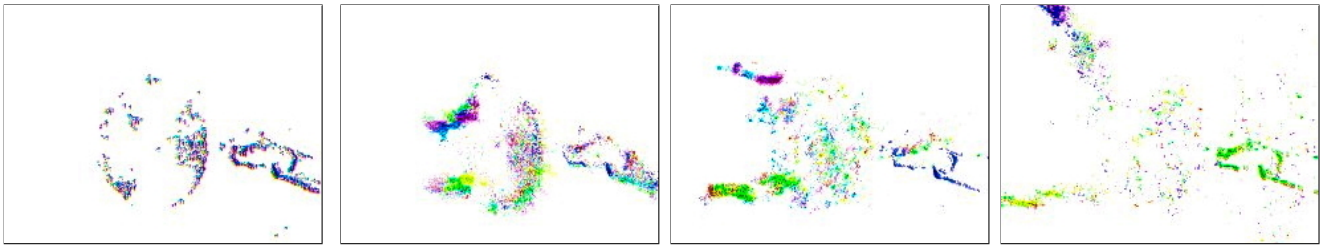
Rocks at Different Speeds. We also tested our algorithm on two real sequences with six objects of textured images of pebbles (qualitatively similar to Fig. 5), in which the relative velocities of the objects were $50\,\text{pixels/s}$ and $6\,\text{pixels/s}$, respectively. Fig. 16 shows the results. If the objects are moving with sufficiently distinct velocities (Fig. 16a), the clusters can be resolved by the two-step approach. However, once the objects move with similar velocities (Fig. 16a), k-means clustering of optical flow is unable to correctly resolve the different clusters. In contrast, our method works well in both cases; it is much more accurate: it can resolve differences of $6\,\text{pixel/s}$ for objects moving at $50\,\text{pixel/s}$ to $80\,\text{pixel/s}$, given the same slice of events.



(a) Cluster 1 (b) Cluster 2

(c) Cluster 3 (d) All clusters (merged IWEs)

Figure 11: *Non-Rigid Scene.* A person walks across a room, arms swinging. The room 11a, the body 11b and the arms 11c are segmented out, with greater uncertainty to the event associations in areas of deformation (such as elbows), visible in the fact that events are associated to both clusters (events colored by cluster in 11d).



(a) (b) (c) (d)

Figure 12: *Non-Rigid Moving Objects.* From left to right: snapshots of segmentation of balloon popping. Run with $N_\ell = 4$ clusters, events colored by cluster membership.

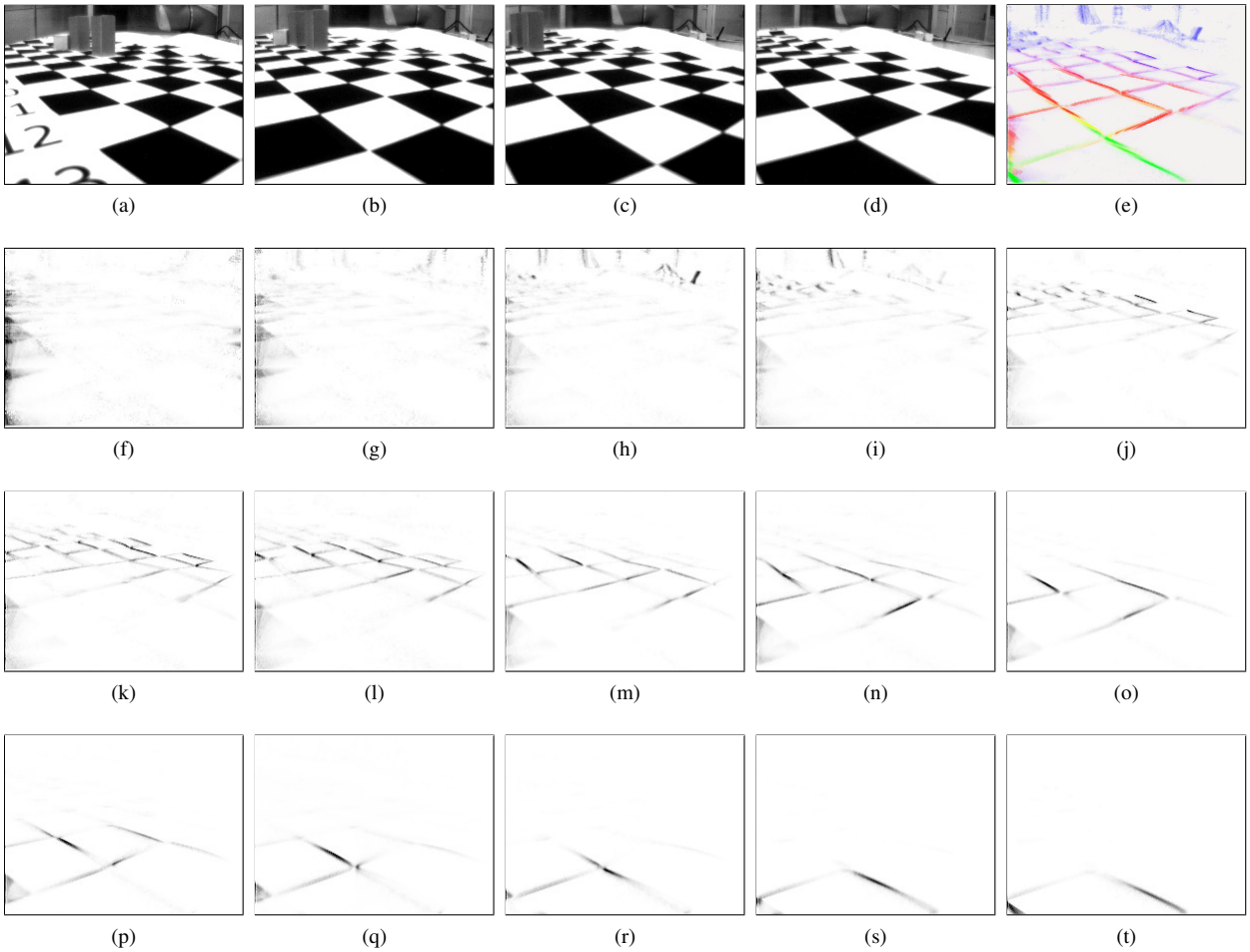


Figure 13: Sequence from a camera translating past a checkerboard (13a-13d). These grayscale frames, provided by the DAVIS [35] are not used by our method; they are just for visualization purposes. Each image in 13f-13t shows the IWE of each cluster (15 clusters, optical flow motion models). 13e shows the segmented output (combined IWEs) in the accustomed colored format.

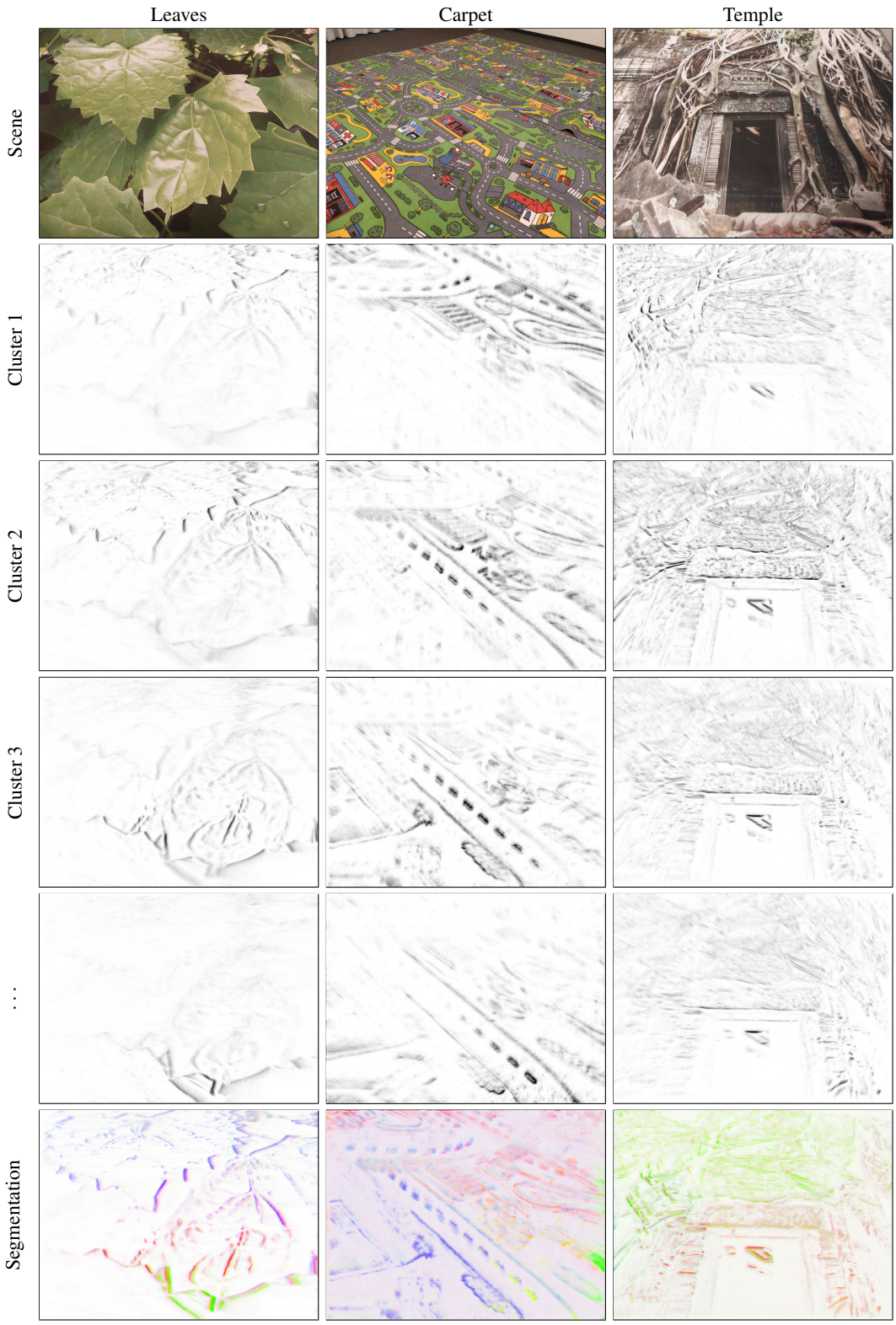
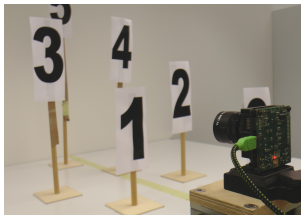
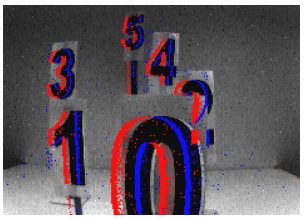


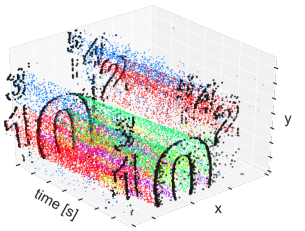
Figure 14: Scenes recorded with a Samsung DVS Gen3 event camera (640×480 pixels); algorithm run with ten clusters ($N_\ell = 10$). From top to bottom: scene recorded, four of the clusters (motion-compensated IWEs, with darkness indicating event likelihoods), and the accustomed colored segmentation (as in Fig. 2). These examples illustrate that our method can be used to segment the scene according to depth from the camera, although it is not its main purpose.



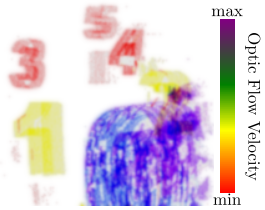
(a) DAVIS performs linear translation over a multi-object scene.



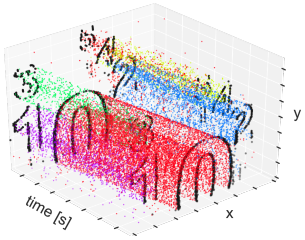
(b) Resulting image and events (red and blue, indicating polarity).



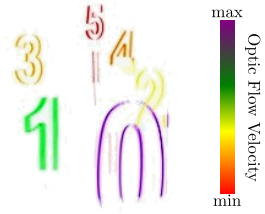
(c) Clustered volume of events (colored by cluster number).



(d) Motion-compensated image (colored by clustered optic flow).

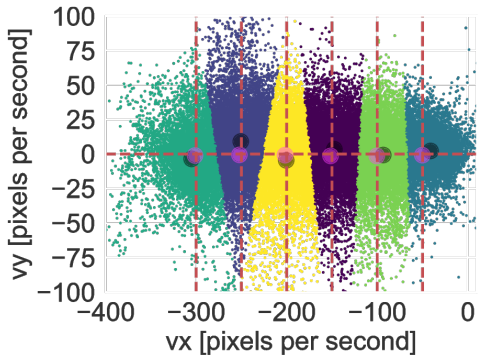


(e) Clustered volume of events (colored by cluster number).

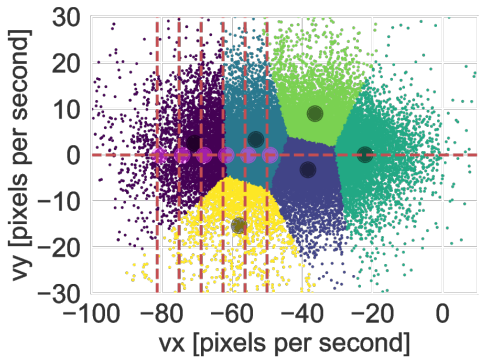


(f) Motion-compensated image (colored by recovered optic flow).

Figure 15: *Numbers Sequence*. Motion Segmentation by k-means clustering on estimated optic flow (center row) and by Algorithm 1 (bottom row).



(a) Minimum velocity between objects: 50 pixel/s



(b) Minimum velocity between objects: 6 pixel/s

Figure 16: *Rocks at Different Speeds*. Segmentation by k-means clustering of estimated optical flow ($k = 6$). The plots show the distribution of optical flow vectors and the six Voronoi diagrams resulting from k-means clustering on optic flow space. The crossings of red dashed lines indicate the ground truth optical flow velocity, the dark circles indicate the centroids of the k-means clusters. The pink circles indicate the cluster's optical flow estimated by our method (Algorithm 1).

References

[1] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck, "A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008. 1, 3

[2] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza, "Event-based vision: A survey," *arXiv:1904.08405*, 2019. 1, 2

[3] Elias Mueggler, Basil Huber, and Davide Scaramuzza, "Event-based, 6-DOF pose tracking for high-speed maneuvers," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pp. 2761–2768, 2014. 1

[4] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis, "Event-based feature tracking with probabilistic data association," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 4465–4470, 2017. 1, 5

[5] Guillermo Gallego, Jon E. A. Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza, "Event-based, 6-DOF camera tracking from photometric depth maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, pp. 2402–2412, Oct. 2018. 1

[6] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza, "EKLT: Asynchronous photometric feature tracking using events and frames," *Int. J. Comput. Vis.*, 2019. 1

[7] Jörg Conradt, Matthew Cook, Raphael Berner, Patrick Lichtsteiner, Rodney J. Douglas, and Tobi Delbruck, "A pencil balancing robot using a pair of AER dynamic vision sensors," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, pp. 781–784, 2009. 1

[8] Tobi Delbruck and Manuel Lang, "Robotic goalie with 3ms reaction time at 4% CPU load using event-based dynamic vision sensor," *Front. Neurosci.*, vol. 7, p. 223, 2013. 1

[9] Erich Mueller, Andrea Censi, and Emilio Frazzoli, "Low-latency heading feedback control with neuromorphic vision sensors using efficient approximated incremental inference," in *IEEE Conf. Decision Control (CDC)*, 2015. 1

[10] Hanme Kim, Stefan Leutenegger, and Andrew J. Davison, "Real-time 3D reconstruction and 6-DoF tracking with an event camera," in *Eur. Conf. Comput. Vis. (ECCV)*, pp. 349–364, 2016. 1

[11] Henri Rebecq, Timo Horstschäfer, Guillermo Gallego, and Davide Scaramuzza, "EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real-time," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 593–600, 2017. 1

[12] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis, "Event-based visual inertial odometry," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 5816–5824, 2017. 1

[13] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza, "Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios," *IEEE Robot. Autom. Lett.*, vol. 3, pp. 994–1001, Apr. 2018. 1

[14] Luca Zappella, Xavier Lladó, and Joaquim Salvi, "Motion segmentation: A review," in *Conf. Artificial Intell. Research and Development*, pp. 398–407, 2008. 2

[15] Björn Ommer, Theodor Mader, and Joachim M. Buhmann, "Seeing the objects behind the dots: Recognition in videos from a moving camera," *Int. J. Comput. Vis.*, vol. 83, pp. 57–71, Feb. 2009. 2

[16] Martin Litzenberger, Christoph Posch, D. Bauer, Ahmed Nabil Belbachir, P. Schön, B. Kohn, and H. Garn, "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *Digital Signal Processing Workshop*, pp. 173–178, 2006. 2

[17] Zhenjiang Ni, Sio-Hoï Ieng, Christoph Posch, Stéphane Régnier, and Ryad Benosman, "Visual tracking using neuromorphic asynchronous event-based cameras," *Neural Computation*, vol. 27, no. 4, pp. 925–953, 2015. 2

[18] Ewa Piatkowska, Ahmed Nabil Belbachir, Stephan Schraml, and Margrit Gelautz, "Spatiotemporal multiple persons tracking using dynamic vision sensor," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, pp. 35–40, 2012. 2

[19] John YA Wang and Edward H Adelson, "Layered representation for motion analysis," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 361–366, 1993. 2, 3

[20] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 3867–3876, 2018. 2, 3, 4, 5, 9

[21] Guillermo Gallego and Davide Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 632–639, 2017. 2, 5, 9

[22] Arren Glover and Chiara Bartolozzi, "Event-driven ball detection and gaze fixation in clutter," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, pp. 2203–2208, 2016. 2

[23] Valentina Vasco, Arren Glover, Elias Mueggler, Davide Scaramuzza, Lorenzo Natale, and Chiara Bartolozzi, "Independent motion detection with event-driven cameras," in *IEEE Int. Conf. Adv. Robot. (ICAR)*, 2017. 2

[24] Timo Stoffregen and Lindsay Kleeman, "Simultaneous optical flow and segmentation (SOFAS) using Dynamic Vision Sensor," in *Australasian Conf. Robot. Autom. (ACRA)*, 2017. 2, 4, 5, 6

[25] Anton Mitrokhin, Cornelia Fermuller, Chethan Parameshwara, and Yiannis Aloimonos, "Event-based moving object detection and tracking," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018. 2, 3, 4, 5, 6, 7, 8

[26] Francisco Barranco, Ching L. Teo, Cornelia Fermuller, and Yiannis Aloimonos, "Contour detection and characterization for asynchronous event sensors," in *Int. Conf. Comput. Vis. (ICCV)*, 2015. 2

[27] Guillermo Gallego, Mathias Gehrig, and Davide Scaramuzza, "Focus is all you need: Loss functions for event-based vision," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 12280–12289, 2019. 2

[28] Timo Stoffregen and Lindsay Kleeman, "Event cameras, contrast maximization and reward functions: An analysis," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, pp. 12300–12308, 2019. 2

[29] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019. 2

[30] Laurent Dardelet, Sio-Hoi Ieng, and Ryad Benosman, "Event-based features selection and tracking from intertwined estimation of velocity and generative contours," *arXiv:1811.07839*, 2018. 2

[31] Rafael C. Gonzalez and Richard Eugene Woods, *Digital Image Processing*. Pearson Education, 2009. 4

[32] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi, "Event-based visual flow," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 407–417, 2014. 4

[33] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis, "EV-FlowNet: Self-supervised optical flow estimation for event-based cameras," in *Robotics: Science and Systems (RSS)*, 2018. 4

[34] C. Fraley, "How many clusters? Which clustering method? Answers via model-based cluster analysis," *The Computer Journal*, vol. 41, pp. 578–588, Aug. 1998. 5

[35] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck, "A 240x180 130dB 3us latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014. 5, 15

[36] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza, "ESIM: an open event camera simulator," in *Conf. on Robotics Learning (CoRL)*, 2018. 6

[37] Bongki Son, Yunjae Suh, Sungho Kim, Heejae Jung, Jun-Seok Kim, Changwoo Shin, Keunju Park, Kyoobin Lee, Jinman Park, Jooyeon Woo, Yohan Roh, Hyunku Lee, Yibing Wang, Ilia Ovsiannikov, and Hyunsurk Ryu, "A 640x480 dynamic vision sensor with a 9um pixel and 300Meps address-event representation," in *IEEE Intl. Solid-State Circuits Conf. (ISSCC)*, 2017. 7, 8, 12

[38] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *Int. J. Robot. Research*, vol. 36, no. 2, pp. 142–149, 2017. 8

[39] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza, "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time," *Int. J. Comput. Vis.*, vol. 126, pp. 1394–1414, Dec. 2018. 8

[40] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Wiley, 2000. 9, 10

[41] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006. 9

[42] Jorge Nocedal and S. Wright, *Numerical Optimization*. Springer-Verlag New York, 2006. 10

[43] M. Pawan Kumar, P. H. S. Torr, and Andrew Zisserman, "Learning layered motion segmentations of video," *Int. J. Comput. Vis.*, vol. 76, pp. 301–319, Mar. 2008. 12

[44] Anthony J. Yezzi and Stefano Soatto, "Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images," *Int. J. Comput. Vis.*, vol. 53, pp. 153–167, July 2003. 12

[45] Cedric Scheerlinck, Nick Barnes, and Robert Mahony, "Continuous-time intensity estimation using event cameras," in *Asian Conf. Comput. Vis. (ACCV)*, 2018. 13

[46] Gunnar Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian Conf. on Im. Analysis (SCIA)*, pp. 363–370, 2003. 13

[47] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018. 13