EMVS: Event-based Multi-View Stereo

Henri Rebecq
rebecq@ifi.uzh.ch
Guillermo Gallego
guillermo.gallego@ifi.uzh.ch
Davide Scaramuzza
sdavide@ifi.uzh.ch

Robotics and Perception Group
University of Zurich
http://rpg.ifi.uzh.ch
Zurich, Switzerland

Abstract

Event cameras are bio-inspired vision sensors that output pixel-level brightness changes instead of standard intensity frames. They offer significant advantages over standard cameras, namely a very high dynamic range, no motion blur, and a latency in the order of microseconds. However, because the output is composed of a sequence of asynchronous events rather than actual intensity images, traditional vision algorithms cannot be applied, so that a paradigm shift is needed. We introduce the problem of Event-based Multi-View Stereo (EMVS) for event cameras and propose a solution to it. Unlike traditional MVS methods, which address the problem of estimating *dense* 3D structure from a set of known viewpoints, EMVS estimates *semi-dense* 3D structure from an event camera with known trajectory. Our EMVS solution elegantly exploits two inherent properties of an event camera: (*i*) its ability to respond to scene edges—which naturally provide semi-dense geometric information without any pre-processing operation—and (*ii*) the fact that it provides continuous measurements as the sensor moves. Despite its simplicity (it can be implemented in a few lines of code), our algorithm is able to produce accurate, semi-dense depth maps. We successfully validate our method on both synthetic and real data. Our method is computationally very efficient and runs in real-time on a CPU.

Multimedia Material

A supplemental video for this work is available on the authors' webpage:
http://rpg.ifi.uzh.ch

1 Introduction

An event camera, such as the Dynamic Vision Sensor (DVS) [9], works very differently from a traditional camera. It has *independent* pixels that only send information (called "events") in presence of brightness changes in the scene at the time they occur. Thus, the output is not an intensity image but a stream of asynchronous events at microsecond resolution, where each event consists of its space-time coordinates and the *sign* of the brightness change (i.e., no intensity). Since events are caused by brightness changes over time, an event camera naturally responds to edges in the scene in presence of relative motion.

Event cameras have numerous advantages over standard cameras: a latency in the order of microseconds, low power consumption, and a high dynamic range (130 dB vs 60 dB).

These properties make the sensors ideal in all those applications where fast response and high efficiency are crucial and also in scenes with wide variations of illumination. Additionally, since information is only sent in presence of brightness changes, the sensor removes all the inherent redundancy of standard cameras, thus requiring a very low data rate (kilobytes vs Megabytes). However, since event cameras became commercially available only very recently [9], very little related work exists, and, because their output is significantly different from that of standard cameras, traditional vision algorithms cannot be applied, which calls for new methods to process the data from these novel cameras.

This paper represents a significant step forward in structure estimation with a *single* event camera. In this regard, we formulate the 3D reconstruction problem in the event-based paradigm by generalizing the Multi-View Stereo (MVS) problem and then develop the first method to solve it.

1.1 Related Work on Event-Based Depth Estimation

To our knowledge, no previous work has addressed depth estimation from a *single* event camera. All related works tackle an entirely different problem: 3D reconstruction with *two or more* event cameras that are rigidly attached (i.e., with a fixed baseline) and share a *common clock*. These methods follow a two-step approach: first they solve the event correspondence problem across image planes and then triangulate the location of the 3D point. Events are matched in two ways: either using traditional stereo methods on artificial frames generated by accumulating events over time [7, 11], or exploiting simultaneity and temporal correlations of the events across sensors [2, 6, 8, 10].

The event-based depth estimation problem that we address significantly departs from state of the art in two ways: (*i*) we consider a *single* camera, (*ii*) we do not require simultaneous event observations. Depth estimation from a single event camera is more challenging because we cannot exploit temporal correlation between events across multiple image planes. Notwithstanding, we show that a single event camera suffices to estimate depth, and, moreover, that we are able to do it without solving the data association problem, as opposed to previous event-based stereo-reconstruction methods.

1.2 The Event-based Multi-View Stereo Problem

MVS with traditional cameras addresses the problem of 3D structure estimation from a collection of images taken from known viewpoints [13]. Our Event-based MVS (EMVS) shares the same goal; however, there are some key differences:

1. Traditional MVS algorithms work on full images, so they cannot be applied to the stream of asynchronous events provided by the sensor. EMVS must take into account the *sparse* and *asynchronous* nature of the events.
2. Because event cameras do not output data if both the sensor and the scene are static, EMVS requires the sensor to be *moved* in order to acquire visual content. In traditional MVS, the camera does not need to be in motion to acquire visual content.
3. Because events are caused by intensity edges, the natural output of EMVS is a *semi-dense* 3D map, as opposed to the dense maps of traditional MVS.

Hence, the EMVS problem consists of obtaining the 3D reconstruction of a scene from the sparse asynchronous streams of events acquired by a moving event camera with known viewpoints. Without loss of generality, it suffices to consider the case of one event camera.

To solve the EMVS problem, classical MVS approaches cannot be directly applied since they work on intensity images. Nevertheless, our event-based approach builds upon previous works on traditional MVS [12]. In particular, we follow (in Section 2) the solving strategy of Scene Space MVS methods [12], which consist of two main steps: computing an aggregated consistency score in a discretized volume of interest (the Disparity Space Image (DSI)) by warping image measurements, and then finding 3D structure information in this volume.The term DSI [14] is interchangeably used to refer to the projective sampling of the volume (i.e., discretized volume) or to the scalar function defined in it (i.e., the score). Just by considering the way that visual information is provided, we can point out two key differences between the DSI approaches in MVS and EMVS:

1. In classical MVS, the DSI is *densely* populated using pixel intensities. In EMVS, the DSI may have holes (voxels with no score value), since warped events are also *sparse*.
2. In classical MVS, scene objects are obtained by finding an optimal surface in the DSI. By contrast, in EMVS, finding *semi-dense* structures (e.g., points, curves) is a better match to the sparsity of the DSI.

1.3 Contribution

In this paper, we address the problem of structure estimation with a single event camera by introducing the concept of Event-based Multi-View Stereo (EMVS), and we propose the first algorithm to solve this problem. Our approach follows a Space-Sweep [3] voting and maximization strategy to estimate semi-dense depth maps at selected viewpoints, and then we merge the depth maps to build larger 3D models. We evaluate the method on both synthetic and real data. The results are analyzed and compared with ground truth, showing the successful performance of our approach. We release datasets to the public.

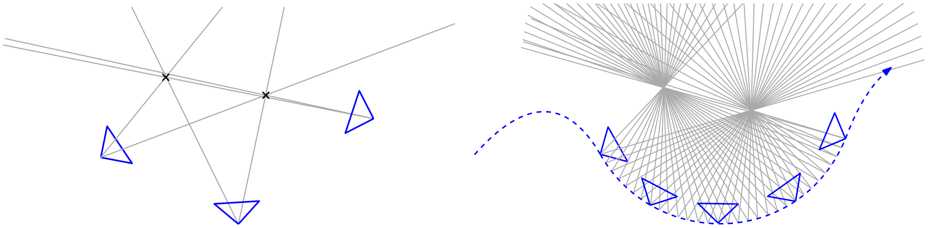
2 Event-Based Space-Sweep Method

Our method to solve the EMVS problem is similar in spirit to Collin's Space-Sweep approach for MVS [3], which shows how sparsity can be leveraged to estimate 3D structures without the need for explicit data association or photometric information. We generalize the Space-Sweep approach for the case of a moving event camera by building a virtual camera's DSI [14] containing only geometric information of edges and finding 3D points in it.

First, we review the classical Space-Sweep method for standard cameras, and then we describe our generalization to a moving event camera, showing that the continuous stream of events produced by the sensor is especially relevant to recover 3D structure.

2.1 Classical Space-Sweep Method

In contrast to most classical MVS methods, which rely on pixel intensity values, the Space-Sweep method [3] relies solely on binary edge images (e.g., Canny) of the scene from different viewpoints. Thus, it leverages the sparsity or semi-density of the view-point dependent edge maps to determine 3D structure. More specifically, the method consists of three steps: warping (i.e., back-projecting) image features as rays through a DSI, recording the number of rays that pass through each DSI voxel and, finally, determining whether or not a a 3D point is present in each voxel. The DSI score measures the geometric consistency of edges in a very simple way: each pixel of a warped edge-map onto the DSI votes for the presence or absence of an edge. Then, the DSI score is thresholded to determine the scene points that most likely explain the image edges.



(a) Classical (frame-based) Space-Sweep: only a fixed number of views is available. Two points of an edge map are visible in each image. The intersections of rays obtained by back-projecting the image points are used as evidence for detection of scene features (object points).

(b) Event-Based Space-Sweep: as the event sensor moves, events are triggered on the sensor. To each observed event corresponds a ray (through back-projection), that spans the possible 3D-structure locations. The areas of high ray density correspond to the locations of the two points, and are progressively discovered as the sensor moves (a visual demonstration is provided in the attached video).

Figure 1: Comparison of the back-projection step in classical Space-Sweep and Event-Based Space-Sweep. This is a 2D illustration with the scene consisting of two points.

2.2 Event-Based Space-Sweep Method

In this section, we extend the Space-Sweep algorithm in Section 2.1 to solve EMVS. Notice that the stream of events provided by event cameras is an ideal input to the Space-Sweep algorithm since (*i*) event cameras naturally highlight edges in hardware, and (*ii*) because edges trigger events from *many* consecutive viewpoints rather than a few sparse ones (cf. Fig. 1). Next we detail the three steps of the event-based Space-Sweep method: back-projection, ray-counting, and determining the presence of scene structure.

2.2.1 Feature-Viewing Rays by Event Back-projection

Let us formally define an event $e_k = (x_k, y_k, t_k, p_k)$ as a tuple containing the pixel position (x_k, y_k), timestamp t_k and polarity p_k (i.e., sign) of the brightness change. We extend the Space-Sweep method to the event-based paradigm by using the event stream $\{e_k\}$ output by the DVS as the input point-like features that are warped into the DSI. Each event e_k is back-projected according to the viewpoint of the DVS at time t_k, which is known according to the assumptions of MVS.

From a geometric point of view, we compare the back-projection step in the classical frame-based and the event-based settings using Fig. 1. Observe that in frame-based MVS the number of viewpoints is small compared to that in the highly sampled trajectory of the DVS (at times $\{t_k\}$). This higher abundance of measurements and viewpoints in the event-based setting generates many more viewing rays than in frame-based MVS, and therefore, it facilitates the detection of scene points by analyzing the regions of high ray-densities.

A major advantage of our method is that no explicit data association is needed. This is the main difference between our method and existing event-based depth estimation methods (Section 1.1). While previous works essentially attempt to estimate depth by first solving the stereo correspondence problem in the image plane (using frames of accumulated events [7, 11], temporal correlation of events [2, 6, 8, 10], etc.), our method works directly in the 3D space. This is illustrated in Fig. 1(b): there is no need to associate an event to a particular 3D point to be able to recover its 3D location.

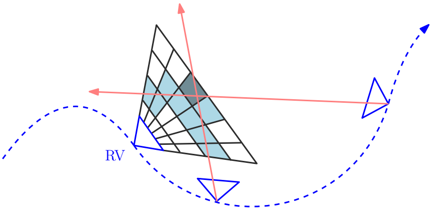


Figure 2: The DSI ray counter is centered at a virtual camera in a reference viewpoint (RV) and its shape is adapted to the perspective projection. Every incoming viewing ray from a back-projected event (in red) votes for all the DSI voxels (in light blue) which it traverses.

2.2.2 Volumetric Ray Counting. Creating the Disparity Space Image (DSI)

In the second step of Space-Sweep, we discretize the volume containing the 3D scene and count the number of viewing rays passing through each voxel using a DSI. To allow for the reconstuction of large scenes in a scalable way, we split the 3D volume containing the scene into smaller 3D volumes along the trajectory of the DVS, compute local 3D reconstructions and then merge them, as will be explained in Section 2.2.4.

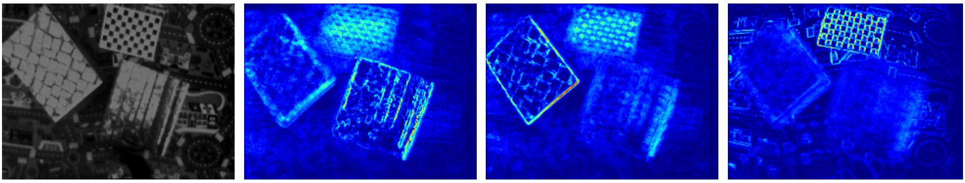
For now, let us focus on computing a local 3D reconstruction of the scene from a subset of events. For this task, we create a virtual camera located at a reference viewpoint that is chosen among those DVS viewpoints associated to the subset of events, and then define a DSI in a volume V adapted to the field of view and perspective projection of the DVS, as illustrated in Fig. 2 (see [14]). The DSI is defined by the DVS pixels and a number N_z of depth planes $\{Z_i\}_{i=1}^{N_z}$, i.e., it has size $w \times h \times N_z$, where w and h are the width and height of the DVS. The score stored in the DSI $f(\mathbf{X}) : V \to \mathbb{R}^+$ is the number of back-projected viewing rays passing through each voxel with center $\mathbf{X} = (X,Y,Z)^\top$, as shown in Fig. 2. Note that the ray-voxel intersections can be computed very efficiently using the two-step technique introduced in [3], allowing for real-time performance on a single CPU.

2.2.3 Detection of Scene Structure by Maximization of Ray Density

In the third step of Space-Sweep, we obtain a semi-dense depth map in the virtual camera by determining whether or not a 3D point is present in each DSI voxel. The decision is taken based on the ray density function stored in the DSI, $f(\mathbf{X})$.

Rephrasing the assumption of the Space-Sweep method [3], scene points are likely to occur at regions where several viewing rays nearly intersect (see Fig. 1(b)), which correspond to regions of high ray density. Hence, scene points are likely to occur at *local maxima* of the ray density function. Fig. 3 shows an example of slicing the DSI from a real dataset at different depths; the presence of local maxima of the ray density function is evidenced by the in-focus areas.

We detect the local maxima of the DSI $f(\mathbf{X})$ following a two-step procedure: we first generate a (dense) depth map $Z^*(x,y)$ in the virtual camera and an associated confidence map $c(x,y)$ by recording the location and magnitude of the best local maximum $f(X(x),Y(y),Z^*) =: c(x,y)$ along the row of voxels in the viewing ray of each pixel (x,y). Then, we select the most confident pixels in the depth map by thresholding the confidence map, yielding a semi-dense depth map (Fig. 4). We use Adaptive Gaussian Thresholding: a pixel (x,y) is selected if $c(x,y) > T(x,y)$, with $T(x,y) = c(x,y) * G_\sigma(x,y) - C$. In practice, we use a 5×5 neighborhood in G_σ and $C = -6$. The adaptive approach yields better results than global



(a) Image at virtual camera. (b) f slice at *close* depth. (c) f slice at *middle* depth. (d) f slice at *far* depth.

Figure 3: The event camera moved above three textured planes located at different depths (close, middle, far). We build the ray density DSI $f(\mathbf{X})$ as described in Section 2.2.2 and show the effect of slicing it at different depths, as simulating a plane sweeping through the DSI. When the sweeping plane coincides with an object plane, the latter appears very sharp while the rest of the scene is "out of focus".

thresholding [3]. A summary of the main elements of our DSI approach is given in Fig. 4.

2.2.4 Merging Depth maps from Multiple Reference Viewpoints

So far, we have shown how to reconstruct the structure of scene corresponding to a subset of the events around a reference view. As pointed out in Section 2.2.2, motivated by a scalable design, this operation is carried out on subsets of the event stream, thus recovering semi-dense depth maps of the scene at multiple *key* reference views. More specifically, we select a new *key* reference view as soon as the distance to the previous *key* reference view exceeds a certain percentage of the mean scene depth, and use the subset of events until the next *key* reference view to estimate the corresponding semi-dense depth map of the scene. The depth maps are then converted to point clouds, cleaned from isolated points (those whose number of neighbors within a given radius is less than a threshold) and merged into a global point cloud using the known positions of the virtual cameras. Other depth map fusion strategies could be implemented. However, such a research topic is out of the scope of this paper. In practice, our approach shows compelling large-scale 3D reconstruction results even without the need for complex fusion methods or regularization.

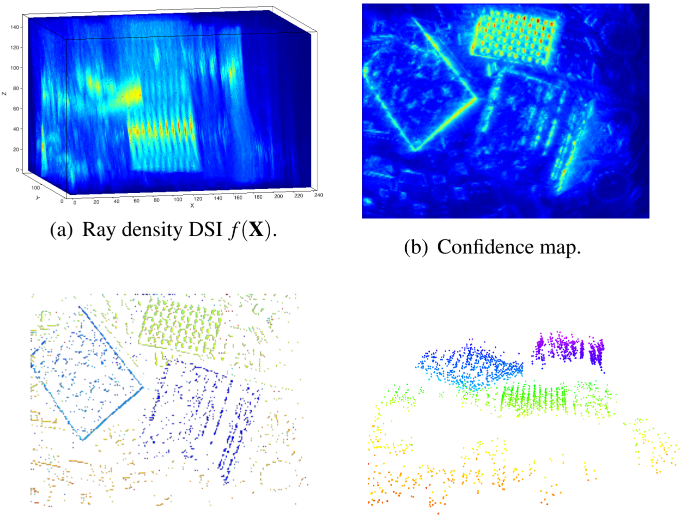
3 Experiments

We now evaluate the performance of our event-based Space Sweep Method, on both synthetic and real datasets.

3.1 Synthetic data

We generated three synthetic datasets with ground truth information by means of an event camera simulator. We set the spatial resolution to 240 × 180 pixels, as that of commercial event sensors. The datasets also contain intensity images along the event camera viewpoints. However, these are not used in our EMVS algorithm; they are solely shown to aid the visualization of the semi-dense depth maps obtained with our method. The datasets exhibit various depth profiles and motions: *Dunes* consists of a smooth surface (two dunes) and a translating and rotating camera in two degrees of freedom (DOF), *3 planes* shows three planes at different depths (i.e., discontinuous depth profile with occlusions) and a linear camera motion; finally, *3 walls* shows a room with three walls (i.e., a smooth depth profile with sharp transitions) and a general, 6-DOF camera motion.

Our EMVS algorithm was executed on each dataset. First, we evaluated the sensitivity of our method with respect to the number of depth planes N_z used to sample the DSI. We



(a) Ray density DSI $f(\mathbf{X})$.

(b) Confidence map.

(c) Semi-dense depth map.

(d) 3D point cloud.

Figure 4: Our method builds the ray density DSI (a), from which a confidence map (b) and a semi-dense depth map (c) are extracted in a virtual camera. The semi-dense depth map gives a point cloud of scene edges (d). Same dataset as in Fig. 3.

Table 1: Depth estimation accuracy in the synthetic datasets ($N_z = 100$)

	Dunes	3 planes	3 walls
Depth range	3.00 m	1.30 m	7.60 m
Mean error	0.14 m	0.15 m	0.52 m
Relative error	4.63%	11.31%	6.86%

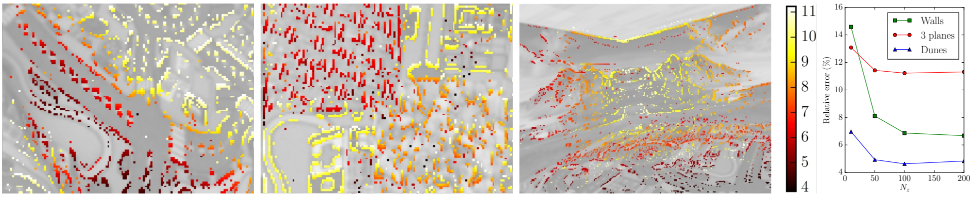
used depth instead of inverse depth in the DSI since it provided better results in scenes with finite depth variations. Fig. 5(d) shows, as a function of N_z, the relative depth error, which is defined as the mean depth error (between the estimated depth map and the ground truth) divided by the depth range of the scene. As expected, the error decreases with N_z, but it stagnates for moderate values of N_z. Hence, from then on, we fixed $N_z = 100$ depth planes. Table. 1 reports the mean depth error of the estimated 3D points, as well as the relative depth error for all three datasets. Depth errors are small, in the order of 10% or less, showing the good performance of our EMVS algorithm and its ability to handle occlusions and a variety of surfaces and camera motions.

3.2 Real data

We also evaluated the performance of our EMVS algorithm on datasets from a DAVIS sensor [1]. The DAVIS outputs, in addition to the event stream, intensity frames as those of a standard camera, at low frame rate (24 Hz). [1] However, our EMVS algorithm does not use the frames; they are displayed here only to illustrate the semi-dense results of the method.

We considered two methods to provide our EMVS algorithm with camera pose information: a motorized linear slider or a visual odometry algorithm on the DAVIS frames. We used the motorized slider to analyze the performance in controlled experiments (since it guaran-

[1] The DAVIS comprises both a frame camera and an event sensor (DVS) in the same pixel array, of size 240 × 180.



(a) Dunes (b) 3 planes (c) 3 walls (d) Depth error.

Figure 5: Synthetic experiments: estimated semi-dense depth maps overlayed over screenshots of the scene, in three datasets (a)-(c). Depth is colored, from close (red) to far (yellow). Our EMVS algorithm successfully recovers most edges, even without regularization or outlier filtering. (d): Relative depth error as a number of depth planes N_z, in all three datasets.

Table 2: Depth estimation accuracy in the HDR experiment

	Close (distance: 23.1 cm)		Far (distance: 58.5 cm)	
Illumination	Mean error	Relative error	Mean error	Relative error
○ constant	1.22 cm	5.29%	2.01 cm	4.33%
○ HDR	1.21 cm	5.25%	1.87 cm	3.44%

tees very accurate pose information) and a visual odometry algorithm (SVO [4]) to show the applicability of our method in hand-held (i.e., unconstrained) 6-DOF motions.

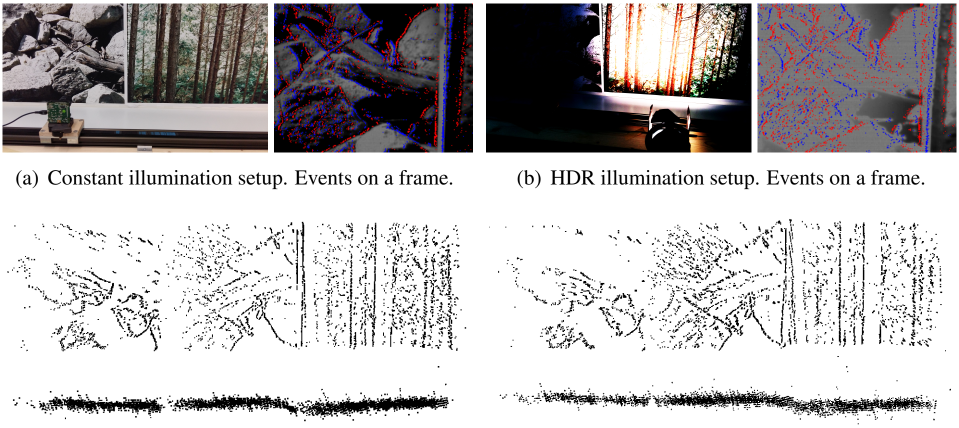
3.2.1 High Dynamic Range and High-Speed Experiments

In this section, we show that our EMVS algorithm is able to recover accurate semi-dense structure in two challenging scenarios, namely (i) high-dynamic-range (HDR) illumination conditions and (ii) high-speed motion. For this, we place the DAVIS on the motorized linear slider, facing a textured wall at a known constant depth from the sensor. In both experiments, we measure the accuracy of our semi-dense maps against ground truth and demonstrate compelling depth estimation accuracy, in the order of 5% of relative error, which is very high, especially considering the low resolution of the sensor (only 240×180 pixels).

High Dynamic Range Experiment. We recorded two datasets under the same acquisition conditions except for illumination (Fig. 6): first with constant illumination throughout the scene and, second, with a powerful lamp illuminating only half of the scene. In the latter case, a standard camera cannot cope with the wide intensity variation in the middle of the scene since some areas of the images are under-exposed while others are over-exposed. We performed the HDR experiment with two different wall distances (close and far).

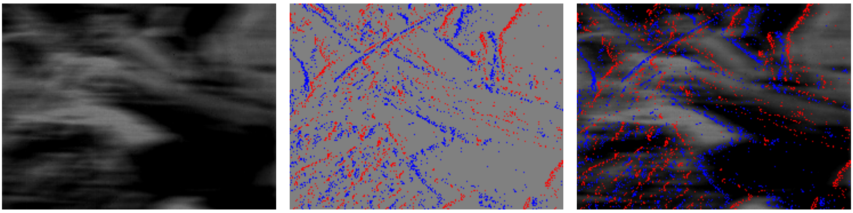
The results of our EMVS algorithm are given in Fig. 6 and Table 2. Observe that the quality of the reconstruction is unaffected by the illumination conditions. In both cases, the EMVS method has a very high accuracy (mean relative error $\approx 5\%$), and also in spite of the low spatial resolution of the sensor or the lack of regularization. Moreover, observe that the accuracy is not affected by the illumination conditions. Hence, we unlocked the high-dynamic range capabilities of the sensor to demonstrate successful HDR depth estimation.

High-Speed Experiment. To show that we can exploit the high-speed capabilities of the event sensor for 3D reconstruction, we recorded a dataset with the DAVIS at 40.5 cm from the wall and moving at 0.45 m/s. This translated into an apparent speed of 376 pixels/s in the image plane, which caused motion blur in the DAVIS frames (Fig. 7). The motion blur makes the visual information unintelligible. By contrast, the high temporal resolution of the event stream still accurately captures the edge information of the scene. Our EMVS



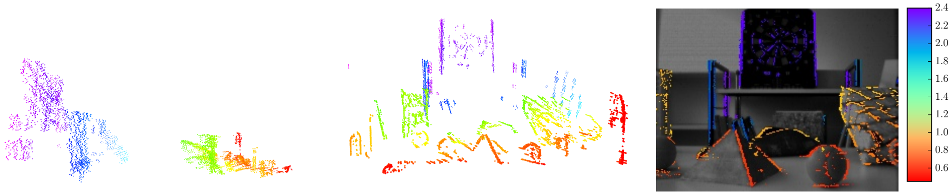
(a) Constant illumination setup. Events on a frame.
(b) HDR illumination setup. Events on a frame.
(c) Constant illum. 3D points: front and top views.
(d) HDR illum. 3D points: front and top views.

Figure 6: *HDR experiment*: Top: Scene and illumination setups, with the DAVIS on the motorized linear slider (a) and a lamp (b). Sample frames show under- and over-exposed levels in HDR illumination (b). By contrast, the events (overlayed on the frames) are unaffected, due to the high dynamic range of the event sensor. Bottom: reconstructed point clouds.



(a) Frame (motion blur).
(b) Events ($\Delta t = 2ms$).
(c) Frame and events.

Figure 7: *High-speed experiment*. Frame and the events from the DAVIS at 376 pixels/s. The frame suffers from motion blur, while the events do not, thus preserving the visual content.



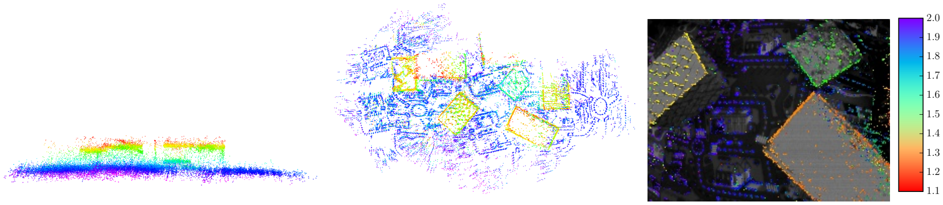
(a) Side view.
(b) Front view.
(c) Projection on a frame.

Figure 8: *Desk dataset*: scene with objects and occlusions.

method produced a 3D reconstruction with a mean depth error of 1.26 cm and a relative error of 4.84%. The accuracy is consistent with that of previous experiments ($\approx 5\%$), thus supporting the remarkable performance of our method and its capability to exploit the high-speed characteristics of the event sensor.

3.2.2 Three-dimensional Scenes

Figs. 8 and 9 show some results obtained by our EMVS method on non-flat scenes. We show both the semi-dense point cloud and its projection on a frame (for better understanding). To



(a) Side view.
(b) Top view.
(c) Projection on a frame.

Figure 9: *Boxes dataset:* large-scale semi-dense 3D reconstruction with a hand-held DAVIS.

ease the visualization, depth is colored from red (close) to blue (far).

In Fig. 8, the DAVIS moves in front of a scene containing various objects with different shapes and at different depths. In spite of the large occlusions of the distant objects, generated by the foreground objects, our EMVS algorithm is able to recover the structure of the scene reliably. Finally, Fig. 9 shows the result of our EMVS algorithm on a larger scale dataset. The sensor was hand-held moved in a big room featuring various textured boxes. Multiple local point clouds are estimated along the trajectory, which are then merged into a global, large-scale 3D reconstruction.

4 Discussion

This work has focused on multi view stereo with a single moving event camera. Our goal was to show that 3D reconstruction with a single event camera is possible, and that we do not need to solve the data association problem. The results showed that (*i*) the method provides accurate results, being able to unlock the capabilities of the sensor in challenging scenarios (HDR and high-speed) where standard cameras fail, and (*ii*) the method can handle inaccurate poses (the experiment with poses provided by a frame-based visual odometry algorithm shows visually appealing results, which suggests that the method is robust to pose uncertainty). The applicability of multi view stereo depends on the availability of pose information, which in our experiments was provided by an external tracking system. However, this is not a limitation, since the method could be extended to operate in combination with an event-based motion estimation algorithm, such as [5], thus removing the need for an external pose estimator.

5 Conclusion

We introduced the EMVS problem, and provided a simple and elegant solution to it that exploits the natural strengths of the sensor, and runs in real-time on a CPU. We validated our algorithm on both synthetic and real data, for various motions and scenes, showing very accurate 3D reconstructions (relative depth error of 5%) in spite of the low resolution of the sensor and the high amount of noise typical of event cameras. We believe this work is a major step towards building 3D reconstruction algorithms robust to speed (the events do not suffer from motion blur), and HDR illumination. This paper further highlights the potential of event cameras and the astounding possibilities it opens to computer vision.

Acknowledgement. We thank Elias Mueggler for helping recording the data. This research was supported by the National Centre of Competence in Research Robotics (NCCR) and the UZH Forschungskredit.

References

[1] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck. A 240x180 130dB 3us latency global shutter spatiotemporal vision sensor. *IEEE J. of Solid-State Circuits*, 49 (10):2333–2341, 2014.

[2] L.A. Camunas-Mesa, T. Serrano-Gotarredona, S.-H. Ieng, R. Benosman, and B. Linares-Barranco. On the use of Orientation Filters for 3D Reconstruction in Event-Driven Stereo Vision. *Front. Neurosci.*, 8(48), 2014.

[3] R. T. Collins. A space-sweep approach to true multi-image matching. In *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 358–363, Jun 1996.

[4] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 15–22, 2014.

[5] G. Gallego, J.E.A. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza. Event-based, 6-DOF Camera Tracking for High-Speed Applications . arXiv:1607.03468, July 2016.

[6] J. Kogler, M. Humenberger, and C. Sulzbachner. Event-Based Stereo Matching Approaches for Frameless Address Event Stereo Data. In *Advances in Visual Computing*, volume 6938 of *Lecture Notes in Computer Science*, pages 674–685. Springer, 2011.

[7] J. Kogler, C. Sulzbachner, M. Humenberger, and F. Eibensteiner. Address-Event Based Stereo Vision with Bio-Inspired Silicon Retina Imagers. In *Advances in Theory and Applications of Stereo Vision*, pages 165–188. InTech, 2011.

[8] J. Lee, T. Delbruck, P. Park, M. Pfeiffer, C. Shin, H. Ryu, and B. C. Kang. Gesture-based remote control using stereo pair of dynamic vision sensors. In *Int. Conf. on Circuits and Systems (ISCAS)*, 2012.

[9] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. of Solid-State Circuits*, 43(2):566–576, 2008.

[10] P. Rogister, R. Benosman, S.-H. Ieng, P. Lichtsteiner, and T. Delbruck. Asynchronous Event-Based Binocular Stereo Matching. *IEEE Trans. Neural Networks and Learning Systems*, 23(2):347–353, Feb 2012.

[11] S. Schraml, A.N. Belbachir, N. Milosevic, and P. Schön. Dynamic stereo vision system for real-time tracking. In *Int. Conf. on Circuits and Systems (ISCAS)*, 2010.

[12] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2006.

[13] R. Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. Springer, 2010.

[14] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *Int. J. Comput. Vis.*, 32(1):45–61, 1999.