Robust Dense Depth Map Estimation from Sparse DVS Stereos

Dongqing Zou¹

dongqing.zou@samsung.com

Feng Shi¹

f0815.shi@samsung.com

Weiheng Liu¹

weiheng.liu@samsung.com

Jia Li¹

j0402.li@samsung.com

Qiang Wang¹

qiang.w@samsung.com

Paul-K.J. Park²

k.j.park@samsung.com

Chang-Woo Shi²

changwoo.shin@samsung.com

Yohan J. Roh²

yohan.roh@samsung.com

Hyunsurk Eric Ryu²

eric_ryu@samsung.com

¹ SAIT China Lab

SRC-Beijing

Samsung Electronics

Beijing, China

² LSI Product and Technology

System LSI Business

Samsung Electronics

South Korea

Abstract

Real-world depth perception applications require precise reaction to fast motion, and the ability to operate in scenes which contain large intensity differences or high dynamic range. Standard CMOS cameras based methods for depth computing, such as stereo matching, run into the problem of huge power consuming at high frame-rates or inaccurate depths with noise or holes. The event camera, DVS (Dynamic Vision Sensor), aims to be robust to fast motion and light change with low power consumption and sparse representation, offering great potential to replace standard cameras for depth perception. However, it is not trivial to directly apply DVS for stereo matching due to its nature of low latency and sparsity which will result in extremely limited available information and imperfect imaging quality. To overcome these problems and make the DVS available for depth perception, this paper introduces a novel method which can enhance the stream of events and estimate the dense depth through event driven stereo matching. Our event stream enhancement algorithm efficiently buffers events according to time continuous rather than using artificially-chosen time intervals, and our stereo matching method can robust estimate depth for complex scenarios regardless of the motion or light changing. To the best of our knowledge, this is the first algorithm provably able to recover dense depth maps from sparse DVS stereos. Experiments in a variety of challenging conditions demonstrate the superior performance of our method.

1 Introduction

Stereo matching is one of the important methods to compute depth and is of great utility for a lot of applications such as autonomous driving, augmented reality and navigation. Though tremendous strides have been made in stereo matching with standard CMOS cameras, they still suffer from motion blur and high dynamic range conditions with strong illumination changes. A novel depth perception method which can resolve such challenges is urgently needed to boost the development of technologies on depth perception.

In this paper, we propose to generate dense depth maps with Dynamic Vision Sensors (DVS) [1] and seek to solve the challenges in traditional stereo matching methods. DVS are a kind of biologically inspired sensors, which have the advantage of large High-Dynamic Range (≥ 120 dB) and low power consumption (≤ 10 mw). Moreover, DVS cameras record all pixel-wise intensity changes of a scene. Ideally, the scene's information, such as geometry and intensity, are either recorded directly by events, or can be implicitly inferred using the related events in current or other moments [2, 3]. Thus DVS offer great potential for dense depth estimation and stereo matching.

Despite the attractive qualities of DVS, it is very difficult to directly apply DVS for stereo matching and dense depth estimation due to the property of sparse representation and asynchronous nature. DVS only transmitted the local pixel-level changes caused by contrast changes in a scene, the imaging results from DVS will inevitably lose large number of information that exists in scenes with similar appearance. Moreover, DVS cameras sent singles asynchronously, buffering DVS data with a fixed time interval will result in over-blur or unclear structures. Furthermore, DVS events always accompany ambiguity of ownership since they are usually triggered at the edges of objects, making 3D geometry estimation intractable.

On the other hand, since the stream of DVS is fundamentally different from the image streams of standard CMOS cameras, new algorithms are required to deal with this data for achieving better performance of dense depth maps estimation. The quality of dense depth maps depend on the accuracy of DVS stereo matching, while the performance of stereo matching is affected by the ability of event descriptors. Thus, powerful event descriptors will lead to accurate dense depths. Unfortunately, existing features for standard COMS and naive DVS features are not good enough to support accurate DVS stereo matching.

Hence, a desired event-driven dense map generation method should overcome the inherent problems of DVS, while can accurately identify the matched data between left and right DVS cameras so that the depth of all DVS data can be obtained through stereo matching. In this paper, one of the major contributions lies in proposing a powerful event feature descriptor that is invariant to event maps *translation, scale, and rotation*. This event feature facilitates the stereo matching for DVS and dense depth maps generation from stereo DVS.

Another contribution of this paper is a DVS data enhancement algorithm that outperforms those methods based on the accumulation of events over fixed, artificially-chosen time intervals. Due to the low latency of DVS, the generated stream of events gives a continuous representation in time. So the gradient values of timestamps that events trigged hint how long the each active event lasts. We develop a *realtime* algorithm to enhance the DVS events by adaptively clustering events according to the timestamp gradients of events. We believe that our method will activate the development of many applications such as autonomous driving, augmented reality and navigation.

2 Related Work

We review the related work from three aspects: Dynamic Vision Sensors, Event enhancement, and event driven stereo matching.

DVS [10] are a kind of biology inspired sensors which send signal when undergoing light illumination changes. They generate an ‘on’ signal when a pixel has a higher light than the previous status while they send an ‘off’ signal when the light is lower. The signals are named as events. Events can be described by a tuple (u, v, t, p) , where u, v are the pixel coordinates of the event, t is the timestamp of the event, and $p \in \{-1, +1\}$ is the polarity of the event. An image containing the events within a period of integral time is called event map.

Mueggler *et al.* [11] developed an event enhancement algorithm that augments each event with its lifetime by buffering a certain number of past and future events. This method assume locally constant velocities thus it can not be used for those DVS data trigged by the light flicking. Besides, this method solves the velocities with RANSAC algorithm, thus it is inefficient. In [12], the events in a sliding window of fixed duration were selected to generate an event image. However the fixed time intervals will lead to either blurred event stream or unclear structures. In this paper, we proposed a real-time event stream enhancement method with Integral Images [13] whose complexity is linear to the number of events.

Some stereo matching approaches [14, 15, 16, 19] for DVS data are published in the last decade. Schraml *et al.* [16] proposed an area-based stereo matching approach and demonstrated in a real-time tracking application. Rogister *et al.* [15] assumed that correlated events are likely to appear within a small time interval and on the same epipolar line and proposed a stereo matching method by using two static DVS. Piatkowska *et al.* [14] presented a dynamic network based stereo matching method by using the history of events. Later, Eibensteiner *et al.* [19] proposed a high performance event-driven matching approach based on time-correlation. However, these methods fail in handling the data with a large time jitter. Schraml *et al.* [17] proposed a method to solve the problem of finding corresponding events in the time domain by defining a novel cost measure, but failed in processing DVS data with similar event distributions in the horizontal direction. In this paper, we propose a novel event feature descriptor for DVS, which has more powerful representative ability, thus can obtain more accurate event depth than existing works.

3 Event Stream Enhancement

DVS outputs not a sequence of video frames like a standard camera, but a stream of asynchronous spikes, indicating when individual pixels record a threshold log intensity change. Previous method [18] accumulating the events depending on a fixed time interval. However, with a fixed accumulation interval, the images will either be over blurred if the integral time is set large, or some structures are not clearly visible if given a little integral time. More smart strategy should be adopted to calculate the time interval within which this event is considered active.

Generally speaking, blur comes out if the time interval is larger than the time that an event takes to move to its neighboring pixel. Thus the time interval can be easily obtained if we know the event’s velocity $\mathbf{v} = (v_x, v_y)$. In theory, the velocity of events can be obtained by calculating the gradient of timestamp image, as [2] did. Specifically, let’s define the event as $e(\mathbf{p}, t) = (\mathbf{p}, t)$, and $\mathbf{p} = (u, v)$ denotes the position and t denotes the time of an event. A function Σ_e mapping the $\mathbf{p}$ to t can be defined as $\Sigma_e : R^2 \rightarrow R, t = \Sigma_e(\mathbf{p})$. The inverse of velocity can be obtained by computing the first partial derivatives with respect to the location: $\nabla \Sigma_e = (1/v_x, 1/v_y) = (\partial \Sigma_e / \partial x, \partial \Sigma_e / \partial y)$. The time interval that the event maintains active

Figure 1: Events stream enhancement. (a) is the original scene. (b) and (c) are images with fixed accumulation intervals of 20 ms and 1 ms. (d) is our result.

is $\tau(\mathbf{p}) = \max(\nabla \sum_e(\mathbf{p}), \Delta \mathbf{p})$ subjecting to $\|\Delta \mathbf{p}\| = 1$. Here, $\Delta \mathbf{p}$ means the displacement of an event.

However, the velocity computed according to the gradient of timestamps only reflects the true velocity of events triggered by motion but not by light changes or flickering. These events triggered by flickering should be filtered out so that the time interval can be estimated more accurately. Only those events whose timestamp are larger than the timestamp of their neighboring events over a threshold η can be used for time interval.

Therefore, we first filter out the events triggered by flickering and then estimate the event's velocity and time interval. According to the work [24], the velocity of an event is proportional to the slope of the fitted plane formed by using the timestamps of its neighboring events. Supposing that each event is represented as a tuple (x, y, t) , and let $\mathbf{P}_i$ be the local neighborhoods of the event p_i . Given the covariance matrix $C_i \in R^{3 \times 3}$ of $\mathbf{P}_i$, the normal of the local plane is the eigenvector corresponding to the smallest eigenvalue. The work [24] constructs the covariance matrix through a brute force searching, which is time consuming. To achieve real-time performance, we employ the Integral Images [24] to accelerate the computation of normals of events buffering in duration $[t - \Delta t, t + \Delta t]$. Specifically, the covariance matrix between any two channels, e.g., (x, y) , of the event's vector can be defined as:

$$C(x, y) = \frac{1}{n-1} \left[\sum_{i,j=1}^n e_x(i)e_y(j) - \frac{1}{n} \sum_{i=1}^n e_x(i) \sum_{j=1}^n e_y(j) \right]. \quad (1)$$

The computation of the sum of each feature dimension $e(i)_{i=1..3}$ as well as the sum of the multiplication of any two feature dimensions can be efficiently implemented by using a series of integral images whose complexity is linear in the number of events. As a result, we can get the normals in real time. For robust, we average the time interval of the first one hundred of events triggered by motion as the final time interval to accumulate events for robustness. Figure 1 shows the result by our method.

4 Event Descriptor

With the enhanced event stream, we now introduce an event descriptor that supports the event-driven stereo matching.

According to the principle of DVS imaging, events often come out where the large contrast take place in a scene. Previous research in computer vision [25] suggested that the context can dramatically improve the performance in object matching. This inspires us to describe each event by leveraging the local context of the events. Hausdorff et al. [26] proposed a distance transform method to describe the context of a pixel in binary images by

calculating the distance from each “off” pixel to the nearest “on” pixels. Intuitively, if we treat all events as “on” pixels in the event map, this algorithm can be directly transferred to describe the local event context.

Unfortunately, due to slightly different sensitivity of the two event cameras equipped in the device of stereo DVS, the events distributions in final imaging results from the left and right event cameras are different from each other. Such slight difference in event distributions might yield different distance transform results, resulting in bad stereo matching results. More smart strategy, such as the use of multidimensional histograms summarizing the distribution of measurements within image regions, should be taken into consideration to make event descriptors more stable. Moreover, the measurements of distance transform are unsigned values that cannot reflect the spatial information well.

We thereby propose to describe the local context of events by considering the spatial distribution of events in the *signed (vector) distance filed*. Specifically, we first separate an event map into two event maps according to the polarities of events, obtaining both positive and negative event maps, as Figure 2(a) shows. Let $E^s = \{\dots, p^s(i, j), \dots, p^s(i, q), \dots\}$ be the input event map where s denotes the polarity and the $p^s(i, j)$ denotes a pixel at location (i, j) . A pixel with event is “on” pixel which subjects to $1(p^s(i, j)) = 1$, 0 otherwise. The feature map of an event map is defined as:

$$\mathbf{D}(p^s(i, j)) = \{(u_{ij}^s, v_{ij}^s) | p^s(i, j) \in E^s\}, \quad (2)$$

where (u_{ij}^s, v_{ij}^s) is the vector pointing to the nearest event which can be quickly searched by distance transform method [4], as the Figure 2 (b) shows.

With the Equation (2), two feature maps can be obtained from positive and negative event maps respectively. As a result, each pixel without event contains two feature vectors which index the nearest positive event and negative event, respectively. With these two feature maps, for an event $p^s(i, j)$, we collect the vectors of pixels in a window centering at (i, j) , and assign each neighboring vector into one bin of an orientation histogram with 12 bins according to the orientation of each vector, as the Figure 2(c) shows. Each bin in the histogram covers 30 degree, and the peaks in the histogram correspond to the dominant orientation. Once the histogram is filled, we normalize the histogram by dividing each bin by the sum of all pixels in the histogram. The final descriptor $\mathbf{B}(p^s(i, j))$ for each event is thus a 24 dimension vector by concatenating the 12 bins in the histogram for the positive map and 12 bins for the negative event map, as shown in Figure 2(d). Obviously, our descriptor is represented relative to the orientation and therefore achieves invariance to event map rotation.

One parameter in the descriptor computation is the size of window that decides where neighboring events are collected. Generally, a large window will introduce noised events while a small window can not capture enough contextual information. We adaptively determined the size of window according to the density of events in an event map. If the density is large, the effective vectors will reduce because each event treat itself as its nearest event, which make the distinctive ability of event descriptor weaken. In contrast, too few events in a window cannot describe the local context or distribution of events in a local region. We thus constrain the window size w to subject to $\frac{n}{(w \times w)} \geq 0.3 \& n > 5$, where n denotes the number of events in the given window. Our descriptor is represented relative to the window size and therefore achieves invariance to event map scale. Figure 3 shows some matching results by using our feature descriptor. From the matching results we can see that our event descriptor is robust to rotation and scaling.

Figure 2: Event descriptor calculation. (a) is the input event map, (b) is vector feature map for a patch of event map, and black blocks represent events. (c) is the histogram of vectors.

Figure 3: Event matching with our descriptor in three cases. (a) translation; (b) rotation, and (c) scaling.

5 Stereo Matching and Dense Depth Maps Generation

5.1 Event-Driven Stereo Matching

With the proposed event feature descriptor, event driven stereo matching is carried out by orderly performing the cost matching, cost aggregation, and disparity estimation.

Matching cost of an event p at disparity d is computed as:

$$C(p, d) = \sum_{q \in N_p} |\mathbf{B}^L(q) - \mathbf{B}^R(qd)|, \quad (3)$$

where $qd = (x - d, y)$ if the coordinate of q is (x, y) . N_p is the set of pixels within a fixed rectangular window centered at p .

Cost aggregation is performed by averaging over a fixed region through

$$C_A^0(p, d) = C(p, d); C_A^i(p, d) = \sum_{q \in N_p} C_A^{i-1}(q, d), \quad (4)$$

where i is the iteration number, and the number of repeat time is 5 for all our experiments.

Disparity estimation is calculated by minimizing the sum of cost with the calculated cost matrix. We estimate disparity map $D_e(p)$ through the winner-take-all strategy, that is,

Figure 4: Surrounding neighbors searching. (a) is the original RGB image and (b) is the corresponding DVS event map. (c) shows the surrounding events of pixel ‘A’ from left sides, (d), (e) and (f) show the neighboring events searched from right, top and bottom, respectively.

finding the disparity d for each event so as to minimize $C_A(p, d)$:

$$D_e(p) = \arg \min_d E(p) * C_A(p, d), \quad (5)$$

where $E(p) = 1$ if p is an event in event map, and $E(p) = 0$ otherwise.

5.2 Dense Depth Generation from Sparse Stereo

According to the principle of DVS imaging, events usually appear on the edges of objects, and a pixel without event usually lies in the same surface with its neighboring events, as the Figure 4 (b) shows. The depth of events on an edge usually are continuous and not change steeply. So if pixels and their neighboring events are in the same surface, the depths of most of these events and pixels in the same surface form a manifold, which approximately obey a model of 3D plane. For example, as shown in Figure 4 (b), the pixel ‘A’ locates in the surface of the ‘Fan’, while the pixel ‘B’ belongs to the surface of ‘Wall’. Therefore, if we can obtain these events which lie in the same surface with the target pixel, the depth of each pixel can be estimated.

Therefore, we first find neighboring events for each pixel, then select out the events that lie in the same surface with the target pixel. To efficiently collect neighboring events and reduce number of the noised events, we search the neighboring events through a novel strategy. Specifically, for each pixel located at (x_i, y_i) , taking pixel ‘A’ in Figure 4(b) as an example, we first search two nearest events $\{e_l = (x_l, y_l), e_r = (x_r, y_r) | x_l < x_i, x_r > x_i\}$ in horizontal orientation and two nearest events $\{e_t = (x_i, y_t), e_b = (x_i, y_b) | y_t < y_i, y_b > y_i\}$ in vertical orientation. Two perpendicular line segments passing the target pixel ‘A’ can be determined with these two paired events, as Figure 4(b) shows. For each pixel on the horizontal (vertical) line, we search two nearest events from top (left) and bottom (right). As a result, the collected events sever as neighboring events. Assuming that $\{e_1^a, e_2^a, \dots, e_k^a\}$ represents the set of neighboring events for the pixel ‘A’, and $\{C(e_1^a), \dots, C(e_i^a), \dots, C(e_k^a)\}$ denotes the coordinates of these surrounding events in the world coordinate system, and $C(e_i^a) = (X_i^a, Y_i^a, Z_i^a)$. The function of the 3D plane can be estimated by minimizing:

$$\min \sum_{i=1}^k |AX_i^a + BY_i^a + CZ_i^a + D| (A^2 + B^2 + C^2)^{-1/2},$$

where $AX + BY + CZ + D = 0$ is the function of 3D plane, and variables A, B, C and D are parameters to be solved. We apply the Integral Images [14] based method presented in Section 3 to efficiently solve the equation above.

Figure 5: A comparison. Events with disparity error more than 4 pixels are plotted in black.

	EDS[	FCVF [	Libelas[	Ours
Avgerr	7.93	2.14	10.07	1.23
A50	2.29	0.34	0.28	0.25
A90	21.15	4.39	49.0	2.05
Bad 2.0	49.1%	14.39%	22.35%	10.0%
Bad 4.0	37.6%	9.78%	21.44%	5.8%

Table 1: Errors for each of the evaluated methods. Our method yields the highest accuracy.

6 Experiments

We test the performance of our method on various dataset, and compare it to the state-of-the-art methods to demonstrate the effectiveness of our method. Our method was performed on a PC with Intel Core i7 Quad processor (3.40GHz, 4 cores) and 20 GB memory. The running time for an event map is about 25 ms.

6.1 Stereo Matching via Descriptor

We first demonstrate the effectiveness of the stereo matching module by comparing with traditional stereo matching methods “Fast Cost-Volume Filtering” (FCVF) [ ], “Libelas” method [ ], and event driven stereo matching method of EDS [ ]. For fairly comparing the stereo matching performance, we run the FCVF [ ] and Libelas [ ] on the RGB images so that these two methods can achieve the best accuracy, and only the depth of pixels with events are selected for comparison.

To quantitatively evaluate the performance of each method, a dataset containing 100 pairs of event maps with groundtruth is constructed for evaluation. The dataset is generated with Kinect and masked by the corresponding reference event map from DVS. We adopt various metrics from Middlebury [ ] to quantitatively compare our method to related methods, as shown in Table 1. In the Table 1, “Avgerr” reflects the average of absolution disparity errors for all events, “AX” reflects the disparity error at the quantiles of “X%”, while “Bad N ” reflects the percent of events with disparity error more than N pixels. From the Table 1 we can see that our method yields the highest accuracy on most of metrics. Particularly, from the errors on metrics of “A50” and “A90”, we can see that while the errors of EDS [ ], FCVF [ ] and Libelas [ ] increase rapidly, our method remains high accuracy almost all the time. This experimental result reflects that the proposed event descriptor is highly distinguishable. Figure 5 further shows an example of comparison in which events with disparity error more than 4 pixels are plotted in black. Our method outperforms other three methods.

6.2 Dense Depth Maps Estimation

We now demonstrate the robustness of our method for dense depth maps estimation.

Figure 6: A result of dense depth map generation for extremely low light scene. (a) is the original scenario captured by CMOS camera, (b) is the DVS event map. (c) and (d) show our result and groundtruth, respectively.

Figure 7: Comparisons between using and without using event enhancement. (a) and (b) are input event depth map without using and using event enhancement, (c) and (d) are the corresponding dense depth maps.

Inclement illumination conditions. The quality of standard RGB images degrade largely when capturing under low-lighting condition due to the insufficient of the ambient light. However, the DVS is large dynamic range, which can robustly response to the relative change of intensity under low lighting. Thus our method is workable in extra low light conditions to estimate dense depth maps. Figure 6 shows a result of dense depth map generation in low light condition.

Event Enhancement for Dense depth estimation. Figure 7 shows a comparison between using and without using event fusion for dense depth maps estimation. Obviously, the geometry structure of the scene after event fusion is more complete than that of without using. As a result, the depth values of the dense depth map from event fusion map are more locally consistent, as the Figure 7 (c) and (d) show.

Depth ambiguity elimination. Recovering the dense depth map from sparse event map is inherently ambiguous for missing priors that the object each event belongs to. Our method estimates the dense depth map by considering both the principle of DVS imaging and the depths consensus of surrounding events of each pixel. The experimental results from our method and the linear interpolation are shown in Figure 8. It is to be noted that the original

Figure 8: Depth estimation of the complex scene. (a) shows the dense depth map from linear interpolation and the tendency of depth variation in the select row. (b) shows our result.

scene contains lots of occlusions, which leads to an over-smooth dense depth through linear interpolation. The right of Figure 8(a) illuminates the tendency of depth variation in the selected row of the final dense depth map. From the Figure 8(b) we can see that the depth values variation is conformed with the real cases.

7 Conclusion

In this paper, a robust method is proposed to estimate dense depth maps from stereo event maps, providing a new way to perceive depth. The proposed event descriptor can improve the accuracy for events matching, and the event stream enhancement algorithm can effectively improve the quality of event maps. Experiments demonstrate that our method can effectively generate dense depth maps in complex and high dynamic range environments.

References

- [1] Middlebury stereo evaluation. <http://vision.middlebury.edu/stereo/eval3/>.
- [2] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE Trans. Neural Netw. Learning Syst.*, 25(2):407–417, 2014.
- [3] F. Eibensteiner, J. Kogler, and J. Scharinger. A high-performance hardware architecture for a frameless stereo vision algorithm implemented on a fpga platform. In *IEEE CVPRW*, pages 637–644, 2014.
- [4] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 8(1):415–428, 2012.
- [5] Andreas Geiger, Martin Roser, and Raquel Urtasun. *Efficient Large-Scale Stereo Matching*. Springer Berlin Heidelberg, 2010.
- [6] A. Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Software Engineering*, 35(2):504–511, 2012.
- [7] Peiyun Hu and Deva Ramanan. Finding tiny faces. *CoRR*, abs/1612.04402, 2016. URL <http://arxiv.org/abs/1612.04402>.
- [8] Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *ECCV*, pages 49–364, 2016.
- [9] Jia Li, Feng Shi, Weiheng Liu, Dongqing Zou, Qiang Wang, Hyunku Lee, Paul-K.J. Park, and Hyunsurk Eric Ryu. Adaptive temporal pooling for object detection using dynamic vision sensor. In *BMVC*, 2017.
- [10] P. Lichtsteiner, T. Delbruck, and J. Kramer. Improved on/off temporally differentiating address-event imager. In *IEEE ICECS*, pages 211–214, 2004.
- [11] Elias Mueggler, Christian Forster, Nathan Baumli, Guillermo Gallego, and Davide Scaramuzza. Lifetime estimation of events from dynamic vision sensors. In *IEEE International Conference on Robotics and Automation*, pages 4874–4881, 2015.

- [12] Zhenjiang Ni, A Bolopion, J Agnus, and R Benosman. Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics. *IEEE Transactions on Robotics*, 28(5):1081–1089, 2012.
- [13] Ewa Piatkowska, Ahmed Nabil Belbachir, and Margrit Gelautz. Asynchronous stereo vision for event-driven dynamic stereo sensor using an adaptive cooperative approach. In *IEEE ICCVW*, pages 45–50, 2013.
- [14] Fatih Porikli and Oncel Tuzel. Fast construction of covariance matrices for arbitrary size image windows. In *IEEE International Conference on Image Processing*, pages 1581–1584, 2007.
- [15] P. Rogister, Ryad Benosman, Sio Hoi Ieng, P. Lichtsteiner, and Tobi Delbruck. Asynchronous event-based binocular stereo matching. *IEEE Transactions on Neural Networks & Learning Systems*, 23(2):347–353, 2012.
- [16] S. Schraml, A. N. Belbachir, N. Milosevic, and P. SchoN. Dynamic stereo vision system for real-time tracking. In *IEEE ISCAS*, pages 1409–1412, 2010.
- [17] Stephan Schraml, Ahmed Nabil Belbachir, and Horst Bischof. Event-driven stereo matching for real-time 3d panoramic vision. In *IEEE CVPR*, pages 466–474, 2015.
- [18] Olga Veksler. Fast variable window for stereo correspondence using integral images. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, pages I–556–I–561 vol.1, 2003.
- [19] Dongqing Zou, Ping Guo, Qiang Wang, Xiaotao Wang, Guangqi Shao, Feng Shi, Jia Li, and Paul K. J. Park. Context-aware event-driven stereo matching. In *IEEE International Conference on Image Processing*, pages 1076–1080, 2016.