A High-Performance Hardware Architecture for a Frameless Stereo Vision Algorithm Implemented on a FPGA Platform

Florian Eibensteiner
Upper Austria University of Applied Sciences
Softwarepark 11, 4232-Hagenberg, Austria
florian.eibensteiner@fh-hagenberg.at

Jürgen Kogler
AIT Austrian Institute of Technology GmbH
Donau-City-Strasse 1, 1220 Vienna, Austria
juergen.kogler.fl@ait.ac.at

Josef Scharinger
Johannes Kepler University Linz
Altenberger Strasse 69, 4040 Linz, Austria
josef.scharinger@jku.at

Abstract

As a novelty, in this paper we present an event-based stereo vision matching approach based on time-correlation using segmentation to restrict the matching process to active image areas, exploiting the event-driven behavior of a silicon retina sensor. Stereo matching is used in depth generating camera systems for solving the correspondence problem and reconstructing 3D data. Using conventionally frame-based cameras, this correspondence problem is a time consuming and computationally expensive task. To overcome this issue, embedded systems can be used to speed up the calculation of stereo matching results. The silicon retina delivers asynchronous events if the illumination changes instead of synchronous intensity or color images. It provides sparse input data and therefore the output of the stereo vision algorithm (depth map) is also sparse. The high temporal resolution of such event-driven sensors leads to high data rates. To handle these and the correspondence problem in real time, we implemented our stereo matching algorithm for a field programmable gate array (FPGA). The results show that our matching criterion, based on the time of occurrence of an event, leads to a small average distance error and the parallel hardware architecture and efficient memory utilization results in a frame rate of up to 1140fps.

1. Introduction

Embedded vision systems are used in applications of our daily life. Especially in automation, ranging from the industrial sector to consumer electronics, vision systems including sensors for 3D reconstruction are omnipresent. For example cars we drive are assembled nearly completely autonomously, driver assistance systems improve safety in traffic, transport systems (rail shuttles) drive completely autonomous without human interaction using the data of sensors retrieving depth data. In all these applications not only the depth data are important, but also the processing and availability of the depth data in real-time is crucial.

Sensors for calculating depth data comprise active sensors, such as laser range finders or laser scanners, time-of-flight (TOF) cameras, ultrasonic detectors, radar, light-section, and structured light as well as passive technologies, including structure from motion, optical flow, and stereo vision. Especially high resolution in space and time lead to huge amounts of data, which makes it difficult to do the depth calculation in real-time. In many computer vision applications the observed scene does not change all the time and the processing of redundant data is occupying the resources for processing the results. This means processing the whole image at each algorithm cycle would end up in lots of redundant work.

To overcome the processing of redundant data we use a so-called silicon retina sensor in a stereo set-up, where only changes in the observed scene are detected by the sensor. On the one hand, with this sensor only the relevant data has to be processed, but on the other hand the sensor has a high temporal resolution which means high data rates must be handled. This lead us to the usage of a Field Programmable Gate Array (FPGA) based embedded system for calculating results in real-time. Thus, in this work we put attention on the implementation of a stereo matching algorithm for sparse silicon retina data in an FPGA to analyze the real-time capability of such stereo vision system. However, such an algorithm can significantly benefit from application specific customizations of the underlying system ar-

chitecture by using optimized memory access patterns and special computation units.

The remainder of the paper is organized as follows. In Section 2 an overview about the silicon retina technology is given. Section 3 presents the related work of silicon retina-based stereo matching algorithms implemented in software as well as in hardware. The implementation of the stereo vision algorithm is explained in detail in Section **??**. Section 5 presents the evaluation of the implemented algorithm with real world data, and the final Section 6 gives a conclusion and outlook of future work.

2. Technology review - Silicon retina

As mentioned, in this work we use two *Silicon Retina* cameras in a stereo vision system to calculate depth data of a scene. In comparison to conventional complementary metal oxide semiconductor (CMOS) or charge coupled device (CCD) imagers, every pixel of a silicon retina sensor independently delivers data only on changes of the luminance. This frame-free, asynchronous, time-continuous, logarithmic photoreceptor offers three substantial advantages:

- The asynchronous illumination change dependent event data generation obviously leads to an significant data reduction because only dynamic parts of the scene are detected and static parts are completely suppressed.
- The construction of the pixel array and the event-based signal processing facilitates a very high temporal resolution of up to 10ns.
- The logarithmic measurement of the photo current yields to a high dynamic range, therefore the sensor is suitable for fast transient light conditions.

The silicon retina research goes back in the 1980s, where the first integration of a silicon retina on a single chip was done by Mead and Mahowald [19] in 1988. This model differs in its function from conventional camera sensors and imitates basic steps of the human visual system. In 1989 Mahowald and Mead introduced the term *Silicon Retina* and presented an implementation of a retina sensor based on silicon in [18] and [16]. Different photo-detector technologies and data encoding methodologies have been developed since this time, ranging from simple light to variable impulse rate transformation [7], time-to-first-spike encoding (TFS) [30], motion sensing and computation systems [2], silicon retinas sensing spatial contrast by doing more on-chip signal processing [6], and a model for a mammalian retina [32, 33]. These sensor technologies have two attributes in common: the read-out of the information is initiated by the pixel itself, and they use an address-event representation (AER) protocol [28, 16] for transmitting the event data from the sensor to the subsequent processing system. The work of Boahen [3] presented an implementation of the AER protocol for a point-to-point communication between neuromorphic chips.

The silicon retina sensor considered in this work (called ATIS, Asynchronous, Time-based Image Sensor) has a spatial resolution of 304×240 pixels, a temporal resolution of up to 10ns, and a dynamic range of 143dB. Details about the sensor and its characteristics can be found in the work of Posch *et al.* [21, 22]. For hardware consideration and evaluation a 128×128 sensor is considered as well, which is presented in the work of Lichtsteiner *et al.* [14, 15]. For generating events whenever the illumination of the observed scene changes, the silicon retina sensor uses an illumination change detector circuit. An event is defined as $e(p,t)$ [23], where $p = (x,y)^T$ is the spatial location of the pixel which fires the event, and t is the time of occurrence given in the units of timestamps. One timestamp corresponds to the temporal resolution of the sensor (1 timestamp $\triangleq$ 10ns, in the case of the ATIS). Due to the slow motion of the objects in our test cases, we use a temporal resolution of $100\mu s$ for one timestamp.

Depending on the polarity of the change of illumination I over a period of time Δt, an event can either be positive (on-event) or negative (off-event):

$$e(p,t) = \begin{cases} +1 & I(p,t) - I(p,t-\Delta t) > \Delta I \\ -1 & I(p,t) - I(p,t-\Delta t) < -\Delta I \end{cases}, \quad (1)$$

with the adjustable on- and off-threshold ΔI.

Especially the asynchronous behavior of the pixels is difficult to handle, because edges and contours build up over a period of time, where not all pixels of an edge are simultaneously active. We chose stereo vision as application for this work because matching sparse input data in real-time is a challenging task. For using the events with stereo matching algorithms the importance of the timestamps and the time period considered for the matching process need to be explained, because this parameters depend on the dynamics of the scene. Generally, static parts of the scene, e.g. non-moving objects, are not recognized by the sensor and therefore completely suppressed. In Figure 1(a) the intensity image of a non-moving person captured from a monochrome camera is shown. Figure 1(b) shows the same non-moving person captured by the silicon retina camera, where off-events are white and on-events are black. The pixels in gray represent pixels, where no event information was received from the silicon retina camera, e.g. stationary background. Because the person is not moving, the contour of the person is not visible and only very few events representing noise are received from the sensor. In contrast, Figure 1(c) shows the intensity image of the same person, but walking this time. The person observed with the silicon retina induces the event generation behavior of such sensor.

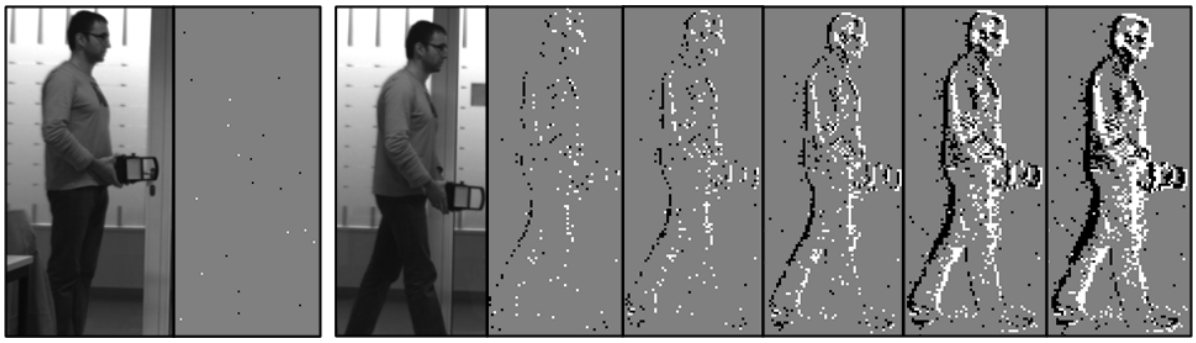

Figure 1. Silicon retina sensor in comparison to a conventional monochrome sensor. White pixels (off-events), Black pixels (on-events), Gray pixels (no events). (a) Person without movement in front of monochrome sensor, (b) silicon retina output without movement, (c) person walking in front of monochrome sensor, (d)-(h) silicon retina data from the walking person with collected events over a time period of 5ms, 10ms, 20ms, 40ms and 60ms.

The high temporal resolution of the sensor results in quite incomplete contours (Figure 1(d)) even with a collection of events over a period of time of 50 (5ms) timestamps. The object's shape gets more complete when more events of several timestamps are collected before they are converted into an image. Figure 1(e-h) show the events collected within a time period of 100 (10ms), 200 (20ms), 400 (40ms), and 600 (60ms) timestamps before visualizing them in an image. The time period should be chosen to get complete object contours as shown in Figure 1(f), but without blurred object edges as illustrated in Figure 1(h). This means the time period (time history) is not a fixed parameter, because it depends on the scene dynamics where the speed of the objects within the scene is very important.

3. Related Work

Different approaches and methods have been developed over years to solve the correspondence problem in stereo matching algorithms. In classical stereo vision the approaches can be subdivided in *area-based* and *feature-based* matching methods. Area-based methods correlate patches to find the best corresponding match based on similarity measures in contrast to feature-based techniques [26], where features are extracted and used for the matching process. Another way is using a local transform, where intensity relations between the actual pixel and the pixels in a certain window are considered before a correlation is applied. Such transforms are the rank transform and census transform, which are introduced in the work of Zabih and Woodfill [31]. A general summary and performance evaluation of area-based stereo matching algorithms is presented in the work of Brown *et al.* [4] and Scharstein and Szeliski [24].

Stereo matching for neuromorphic sensors started with the development of the silicon retina sensors, and in 1989 Mahowald and Delbrück [17] presented a stereo matching approach applied on event-based data using static and dynamic image features. Schraml *et al.* [25] evaluated in their work an area-based approach with different image distance functions. A comparison between area-based techniques and a feature-based center matching algorithm was evaluated 2009 in the work of Kogler *et al.* [13]. One year later Kogler *et al.* [11] presented an algorithm using weighted time differences as a correlation criterion for solving the correspondence problem. In 2011 Kogler *et al.* [12] proposed an event-image matching algorithm based on a local transform. The work of Benosman *et al.* [1], shows the correspondence search using coactivation sets and Rogsiter *et al.* [23] proposed in 2012 an event-based matching using the spatial distance to epipolar lines as a matching criteria. Another event-based stereo matching approach is presented in the work of Carneiro *et al.* [5] which uses multiple camera views to increase the accuracy of the matching output. Recently, in the work of Piatkowska *et al.* [20], a cooperative approach was presented, where a positive (excitatory) feedback from matches within the same disparity and a negative (inhibitory) feedback from matches of competing disparity planes are considered for calculating the final matching results.

All these approaches, either frame-based or event-based, use the sparse silicon retina data with its dynamics and high temporal resolution to solve the correspondence problem. These calculations are mainly done on desktop platforms, and therefore have no capabilities for processing the sparse silicon retina output in real-time. In 2008 Shimonomura *et al.* [27] proposed a neural network using a disparity energy model to emulate the stereo matching in the visual cortex (V1), which was implemented on a FPGA platform. An area-based SAD algorithm was implemented from Eibensteiner *et al.* [8] in hardware with the goal to evaluate the performance of the events to image conversion necessary for executing frame-based stereo matching algorithms. The frame rate of this systems depends mainly on

the framing duration used for gathering incoming events to an image. Using a frameing period of 1ms, a frame rate of 820fps could be achieved. The approach in the work of Kogler *et al.* [11], which is based on time correlation was implemented on a DSP as proposed by Sulzbachner *et al.* [29]. This approach was redesigned and adapted for a realization on an FPGA by Eibensteiner *et al.* [9]. They proposed an algorithmic concept, but no implementation was outlined.

In our work we present an approach implementing the stereo matching algorithm in hardware, such as FPGA platforms, facilitating a real-time computation of depth maps based on event-driven silicon retina sensors. To do this we introduce an effective time-based distance measure based on a logarithmic weighting function which can be easily computed and gives promising results. Furthermore, by using segmentation to isolate active image areas, the real-time capabilities of the implementation are increased significantly.

4. Event-based Stereo Matching Algorithm

Basis of the implementation of the stereo vision core in hardware is the algorithmic concept outlined in [9]. Due to the low density of information, a retina delivers only on- and off-events, so the polarity and the time of occurrence of an event are used as major matching criterion. Thus, as consequence of the asynchronous behavior, concerning the data delivery, for the correspondence search not only the current timestamp is considered, but also a predefined history. As Eibensteiner *et al.* proposed in [9], here also a logarithmic weighting function for calculating the matching probability is used.

As a novelty, the image plane is segmented into blocks of a predefined size $m \times n$, e.g. for this work we used segments with a size of 16×16 pixels, and the correspondence search is done only for image fields with a certain activity and not for each pixel of the whole image. The benefit of this approach is twofold: The stereo matching is done only for regions of interests, where events occurred respectively, combined with the advantages of significantly minimizing memory accesses and an efficient parallel memory architecture because for each segment a separate memory is used. Both lead to a faster matching process.

The event-based algorithm is briefly shown in Algorithm 1. First, new events from the left l and the right r sensor are written to an array of image memories $B_{his} \in \mathbb{R}^{c \times b \times m \times n}$ recognizing a history t_{his}, where c denotes the available channels (l or r), and b defines the amount of segments and m and n specify the dimensions of the segments.

$$B_{his}[g][k][i,j] = e(p,t) \quad (2)$$
$$\forall i \in \{0,\dots,m-1\} \wedge j \in \{0,\dots,n-1\}$$
$$\wedge k \in \{0,\dots,b-1\} \wedge g \in \{l,r\} \wedge t > t_{cur} - t_{his},$$

where g defines the channel (left or right), k is the segment number, i and j are the coordinates within a segment, t_{cur} is the current time, and t_{his} the recognized history. The position p is mapped onto entries in B_{his} according to the rule

$$p \mapsto k,i,j := \{i = x \bmod m, j = y \bmod n, \quad (3)$$
$$k = \lfloor \frac{x}{m} \rfloor + \lfloor \frac{y}{n} \rfloor \cdot \lfloor \frac{c_w}{m} \rfloor\},$$

where c_w denotes the width of the sensor array in pixels.

Furthermore, the functions used in Algorithm 1 work on segment level which leads to

$$B_{seghis}(g,k) = B_{his}(g,k,[0:m-1],[0:n-1]). \quad (4)$$

Old events are overwritten by new ones if they are at the same spatial location and events which are older than t_{his} are deleted. Subsequently, a segment queue $q \in \mathbb{R}^b$ is calculated by counting the events within the segments in the left image memories. Since the left sensor is used as reference for the depth map, a segment queue for the right channel must not be determined because the required segments for the consistency check results form the left segments. The amount of events determines not only the order for the matching process, therefore the queue is sorted descending, but also can be used for noise filtering by rejecting all segments from the queue which are below a certain threshold. After this, the stereo matching and the consistency check are done, and finally the resulting disparity map D_{cc} is returned.

4.1. Hardware Realization

The hardware implementation of this algorithm is depicted in the block diagram in Figure 2. In order to achieve

Algorithm 1 Event-based Stereo Matching Algorithm

Require: Two retinas R_l, R_r
Require: rectified event streams E_l and E_r
  for all events $e_l(p,t)$ in E_l **do**
    Build history by
    $B_{seghis}(k,l) = \text{merge}(B_{seghis}(k,l), e_l(p,t))$;
    Determine segment queue by
    $q = \text{count}(B_{seghis}([0:k-1],l))$;
    Prioritize queue by sorting $q_s = \text{sort\_desc}(q)$;
  end for
  Build history for right event stream
  $B_{seghis}(k,r) = \text{merge}(B_{seghis}(k,r), e_r(p,t))$;
  for all segments s_i in q_s **do**
    Compute disparities according to
    $D = \text{match}(B_{seghis}(s_i,l), B_{seghis}([0:k-1],r)$;
  end for
  Do consistency check $D_{cc} = \text{check}(D)$;
  return D_{cc}

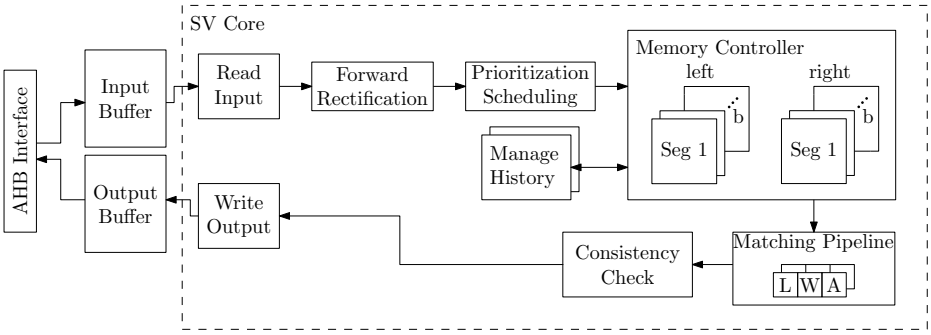

Figure 2. Block diagram of the hardware architecture of the event-based stereo vision core, coupled with dual port memories and a AHB interfaces to a SoC. The direction of the arrows shows the data flow.

a high processing speed and a high throughput, parallel processing, pipelining mechanisms, and efficient memory architectures must be used. Similar to the approach of Eibensteiner *et al.* [9], the algorithm is divided into: read and write interface, forward rectification unit, prioritization and scheduling unit where the segment queue is calculated, memory controller and a unit managing the history, matching pipeline and consistency check units. In the current implementation only internal on-chip memories of the FPGA are used, resulting in optimal memory access patterns and a very high memory bandwidth. By the standard AHB interface and the dual port memories, the core can be embedded into a SoC.

4.1.1 Segment Management

As a novelty, after the rectification step incoming events are segmented by calculating the segment number and the address within the segment from the pixel's coordinates which fires the event. Figure 3 shows the segmentation of the image plane using a block size of 16×16 pixels and a resolution of 128×128 pixels.

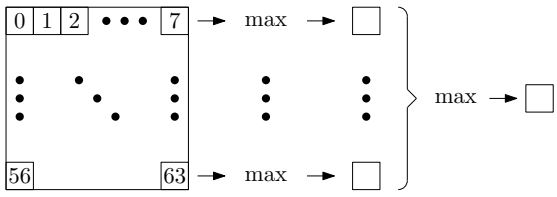

Figure 3. Segmentation of an image plane with a resolution of 128×128 pixels, using a block size of 16×16 pixels. Prioritization is done in two steps: search highest row counter, and from these the highest count value is computed.

In addition, for each segment a counter exists, which is incremented if a new event is stored into the corresponding memory block. After all events of the current timestamp have been rectified, too old events are deleted by the unit *manage history*. The unit processes always two segments at the same time, one from the left image and the corresponding segment from the right image. After updating all events, the counter values for the segments are decremented accordingly. As indicated in Figure 2, this unit can exist more than once, although each unit works on different areas. How many units are needed depends on the history length and amount of segments.

After that, the prioritization is carried out in two steps as shown in Figure 3. First, the highest counter of each row is determined, this is done for all rows in parallel, and in the second step from these again the highest count value is searched which schedules the first segment for the matching process.

As depicted in Figure 4, in order to do the consistence check, not only disparities for the reference segment in the left image must be calculated, but also the disparities for segments from the right image.

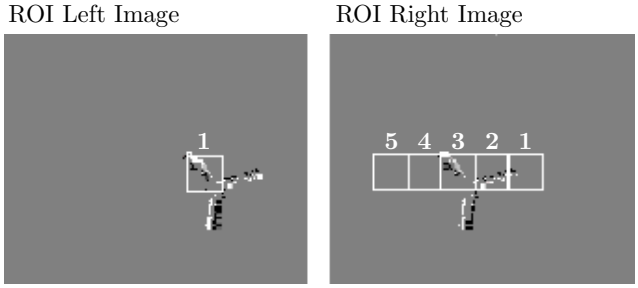

Figure 4. Segments which must be processed in order to calculate disparities for one segment in the left image. In the case of a disparity range of 50 and a block size of 16×16 pixels, the disparities of 6 segments (1 in the left and 5 in the right image) must be computed, hence the consistency check can be done.

The memory controller commands the access to the segments, using a three port memory architecture with one write and two read channels. In order to take advantage of the flexible embedded on-chip memories and to increase the bandwidth, per access 4 events are processed in parallel.

4.1.2 Matching Process and Consistency Check

The *matching pipeline* is designed as a 3-stage pipeline and works on segment level, this means in each stage another segment is processed. In the current implementation, two matching pipelines are used, processing up to 6 segments simultaneously. The pipeline composes of the stages load *L*, weight *W*, and aggregate *A*. The execution is as follows: first the data of a whole segment line from the left image and the corresponding disparity range of the right image is loaded. In the second step, the events are weighted together with

$$w(t_{eval}) = \left\lceil (10^{t_{his}-1-\Delta t})^{\frac{1}{t_{his}}} \right\rfloor \cdot \left\lceil (10^{t_{eval}-1})^{\frac{1}{t_{his}}} \right\rfloor, \quad (5)$$

where t_{eval} is the timestamp of the currently processed event, t_{his} is the length of the history considered in the matching process, and Δt is the difference of the timestamp of the evaluated event and the timestamp of the corresponding event from the data stream of the other image sensor. For performance reasons, the weights are not calculated at run-time but stored in a lookup table.

Since an area based matching technique is used, the weights are aggregated and the maximum weight in the disparity range, which actually defines the disparity, is determined. Thus, a stage works on one segment line, the height of the matching window defines how many iterations are needed before the first valid results are calculated.

As soon as results from all segments in the matching pipeline are available, the consistency check is done line by line. Thus, the unit increases the latency only by the calculation time of one line which is minimal compared with the overall time.

5. Experimental Results

The introduced implementation was evaluated with two real world indoor scenarios, where the sensor was static and observes a dynamic scene. In Figure 5 the achieved depth maps of both data sets, a rotating disc and moving persons, are shown.

As evaluation metric the average distance error in depth, calculated from a comparison with ground truth data is used. For the latter we use a different stereo sensor consisting of monochrome cameras which achieves an accuracy of 3% at a distance of $3.4m$. More details about the ground truth generation and the registration of the depth results from both stereo sensors onto each other are presented in the work of Kogler *et al.* [10]. Figure 6 shows the average distance error achieved by the event-based stereo vision algorithm (triangles) compared to an area-based SAD algorithm (squares) which is explained in [12]. This approach uses the events and generates grayscale images used for the matching with the SAD algorithm. In our experiments for both algorithms we use a correlation window size of 5×5, 9×9, and 15×15. The history length varies between 50, 100, and 150 for test set A, and between 200, 250, and 300 for test set B.

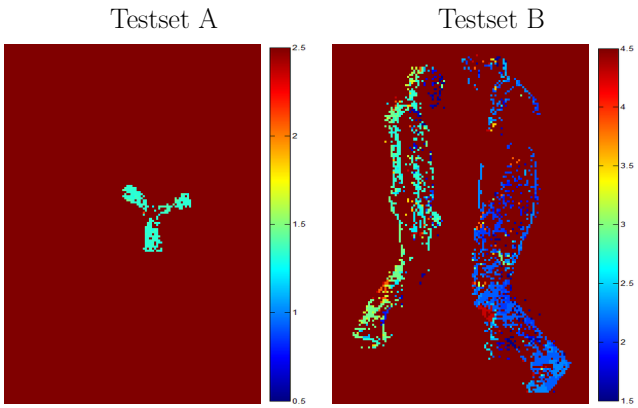

Figure 5. Depth maps of all data sets calculated by the event-driven stereo vision algorithm. Testset A a rotating disk in$1.5m$, and testset B two persons walking in $2.5m$ and $3.5m$.

Figure 6. Average distance errors of both test data sets achieved with the time-based approach and compared with the area-based SAD algorithm.

In general, the results show that larger window sizes and longer histories tend to lead to a smaller average distance error. In case of the rotating disc many events are spiking in a small neighborhood, and sharp contours blur to bigger edges resulting in more mismatches of the time-based algorithm because of repeating event sequences. The SAD algorithm overcomes this problem with the generation of grayscale images, where many events are represented by gray values, which are better for the matching process due to the higher information density. Considering the results of using the test set with the two persons, short histories do not generate enough gray value information for the correspondence search, but the time-based approach achieves good matching results because of only the on- and off-event data and the timestamps are correlated. With increasing his-

tory length the event data is better summarized during the grayscale conversion process and, thus, the SAD algorithm achieves a better average distance error.

Despite the low density of information of the event data, the hardware implementation almost achieve the quality of the SAD algorithm using grayscale images for the matching process. However, the generation of the grayscale image is very memory-intensive because all events occurred within the considered history must be stored. Due to the efficient memory utilization, only a fraction of the resources are required for the hardware implementation. Own representative evaluations have proved memory savings by a factor of about 7 for the management event history.

The impact of larger correlation windows is lower for the time-based approach as for the SAD algorithm. This can mainly be attributed to the fact that the time-based implementation only uses the event data for the matching process and SAD algorithm also considers the background information.

Table 1. Cycle count of the individual units of the stereo vision core for the calculation of one segment. In the case is an image plane of 128×128 pixels, a segment size of 16×16, a window size of 5×5, and disparity range of 36 assumed

Uint	Cycles
Manage History	66
Prioritization and scheduling	18
Matching Pipeline	1268
Consistency Check	18
$\sum$	1370

The performance of the hardware implementation in terms of execution time and data throughput mainly depends on the segment size, the size of the correlation window, and the disparity range. Table 1 shows the cycle count needed to calculate disparities for one segment. In this example we assume an image plane of 128×128 pixels, segment size of 16×16, a window size of 5×5, and disparity range of 36. Using a clock frequency of 100MHz, this leads to a calculation time of 13.7μs. Even in the worst-case scenario where all 64 segments must be calculated a frame rate of 1140fps can be achieved, referring the matching process. In real world scenarios the frame rate will be even higher, because changes in all segments at the same time are highly improbable and so less segments must be processed per calculation step, or timestamp, respectively. In comparison with the implementation outlined in [8] the event-driven approach works 28% faster.

The latency caused by the rectification unit depends on the event rate. In test case A the average event rate was about 4 events per timestamp per channel and in the case of the moving persons about 10 events per timestamp per channel. So in those cases, the latency will be increased only by 22 cycles or 28 cycles, respectively, which hardly influences the frame rate of the stereo vision core.

The distance measure used for the time-based approach can more easily be turned into a parallel hardware architecture and in addition the segmentation further increases the real-time capabilities which leads to a short processing time. The direct processing of the event data enables a resource-saving realization which obtains almost a small distance error as the SAD algorithm.

6. Conclusion

In this work we presented an event-driven stereo vision algorithm for a silicon retina stereo camera system implemented in hardware. The advantages of the novel approach using segmentation and pipelined parallel event processing are twofold: first the amount of memory accesses is significantly reduced because only changed segments are processed, and that, second, the efficient hardware architecture leads to a very high frame rate of up to of 1140fps at 100MHz. In addition, the results show that despite the low information density and the simple matching criterion almost the same accuracy can be achieved as the SAD algorithm but at a significantly higher processing speed. Furthermore, the advantages of a FPGA are exploited by using parallel architectures and optimized memory access patterns and due to the standard on-chip bus interface the extensibility and reusability of the stereo vision core is absolutely enhanced.

Future work may include the evaluation of the restriction of the search space because by knowing which segments are active the required amount of segments for the matching process can be reduced. Another possibility to increase the frame rate is a further parallelization of the matching process by using more pipelines.

References

[1] R. Benosman, S.-H. Ieng, P. Rogister, and C. Posch. Asynchronous Event-Based Hebbian Epipolar Geometry. *IEEE Transactions on Neural Networks*, 22(11):1723–1734, 2011.

[2] K. Boahen. Retinomorphic chips that see quadruple images. In *Proceedings of the 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems. MicroNeuro '99.*, pages 12–20, Granada/Spain, 1999.

[3] K. Boahen. Point-to-point connectivity between neuromorphic chips using address events. *IEEE Journal of Transactions on Circuits and Systems II*, 47(5):416–433, 2000.

[4] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in Computational Stereo. *IEEE Journal of Transactions on Pattern Analysis and Machine Intelligence*, 25:993–1008, 2003.

[5] J. a. Carneiro, S.-H. Ieng, C. Posch, and R. Benosman. Event-based 3D reconstruction from neuromorphic retinas. *Journal of Neural networks*, 45:27–38, september 2013.

[6] J. Costas-Santos, T. Serrano-Gotarredona, R. Serrano-Gotarredona, and B. Linares-Barranco. A Spatial Contrast

Retina With On-Chip Calibration for Neuromorphic Spike-Based AER Vision Systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(7):1444–1458, 2007.

[7] E. Culurciello, R. Etienne-Cummings, and K. A. Boahen. A biomorphic digital image sensor. *IEEE Journal of Solid-State Circuits*, 38(2):281–294, feb 2003.

[8] F. Eibensteiner, A. Gschwandtner, and M. Hofstätter. A high-performance system-on-a-chip architecture for silicon-retina-based stereo vision systems. In *Proc. of the 2010 IRAST International Congress on Computer Application and computational Science*, pages 976 – 979, Singapur, 2010.

[9] F. Eibensteiner, J. Kogler, C. Sulzbachner, and J. Scharinger. Stereo-Vision Algorithm Based on Bio-Inspired Silicon Retinas for Implementation in Hardware. In *Proceedings of the 13th International Conference on Computer Aided Systems Theory EUROCAST*, Lecture Notes in Computer Science, pages 624–631, Las Palmas/Spain, 2011.

[10] J. Kogler, F. Eibensteiner, M. Humenberger, M. Gelautz, and J. Scharinger. Ground Truth Evaluation for Event-Based Silicon Retina Stereo Data. In *Proceedings of the 9th IEEE Embedded Vision Workshop EVW (held in conjunction with IEEE CVPR)*, pages 649–656, Portland/USA, 2013.

[11] J. Kogler, C. Sulzbachner, F. Eibensteiner, and M. Humenberger. Address-event matching for a silicon retina based stereo vision system. In *Proceedings of the 4th International from Scientific Computing to Computational Engineering IC-SCCE*, pages 17–24, Athens/Greece, 2010.

[12] J. Kogler, C. Sulzbachner, and M. Humenberger. Event-Based Stereo Matching Approaches for Frameless Address Event Stereo Data. In *Proceedings of the 7th International Symposium on Visual Computing ISVC*, pages 674–685, Las Vegas/USA, 2011. Springer-Verlag Berlin Heidelberg.

[13] J. Kogler, C. Sulzbachner, and W. Kubinger. Bio-inspired stereo vision system with silicon retina imagers. In *Proceedings of the 7th International Conference on Computer Vision Systems ICVS*, pages 174–183, Liege/Belgium, 2009. Springer-Verlag Berlin Heidelberg.

[14] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128x128 120 dB 30 mW asynchronous vision sensor that responds to relative intensity change. In *Proceedings of the IEEE International Solid-State Circuits Conference ISSCC*, SanFrancisco/USA, 2006.

[15] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128x128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43(2), 2008.

[16] M. Mahowald. *VLSI analogs of neuronal visual processing: a synthesis of form and function*. Phd-thesis, California Institute of Technology, 1992.

[17] M. Mahowald and T. Delbrück. Cooperative Stereo Matching Using Static and Dynamic Image Features. In C. Mead and M. Ismail, editors, *Analog VLSI Implementation of Neural Systems*, volume 80 of *The Kluwer International Series in Engineering and Computer Science*, pages 213–238. Springer US, 1989.

[18] M. Mahowald and C. Mead. Silicon retina. *Journal of Analog VLSI and Neural Systems*, pages 257–278, 1989.

[19] C. A. Mead and M. Mahowald. A silicon model of early visual processing. *Journal of Neural Networks*, 1(1):91–97, 1988.

[20] E. Piatkowska, A. N. Belbachir, and M. Gelautz. Asynchronous stereo vision for event-driven dynamic stereo sensor using an adaptive cooperative approach. In *Proceedings of the 3rd Workshop on Consumer Depth Cameras for Computer Vision CDC4CV (held in conjunction with IEEE ICCV)*, pages 45–50, Sydney/Australia, 2013.

[21] C. Posch, D. Matolin, and R. Wohlgenannt. An asynchronous time-based image sensor. In *Proceedings of the IEEE International Symposium on Circuits and Systems ISCAS*, pages 2139–2133, Seattle/USA, 2008.

[22] C. Posch, D. Matolin, and R. Wohlgenannt. A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2011.

[23] P. Rogister, R. Benosman, S.-H. Ieng, P. Lichtsteiner, and T. Delbruck. Asynchronous event-based binocular stereo matching. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2):347–353, february 2012.

[24] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *Journal of Computer Vision*, 47(1-3):7–42, 2002.

[25] S. Schraml, P. Schön, and N. Milosevic. Smartcam for real-time stereo vision - address-event based embedded system. In *Proceedings of the 2nd International Conference on Computer Vision Theory and Applications VISAPP*, volume 2, pages 466–471, Barcelona/Spain, 2007.

[26] J. Shi and C. Tomasi. Good Features to Track. In *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference CVPR*, pages 593–600, Seattle/USA, 1994.

[27] K. Shimonomura, T. Kushima, and T. Yagi. Binocular robot vision emulating disparity computation in the primary visual cortex. *Journal of Neural Networks*, 21(2-3):331–340, 2008.

[28] M. Sivilotti. *Wiring consideration in analog vlsi systems with application to field programmable networks*. Phd-thesis, California Institute of Technology, 1991.

[29] C. Sulzbachner, C. Zinner, and J. Kogler. An optimized silicon retina stereo matching algorithm using time-space correlation. In *Proceedings of the 7th IEEE Embedded Computer Vision Workshop ECVW (held in conjunction with IEEE CVPR)*, pages 1–7, Colorado Springs/USA, 2011.

[30] G. Xiaochuan, Q. Xin, and J. G. Harris. A time-to-first-spike CMOS image sensor. *IEEE Sensors Journal*, 7(8):1165–1175, august 2007.

[31] R. Zabih and J. Woodfill. Non-parametic Local Transforms for Computing Visual Correspondence. In *Proceedings of the 3rd European Conference on Computer Vision ECCV*, pages 151–158, Stockholm/Sweden, 1994.

[32] K. A. Zaghloul and K. Boahen. Optic Nerve Signals in a Neuromorphic Chip I: Outer and Inner Retina Models. *IEEE Transactions on Biomedical Engineering*, 51(4):657–666, april 2004.

[33] K. A. Zaghloul and K. Boahen. Optic Nerve Signals in a Neuromorphic Chip II: Testing and Results. *IEEE Transactions on Biomedical Engineering*, 51(4):667–675, april 2004.