

Article

Stereo Imaging Using Hardwired Self-Organizing Object Segmentation

Ching-Han Chen ¹, Guan-Wei Lan ^{1,*} , Ching-Yi Chen ² and Yen-Hsiang Huang ³

¹ Department of Computer Science and Information Engineering, National Central University, Taoyuan 32001, Taiwan; pierre.miat@gmail.com

² Department of Information and Telecommunications Engineering, Ming Chuan University, Taoyuan 333321, Taiwan; chingyi@mail.mcu.edu.tw

³ National Chung-Shan Institute of Science and Technology, Taoyuan 32546, Taiwan; wantai943@hotmail.com

* Correspondence: michaellan@g.ncu.edu.tw

Received: 15 August 2020; Accepted: 13 October 2020; Published: 15 October 2020



Abstract: Stereo vision utilizes two cameras to acquire two respective images, and then determines the depth map by calculating the disparity between two images. In general, object segmentation and stereo matching are some of the important technologies that are often used in establishing stereo vision systems. In this study, we implement a highly efficient self-organizing map (SOM) neural network hardware accelerator as unsupervised color segmentation for real-time stereo imaging. The stereo imaging system is established by pipelined, hierarchical architecture, which includes an SOM neural network module, a connected component labeling module, and a sum-of-absolute-difference-based stereo matching module. The experiment is conducted on a hardware resources-constrained embedded system. The performance of stereo imaging system is able to achieve 13.8 frames per second of 640 × 80 resolution color images.

Keywords: object segmentation; SOM; stereo vision

1. Introduction

Stereo vision is an imaging technique developed on the basis of biological vision principles. It involves using two cameras to simultaneously capture two images, comparing the images to find the objects that match, and estimating the distance between the objects and the cameras using the disparity of the objects within the two images. Therefore, image segmentation and stereo matching are often two important technologies for developing stereo vision systems. First, the image segmentation technique is used to separate the objects in the two images and identify the corresponding points of the same objects within both images for further matching. Then, the principle of stereo matching is used to acquire the depth of the objects.

The purpose of image segmentation is to partition an image into several regions or objects that do not overlap and that have similar characteristics. Image segmentation methods can be divided into the following types: the thresholding method, edge-based method, region-based method, watershed method, clustering-based method, and neural-network-based method. The thresholding method [1,2] generates threshold values by analyzing the gray-level histogram of full or partial images and segments the objects within the images by clustering all of the pixels of these images. Of the numerous commonly used segmentation techniques, this method has been widely well-received because of its simple implementation procedures, robustness, and accuracy [3]. Even in recent years, academia has continued to develop novel thresholding approaches or improve existing ones [4–8]. The edge-based method [9] measures or extracts region boundaries by detecting the edges of images to identify objects in these images; this method has the advantage of low complexity, but it can also be easily affected by

noise. The region-based method [10,11] utilizes the homogeneity of pixel features (e.g., gray scale, color, or texture) in the same region as the criterion of segmentation, the purpose of which is to partition an image into regions with distinct characteristics. The watershed method-based algorithm [12,13] segments images by considering an image to be a topographic map and utilizing variations in flood water heights and watershed lines.

Segmentation methods that do not use spatial information to group pixels into regions are often called clustering techniques. Clustering is a useful unsupervised data mining technique that partitions the input space into K regions depending on a similarity/dissimilarity metric. The clustering-based methods use measures of intraclass and interclass homogeneity to determine the optimal segmentation. The simplest clustering approach commonly employs well-known techniques such as K-means [14], fuzzy c-means [15], and probabilistic extension using the Gaussian mixture model and expectation–maximization algorithm [16]. This type of image segmentation is widely used due to the simplicity of understanding and accurate segmentation results.

Artificial neural networks (ANNs) are well known for their excellent performance in classification and function approximation and have been used with success in digital image processing. Neural network-based segmentation methods rely on processing small regions of an image using an ANN or a set of different ANNs. After such processing, the decision-making method marks the regions of an image according to the categories recognized by the neural network. A self-organizing map (SOM) is a type of ANN designed specially to address such problems [17]. The use of a SOM in image segmentation is well reported in the literature [18,19].

Because the development of stereo vision systems involves complex algorithms, powerful CPU resources and high-performance computational platforms are typically required. Thus, the performance of embedded systems used for stereo vision processing is often constrained by cost and technical limitations, which render it difficult to achieve real-time processing. Performing image segmentation solely using software is simply a computationally intensive task [20] and is therefore impractical for developing real-time embedded systems. However, several studies involving the hardware implementation of SOM have been conducted [20–22]; for example, in [21], Porrmann, Ruping, and Ruckert applied a scalable parallel architecture to understand SOM digital hardware, which features a classification rate of 250,000 vectors per second. In [22], the researchers designed a modular SOM systolic architecture that can classify data vectors with thousands of elements in real time. The architecture is described as a soft intellectual property core in synthesizable VHSC hardware description language. Observing that a SOM architecture can be easily converted into parallel processing units, Kurdtongmee [20] proposed a novel Kohonen SOM-based architecture that involves using field programmable gate array (FPGA) chips. This architecture was developed based on unsigned integer arithmetic in all operations and performs adequately in terms of image quality, frame rate throughput, and FPGA chip resource utilization.

Building on this previous scholarship, the present study proposed a SOM-based object segmentation method that can be applied to stereo vision systems and employed FPGA techniques to develop a hardware accelerator that performs real-time and rapid object segmentation. We also designed a pipeline controller to control the operation of three modules (i.e., a SOM neural network module, connected component labeling [CCL] module, and stereo matching module) to effectively improve overall system performance and achieve the requirements of real-time processing and high efficiency.

The remainder of this paper consists of five sections: Section II introduces relevant stereo vision techniques, Section III elaborates on the methods for designing the hardware of an embedded stereo vision system, Section IV presents the experiment results and compares system performance, and Section V provides a summary of the study findings.

2. Materials and Methods

2.1. Stereo Vision Algorithms

Figure 1 presents the stereo vision system architecture proposed in this study. The dual camera vision module captures left and right images and sends the left images to the image preprocessing module for classification and CCL. Stereo matching is performed on the obtained objects and right images to estimate the similarity and disparity of the same object between two images. A lookup table (LUT) is then utilized to generate depth values.

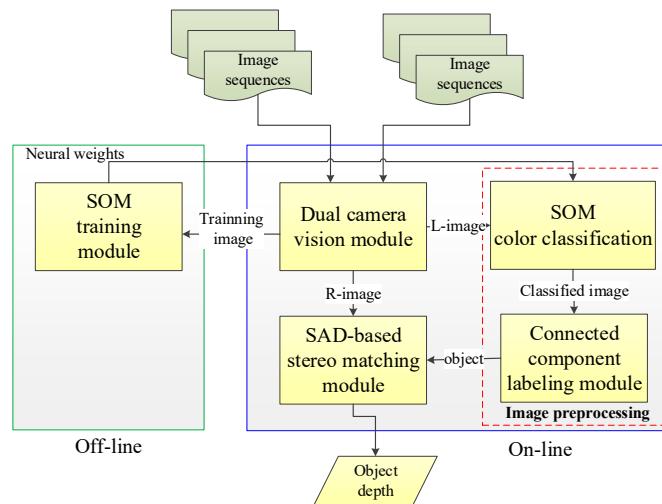


Figure 1. System structure of the proposed embedded stereo system.

2.1.1. Self-Organized Map

A SOM [23] is a neural network based on competitive learning; that is, the neurons of the output layer compete with each other to reach an activated opportunity. In most competitive learning neural networks, the winner is selected through a competitive phase and the weight vector of the winner is adjusted in a rewarding phase. However, in a SOM, both the winning neuron and its neighboring neurons have a chance to learn after the competitive phase. The conventional SOM learning algorithm can be explained using the following steps:

- (1) Initialize the weight vectors of the $M \times N$ neurons
- (2) Repeat until convergence

(a) Select the next input vector x_i from the data set:

(i) Find the unit W_j^* that best matches the input vector x_i

$$J^* = \underset{j}{\operatorname{argmin}} \|x_i - W_j\|, j = 1, \dots, M * N \quad (1)$$

(ii) Update the weights of the winner W_j^* and its neighboring neurons W_k

$$JW_k = W_k + \eta(t) \cdot h_{j^*k}(t) \cdot (x_i - W_k) \quad (2)$$

where $h_{j^*k}(t)$ is the Gaussian neighborhood function given as

$$h_{j^*k}(t) = \exp\left(-\frac{d_{j^*k}^2}{2\sigma(t)^2}\right) \quad (3)$$

(b) Select the next input vector x_i from the data set:

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau_1}\right) \quad (4)$$

(c) Decrease neighborhood size $\sigma(t)$ that defines the topological neighborhoods:

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau_2}\right) \quad (5)$$

2.1.2. Connected Component Labeling

CCL is a task that detects connected regions in input data, and which can be applied for pattern recognition and image segmentation. A CCL module scans an image and groups its pixels into components based on pixel connectivity. All pixels in a connected component share similar pixel intensity values and are in some way connected with each other. Once all groups have been determined, each pixel is labeled with a gray level or a color according to the component to which it was assigned.

2.1.3. Stereo Matching

The sum of absolute difference (SAD) technique [24] is the most common matching criterion in stereo matching algorithms because of its low complexity, excellent performance, and ease of hardware implementation. The technique computes the intensity differences for each center pixel (i,j) in a window $W(x,y)$ as follows:

$$\text{SAD} = \sum_{(i,j) \in W} |I_L(i, j) - I_R(i, j + d)|, \quad (6)$$

where I_L and I_R denote the left and right image pixel intensity functions, respectively, and $W(x,y)$ is the square window that surrounds the location (x,y) of a pixel.

Figure 2 illustrates the process of stereo matching. First, a target block is selected from a left image as a template, which is then gradually moved along a baseline in the right image to compare both images and locate the block that best resembles the target block from the left image. Stereo matching is then conducted to calculate disparity values and determine the level of similarity between both blocks. Finally, an LUT or formula is used to obtain depth information about the objects in the images.

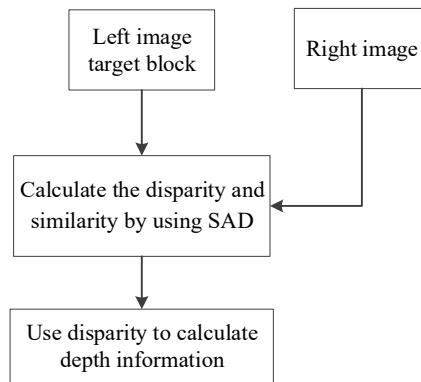


Figure 2. Stereo matching flowchart.

2.2. Hardware Architecture of the Embedded Stereo Vision System

2.2.1. Dual Camera Vision Module

Figure 3 shows the hardware architecture of the dual camera vision module. A signal filter removes excess noise, and a serial camera control bus module performs the initial setting of the

complementary metal-oxide-semiconductor sensors on both sides. After receiving the input image data, the dual camera vision module then sends the data to the color interpolation module to generate RGB color-based image data, which is finally stored in a synchronous dynamic random-access memory (SDRAM) cell. The RTL schematic view of the dual camera vision module is presented in Figure 4.

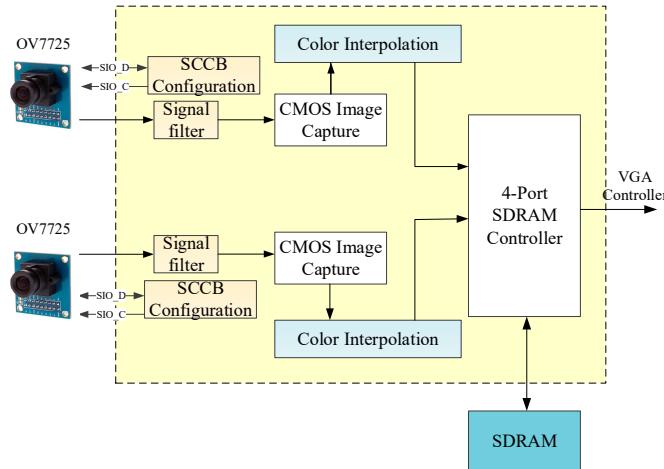


Figure 3. Dual camera vision module.

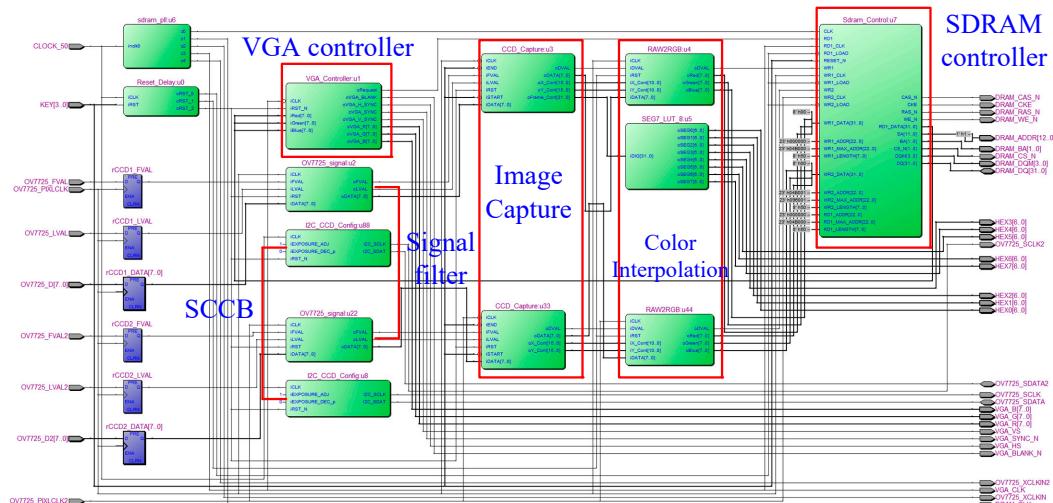


Figure 4. RTL-schematic diagram of the dual camera vision module.

2.2.2. Self-Organized-Map-Based Image Segmentation Module

The SOM-based image segmentation module contains two submodules, namely an SOM training module and an SOM color classification module (Figure 5). Notably, only the data of the first image is input into the SOM training module to train the color segmentation module; the data of the remaining images are input directly into the SOM color classification module to execute various pixel classification tasks.

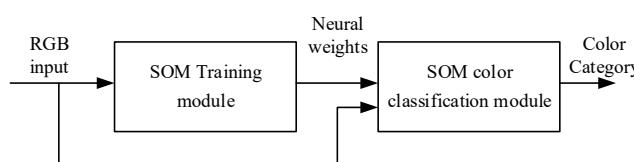


Figure 5. SOM-based image segmentation module.

SOM Training Module

Figure 6 displays the system architecture of the SOM training module, wherein a random generator generates initial weights for the SOM. The distance calculation module and decision module are then used to calculate input vectors and the distance of weights for all output neurons, thereby determining the winner neuron. The weights-update module adjusts the bonding values of the winner neuron and its neighboring neurons. Subsequently, a divider provides the computational power for the weights-update module. Figure 7 presents the Grafcet discrete-event model [25] of the SOM training module.

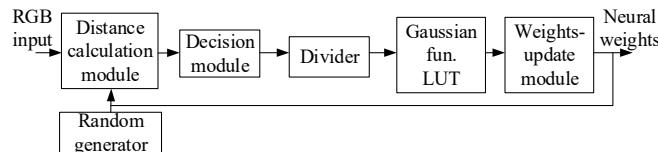


Figure 6. System architecture of the SOM training module.

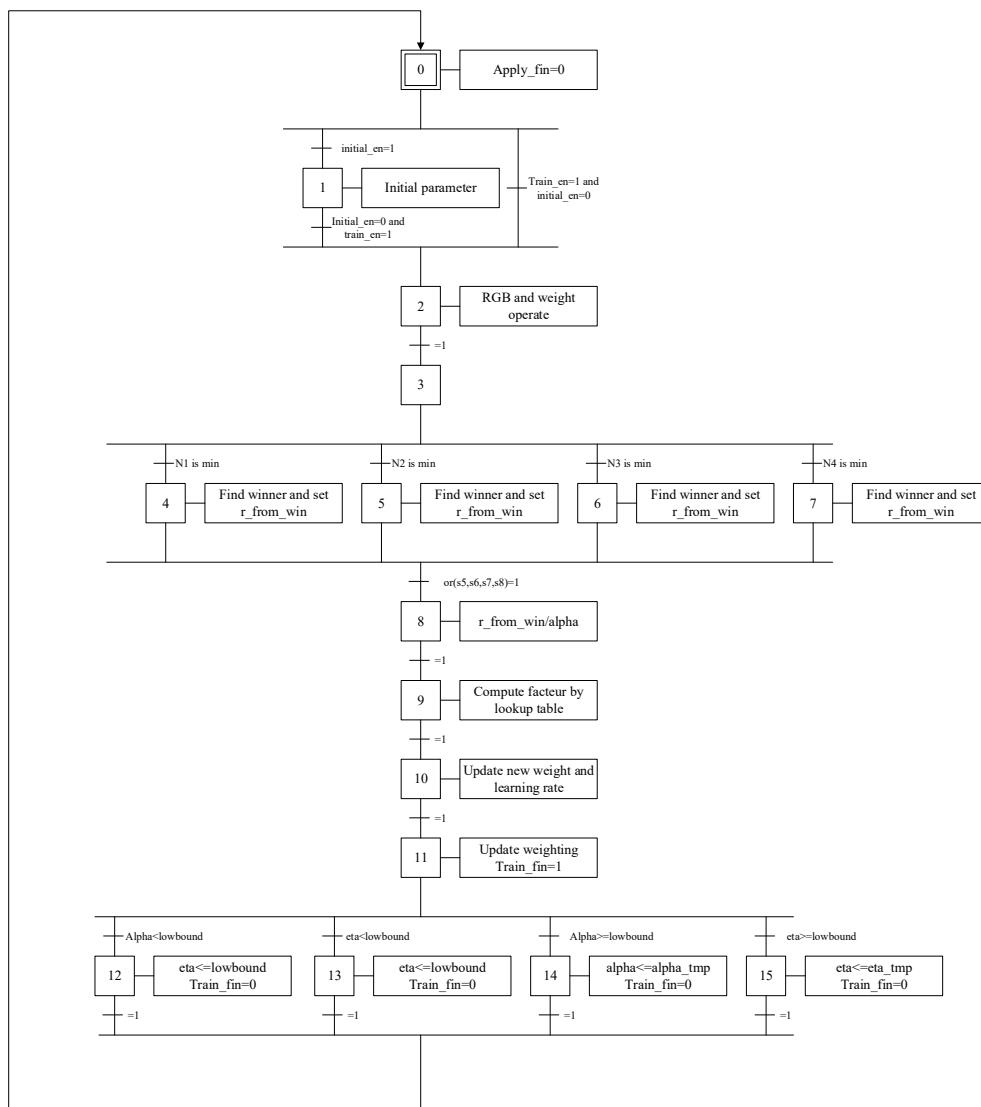


Figure 7. Grafcet discrete-event model of the SOM training module.

Random Generator

During the training phase of a SOM, the initial weights of all neurons are random. Thus, a circuit module that generates random numbers must be designed. Random number generators using linear feedback shift registers (LFSRs) were developed in ref. [26]. Because the state of a register is often limited, the process of random number generation is a repeated cycle; however, the use of primitive polynomials enables the LFSRs to generate random sequences with extended cycle periods. LFSRs have been used to generate pseudo-random binary sequences, which are then used to derive the required random sequences. Figure 8 provides an illustration of the random generator for the SOM training module adopted in the present study, which was designed using LFSRs.

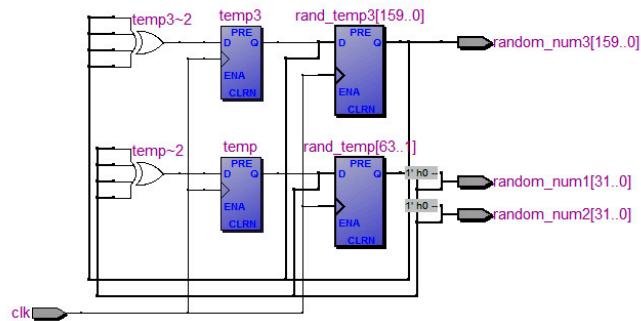


Figure 8. RTL-schematic diagram of the random generator.

Lookup Table of Gaussian Function

In the SOM algorithm, a winner neuron stimulates its neighboring neurons through lateral connections. The level of stimulation is related to the distance between these lateral connections. Specifically, neurons at a shorter distance from the winner neuron are stimulated to a higher level, whereas neurons at a longer distance from the winner neuron are stimulated to a lower level. The Gaussian function employed in this study is expressed as (7):

$$h_{j*k} = \exp\left(-\frac{d_{j*k}^2}{\alpha}\right) \quad (7)$$

SOM Color Classification Module

The SOM color classification module can be divided into two submodules: a distance calculation module and a decision module. These modules are mainly used for calculating input vectors and the distance of the weights for all output neurons, selecting the winner neuron, and determining the category of the input vector. Figures 9 and 10 display the system architecture and the Grafcet discrete-event model of the SOM color classification module, respectively.

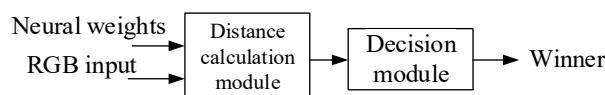


Figure 9. System architecture of the SOM color classification module.

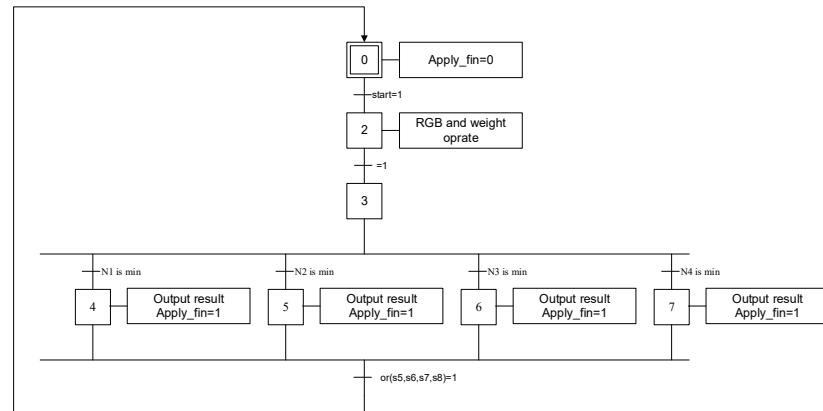


Figure 10. Discrete-event model of the SOM color classification module.

Figure 11 shows the RTL-schematic diagram of the SOM-based image segmentation module, with the SOM training module on the left and the SOM color classification module on the right. According to the analysis results of the maximum clock frequency, the employed SOM-based image segmentation module processes 322 color images with a resolution of 640×480 per second. However, it can only process up to 50 color images with a resolution of 640×480 per second when coupled with the dual camera vision module, whose system clock constrains the processing speed.

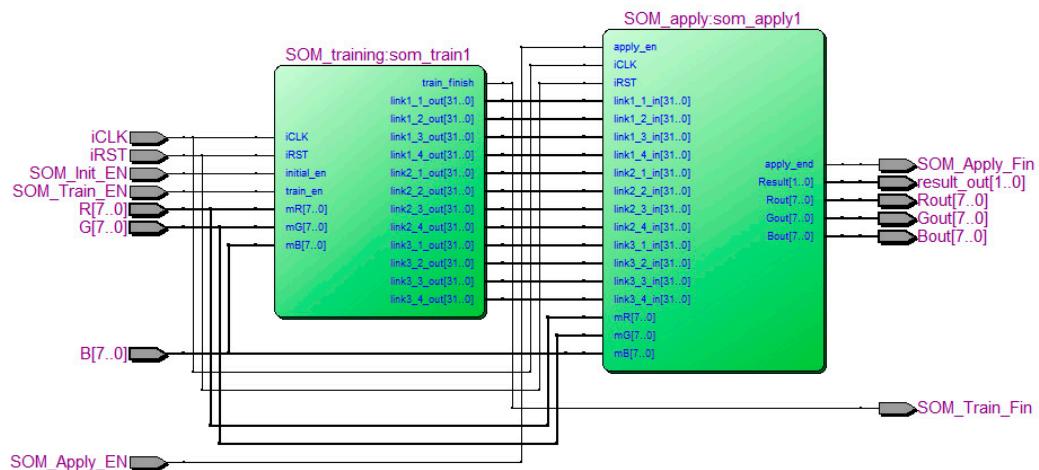


Figure 11. RTL-schematic diagram of the SOM-based image segmentation module.

2.2.3. Connected Component Labeling Module

Binary images are scanned in the CCL module through a raster scan, which is performed along a path from left to right and from top to bottom. Notably, the scanning of a pixel requires only identifying the position of its neighboring pixels. Figure 12 illustrates the hardware architecture of the CCL module, which shows that the information of neighboring pixels q , r , s , and t in the 8-connected component is stored using a line buffer. The labeling information obtained from the label assigner module and merge controller module is stored with the pixel information in a merge table and data table.

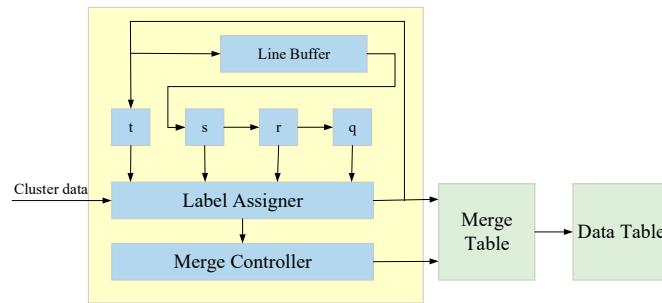


Figure 12. Hardware architecture of the CCL module.

Sum-of-Absolute-Difference-Based Stereo Matching Module

The SAD-based stereo matching module constitutes two submodules: a SAD matching module and a depth calculation module. Figure 13 illustrates the system architecture of the SAD-base stereo matching module. The purpose of the SAD matching module is to perform the SAD matching computation for the left-image target blocks stored in the static random-access memory (SRAM) and the right images stored in the SDRAM. Then, the depth calculation module estimates depth information by applying a LUT to the obtained disparity values.

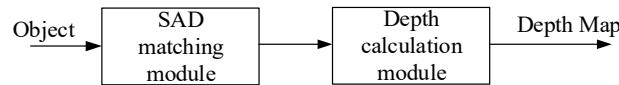


Figure 13. System architecture of the SAD-based stereo matching module.

Pipeline Controller and System Integration

Pipeline techniques, which are widely used for designing controllers, improve controller performance by utilizing the parallel processing ability of hardware. In this study, a pipeline controller was developed to control the parallel processing procedures, the data flow, and communication among all modules. In Figure 14a, X1–X5 represent the primary operating modules of the proposed system, namely the dual camera vision module, SOM image segmentation module, CCL module, SAD-based stereo matching module, and video graphics array (VGA) controller, and T1–T5 and B1–B5 represent idle stages, which are used to stabilize the transfer of control signals. The pipeline controller views each circuit module as a task and conditionally controls them. On top of the pipeline controller, a top controller was designed to control state sequences (Figure 14b).

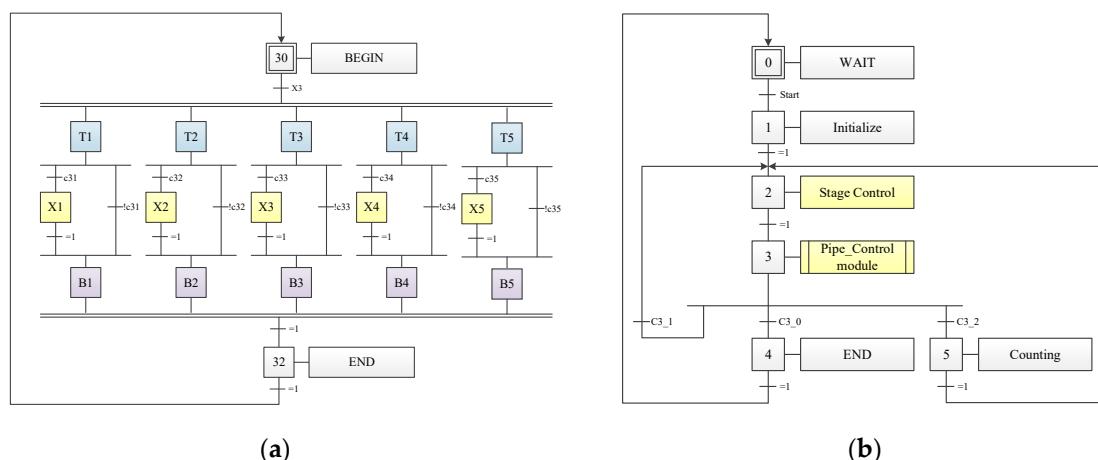


Figure 14. Discrete-event model of the pipeline controller. (a) Stage controller. (b) Top controller.

3. Results

3.1. Software Simulation and Verification

To verify the performance of the stereo vision algorithm, software simulation was conducted on a personal computer that had an Intel Core 2 E8400 3.00 GHz, 3-GB DDRIII RAM, and the 32-bit Windows 7 operating system. The software was developed using Borland C++ Builder 6. To compare the effects of image segmentation between various methods, an overlap measure was used as the criterion of evaluation. The formula of the overlap measure is given as [27]:

$$\text{overlap} = \frac{TP}{TP + FP + FN} \quad (8)$$

where TP , FP , and FN denote the true positive (i.e., the area correctly classified as the object), false positive (i.e., the area incorrectly classified as the object), and false negative (i.e., the area incorrectly classified as the background) areas, respectively.

Figure 15 shows the results of segmenting an image of the Sydney Opera House using different methods. Specifically, Figure 15a is the original image; Figure 15b is the ideal result of image segmentation; Figure 15c,d are images segmented using K-means and an SOM, respectively. A comparison of these images indicates that the SOM-based method generated more accurate results than did the K-means method. Table 1 presents a list of the results of applying these two methods to segmenting three different images, with the overlap measure and correct rate of both methods calculated.

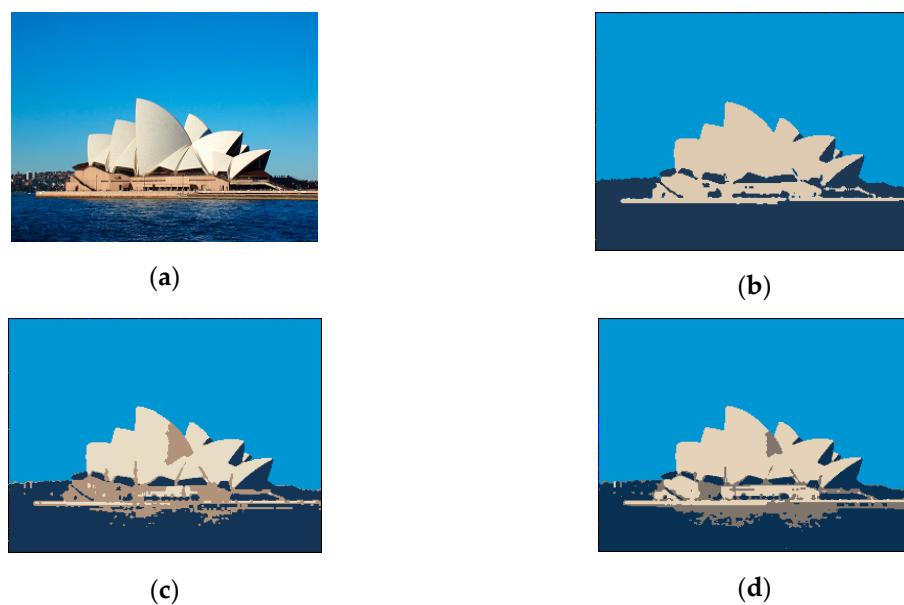


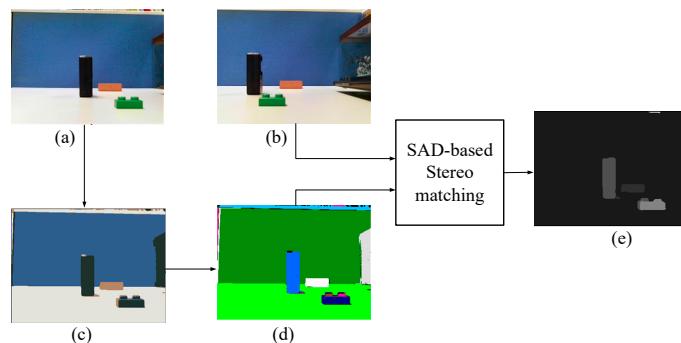
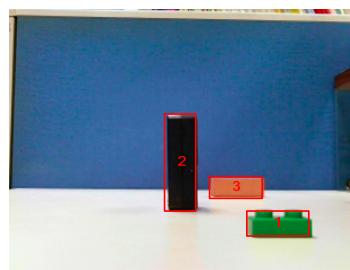
Figure 15. Segmentation results of an image of the Sydney Opera House. (a) Original image. (b) Ideal image segmentation. (c) Segmented image using K-means. (d) Segmented image using SOM.

Next, a depth map was generated by applying CCL processing and stereo matching to the segmented objects (Figure 16). Figure 17 depicts the objects that were segmented, and the disparity values and estimated depth information derived from these objects (which suggest that the employed algorithm was able to accurately estimate depth information) are presented in Table 2.

Table 1. Comparison of k-means clustering and the SOM neural network.

Method	Image	Opera House	Sky	Ocean	Total	Correct Rate
K-Means	TP	7058	38,400	15,994	61,452	0.9104
	FN	0	26	0	26	
	FP	3416	131	1455	5002	
	overlap	0.673	0.995	0.916	-	
SOM	TP	9096	38,420	14,280	61796	0.9154
	FN	12	16	0	28	
	FP	1378	111	3169	4658	
	overlap	0.867	0.996	0.818	-	

Image size: 300 × 255

**Figure 16.** Results of the proposed stereo vision method. (a) Left image. (b) Right image. (c) Image after color classification. (d) Image after CCL. (e) Depth map.**Figure 17.** Labeled objects.**Table 2.** Estimated depth information.

Objects	Center-Point of Original Object	Center-Point of Target Block	Disparity	Estimated Depth	Observed Depth
1	(247,197)	(116,197)	131	18.8 cm	19.0 cm
2	(158,142)	(76,142)	82	36.2 cm	35.0 cm
3	(207,162)	(153,162)	54	56.0 cm	56.0 cm

3.2. Performance of the Embedded Stereo Vision System

The proposed embedded stereo vision system was implemented on an Altera DE2-115 FPGA board, which provides Altera Cyclone IV chips that enable the user to operate and verify their design. The system architecture and a photo of this architecture are presented in Figure 18a,b, respectively. During the operation process, a 128-MB SDRAM (with 32 MB of 32-bit memory) was used to store images and a 2-MB SRAM (with 1 MB of 16-bit memory) was used to store the information about the segmented object.

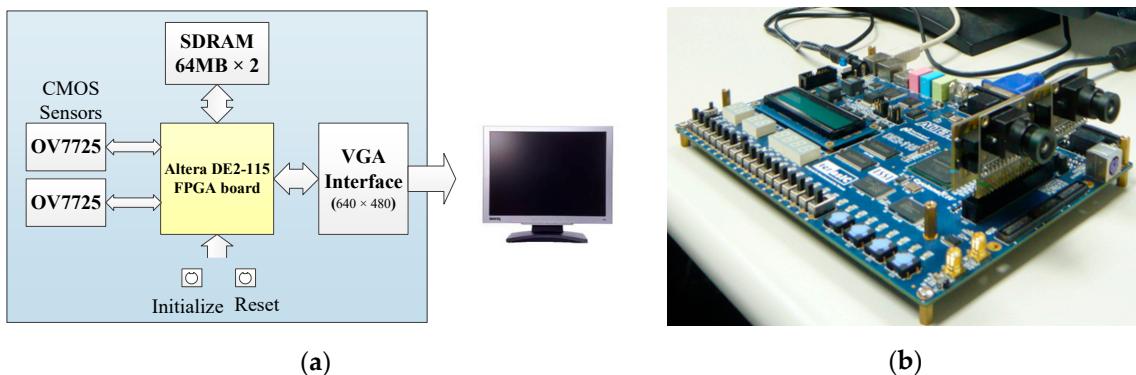


Figure 18. Hardware configuration of the embedded stereo vision system. **(a)** System hardware architecture. **(b)** Photo of the system.

3.2.1. System Performance Analysis

A sample image was used to verify the performance of the proposed stereo vision system (Figure 19). Figure 19a,b are the left and right images captured by the dual camera vision module, respectively. Figure 19c,d present the results of object segmentation and CCL processing using the left image, respectively, and Figure 19e shows the results of the image after stereo matching and depth estimation were conducted. The image in Figure 19e confirms that the proposed system can accurately generate depth maps by utilizing left and right images.

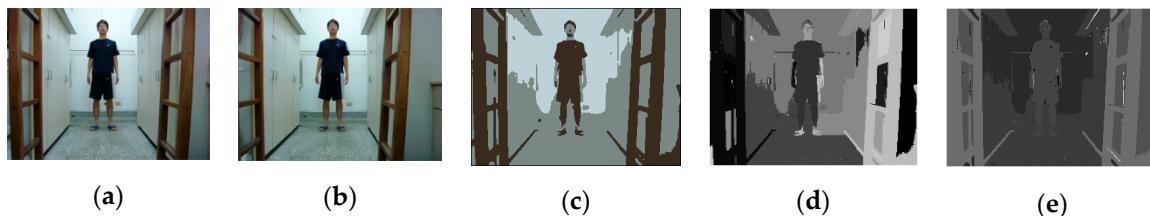


Figure 19. **(a)** Left image. **(b)** Right image. **(c)** Image after left-image object segmentation. **(d)** Image after left-image CCL processing. **(e)** Image after stereo matching.

3.2.2. Hardware Resource Utilization and Performance Analysis

The resource allocation of the proposed stereo vision system is presented in Table 3. At a system frequency of 48 MHz, the dual camera vision module captured 50 color images with a resolution of 640×480 per second. The SOM training module has a system frequency of 6.63 MHz. By contrast, the SOM color classification module has a maximum system frequency of 105.88 MHz, and thus could immediately classify color images. A comparison of the performance of the existing SOM hardware accelerators when applied to process images with a resolution of 640×480 in terms of their learning and color classification rates was conducted. The results, presented in Table 4, indicate that the proposed system architecture produced higher learning and classification rates than did the other three previous methods, and therefore can perform real-time processing in an embedded system.

Table 3. Resource allocation of the proposed stereo vision system on an altera cyclon IV chip.

Module Resources \	Dual Camera Vision System	SDRAM	SOM-Based Image Segmentation	VGA Controller
System clock	48 MHz	122.55 MHz	6.63 MHz–105.88 MHz	265.82 MHz
Total Logic Element	2117/114,480 (2%)	1672/114,480 (1%)	22,498/114,480 (20%)	79/114,480 (<1%)
Total Register	1429	757	2709	56
Total Memory Bits	425,952/3,981,312 (11%)	49,152/3,981,312 (1%)	0/3,981,312 (0%)	0/3,981,312

Table 4. Performance comparison with exiting SOM hardware architectures.

		Manolakos's Method [22]	Kurdthongmee's Method [20]	Porrmann's Method [21]	Our Method
SOM Training Module	learning rate (vectors/s)	68,900	N/C	94,000	413,125
	system clock classification	148 MHz	24.2 MHz	40 MHz	6.63 MHz
SOM Color Classification Module	rate (vectors/s)	144,000	N/C	250,000	42,265,000
	system clock	148 MHz	24.2 MHz	40 MHz	105.88 MHz

To further verify the improvement in the performance of all operating modules after they were implemented with hardware, an Altera DE2-115 control panel was used to load color images with a resolution of 640×480 into the SDRAM. The SDRAM controller was then employed to read the obtained image data and implement various image processing experiments. Next, the data derived from these experiments were compared with those obtained from software simulation. The comparison results, shown in Table 5, reveal that the proposed operation modules, after being implemented with hardware, had excellent performance, and exhibited higher computational efficiency than that of the simulation software program. In addition, the parallel operations and pipeline architecture effectively reduced the idle time of the circuit modules. The proposed stereo vision system can process 13.8 images with a resolution of 640×480 per second, which satisfies the requirement of real-time processing.

Table 5. Comparison of time needed to process a 640×480 color image using a software program or hardware module.

	Software Program	Pipelined Hardware Module
Clock Frequency	3 GHz	84.53 MHz
Object Segmentation	0.039 s	0.004 s
CCL	0.037 s	0.066 s
Stereo Matching	0.122 s	0.064 s

In this section, we show that the SOM method used in this research has good performance in several other methods in terms of learning rate and classification rate, and we also compare it with the software implementation of a computer with a system frequency of 3 GHz. Still can have outstanding performance. Besides, we are even better than the methods of Jin's [28], Michalik's [29], and Wang's [30] in the use of FPGA resources, and also have quite excellent performance at a system clock of only 48 MHz.

4. Conclusions

Deep learning is the most popular method in contemporary computer vision research, but it requires a lot of system resources and power consumption. The verification platform used in this study is low power consumption, low cost, and low system clock. Even so, the method of this research has quite good performance in several aspects.

This study designed an SOM neural network module architecture and developed it into a hardware accelerator that effectively enhances the computational power of conventional modules in image segmentation. In short, neural network training and color classification can be efficiently performed using the proposed system. During the operation process, a pipeline architecture coupled with parallel processing functions was designed by incorporating the aforementioned SOM hardware accelerator, a dual camera vision module, a CCL module, and a stereo matching module into one stereo vision system that was assembled on an Altera DE2-115 board. Empirical verification determined that the proposed stereo vision system can process 13.8 color images with a resolution of 640×480 per second, and therefore has commercial value as a potential industrial application.

Author Contributions: C.-H.C. was involved in methodology elaboration. G.-W.L. was responsible for drafting the manuscript, and critically revised and approved the manuscript to be submitted. C.-Y.C. and Y.-H.H. performed system design and experiments. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sahoo, P.K.; Soltani, S.; Wong, A.K.; Chan, Y.C. A survey of thresholding techniques. *Comput. Vis. Graph. Image Process.* **1988**, *4*, 233–260. [[CrossRef](#)]
2. Wesolkowski, S. Color image edge detection and segmentation: A comparison of the vector angle and the Euclidean distance color similarity measures. Master’s thesis, Systems Design Engineering, University of Waterloo, Waterloo, ON, Canada, 1999.
3. Li, Z.; Yang, J.; Liu, G.; Cheng, Y.; Liu, C. Unsupervised range-constrained thresholding. *Pattern Recognit. Lett.* **2011**, *32*, 392–402. [[CrossRef](#)]
4. Hu, M.; Li, M.; Wang, R.G. Application of an improved Otsu algorithm in image segmentation. *J. Electron. Meas Instrum.* **2010**, *24*, 443–449. [[CrossRef](#)]
5. Long, J.W.; Shen, X.J.; Chen, H.P. Adaptive minimum error threshold algorithm. *Acta Autom. Sin.* **2012**, *38*, 1134–1144. [[CrossRef](#)]
6. Liu, J.; Jin, W.D. Three-dimensional adaptive minimum error thresholding segmentation algorithm. *J. Image Graph.* **2013**, *18*, 1416–1424.
7. Wu, Y.Q.; Zhang, B.C.; Wu, J.M. An infrared image segmentation method based on within-class absolute difference and chaotic particle swarm optimization. *Acta Opt. Sin.* **2010**, *30*, 79–85.
8. Qiao, L.Y.; Xu, L.X.; Gao, M. Fast maximum entropy thresholding based on two-dimensional histogram oblique segmentation in infrared imaging guidance. *Infrared laser Eng.* **2013**, *42*, 1691–1699.
9. Al-Amri, S.S.; Kalyankar, N.V.; Khamitkar, S.D. Image segmentation by using edge detection. *Int. J. Comput. Sci. Eng.* **2010**, *2*, 804–807.
10. Angelina, S.; Suresh, L.P.; Krishna Veni, S.H. Image segmentation based on genetic algorithm for region growth and region merging. In Proceedings of the International Conference on Computing, Electronics and Electrical Technologies (ICCEET), Kumaracoil, India, 21–22 March 2012; pp. 970–974.
11. Khokher, M.R.; Ghafoor, A.; Siddiqui, A.M. Image segmentation using multilevel graph cuts and graph development using fuzzy rule-based system. *IET Image Process.* **2013**, *7*, 201–211. [[CrossRef](#)]
12. Chai, Y.H.; Gao, L.Q.; Lu, S.; Tian, L. Wavelet-based watershed for image segmentation algorithm. Proceeding of The Sixth World Congress on Intelligent Control and Automation, Dalian, China, 21–23 June 2006; pp. 9595–9599.
13. Kang, W.X.; Yang, Q.Q.; Liang, R.R. The comparative research on image segmentation algorithms. In Proceedings of the 2009 First International Workshop on Education Technology and Computer Science, Wuhan, China, 7–8 March 2009; pp. 703–707.
14. Ilea, D.E.; Whelan, P.F. Color image segmentation using a spatial k-means clustering algorithm. In Proceedings of the Irish Machine Vision & Image Processing Conference 2006 (IMVIP 2006), Dublin, Ireland, 30 August–1 September 2006; pp. 146–153.
15. Shen, Y.; Li, Y. Robust image segmentation algorithm using fuzzy clustering based on kernel-induced distance measure. In Proceedings of the International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008; pp. 1065–1068.
16. Fauvel, M.; Tarabalka, Y.; Benediktsson, J.A.; Chanussot, J.; Tilton, J.C. Advances in spectral-spatial classification of hyperspectral images. *Proc. IEEE* **2013**, *101*, 652–675. [[CrossRef](#)]
17. Torbati, N.; Ayatollahi, A.; Kermani, A. An efficient neural network based method for medical image segmentation. *Comput. Biol. Med.* **2014**, *44*, 76–84. [[CrossRef](#)] [[PubMed](#)]
18. Yeo, N.C.; Lee, K.H.; Venkatesh, Y.V.; Ong, S.H. Colour image segmentation using the self-organizing map and adaptive resonance theory. *Image Vis. Comput.* **2005**, *23*, 1060–1079. [[CrossRef](#)]
19. Ortiz, A.; Gorri, J.M.; Ramirez, J.; Salas-Gonzalez, D. Improving MR brain image segmentation using self-organising maps and entropy-gradient clustering. *Inf. Sci.* **2014**, *262*, 117–136. [[CrossRef](#)]

20. Kurdthongmee, W. A novel Kohonen SOM-based image compression architecture suitable for moderate density FPGAs. *Image Vis. Comput.* **2007**, *26*, 1094–1105. [CrossRef]
21. Porrmann, M.; Ruping, S.; Ruckert, U. SOM hardware with acceleration module for graphical representation of the learning process. In Proceedings of the Seventh International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems, Granada, Spain, 9 April 1999; pp. 380–386.
22. Manolakos, I.; Logaras, E. High Throughput Systolic SOM IP Core for Fpgas. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Honolulu, HI, USA, 15–20 April 2007; *2*, pp. II-61–II-64.
23. Kohonen, T. *Self-Organization and Associative Memory*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 1988.
24. Zhang, X.; Chen, Z. SAD-Based Stereo Vision Machine on a System-on-Programmable-Chip (SoPC). *Sensors* **2013**, *13*, 3014–3027. [CrossRef] [PubMed]
25. David, R. Grafcet: A powerful tool for specification of logic controllers. *IEEE Trans. Control. Syst. Technol.* **1995**, *3*, 253–268. [CrossRef]
26. Alfke, P. Efficient Shift Registers, LFSR Counters, and Long Pseudo-Random Sequence Generators. Available online: https://www.xilinx.com/support/documentation/application_notes/xapp052.pdf (accessed on 14 October 2020).
27. Van Ginneken, B.; Frangi, A.F.; Staal, J.J.; ter Haar Romeny, B.M.; Viergever, M.A. Active shape model segmentation with optimal features. *IEEE Trans. Med Imaging* **2002**, *21*, 924–933. [CrossRef] [PubMed]
28. Jin, M.; Maruyama, T. Fast and accurate stereo vision system on FPGA. *ACM Trans. Reconfig. Technol. Syst.* **2014**, *7*, 1–24. [CrossRef]
29. Wang, W.; Yan, J.; Xu, N.; Wang, Y.; Hsu, F.H. Real-time high-quality stereo vision system in FPGA. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1696–1708. [CrossRef]
30. Michalik, S.; Michalik, S.; Naghmouchi, J.; Berekovic, M. Real-time smart stereo camera based on FPGA-SoC. In Proceedings of the 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), Birmingham, UK, 15–17 November 2017; pp. 311–317.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).