PAPER

Grid Sample Based Temporal Iteration for Fully Pipelined 1-ms SLIC Superpixel Segmentation System

Yuan LI[†a)], Tingting HU[††], Ryuji FUCHIKAMI[††], *Nonmembers*, *and* Takeshi IKENAGA[†], *Member*

SUMMARY A 1 millisecond (1-ms) vision system, which processes videos at 1000 frames per second (FPS) within 1 ms/frame delay, plays an increasingly important role in fields such as robotics and factory automation. Superpixel as one of the most extensively employed image oversegmentation methods is a crucial pre-processing step for reducing computations in various computer vision applications. Among the different superpixel methods, simple linear iterative clustering (SLIC) has gained widespread adoption due to its simplicity, effectiveness, and computational efficiency. However, the iterative assignment and update steps in SLIC make it challenging to achieve high processing speed. To address this limitation and develop a SLIC superpixel segmentation system with a 1 ms delay, this paper proposes grid sample based temporal iteration. By leveraging the high frame rate of the input video, the proposed method distributes the iterations into the temporal domain, ensuring that the system's delay keeps within one frame. Additionally, grid sample information is added as initialization information to the obtained superpixel centers for enhancing the stability of superpixels. Furthermore, a selective label propagation based pipeline architecture is proposed for parallel computation of all the possibilities of label propagation. This eliminates data dependency between adjacent pixels and enables a fully pipelined system. The evaluation results demonstrate that the proposed superpixel segmentation system achieves boundary recall and under-segmentation error comparable to the original SLIC algorithm. When considering label consistency, the proposed system surpasses the performance of state-of-the-art superpixel segmentation methods. Moreover, in terms of hardware performance, the proposed system processes 1000 FPS images with 0.985 ms/frame delay.

key words: *image processing system, real-time, SLIC, superpixel, FPGA*

1. Introduction

With the rapid advancement of factorial digitalization and Internet technologies, machine vision has emerged as a critical technology in various domains, including robotics and factory automation (FA). Machine vision systems typically employ cameras to capture visual data, utilize machine vision algorithms to process this data, and provide feedback to actuators based on the processed information. However, it is crucial to minimize delays in the algorithmic processing stage, as real-world scenes continuously change during this time. If there is a significant delay, the feedback provided to the actuator becomes outdated and ineffective. On the contrary, if the algorithm with ultra-low delay is implemented, interactions between the algorithm and actuator become more frequent. The actuator responds in a timely manner and works seamlessly. In order to achieve more efficient and accurate machine vision systems, ultra-low delay implementation is necessary. Specifically, systems capable of processing 1000 frames per second (FPS) with processing speeds under 1 ms are desired. This consideration arises from the fact that the actuators can operate at frequencies of 1 kHz [1]. Many previous studies have already explored the use of Field-Programmable Gate Arrays (FPGAs) to develop high frame rate and ultra-low delay systems in various fields. Examples include object tracking [2], straight-line detection [3], and subpixel displacement measurement [4].

Superpixels involve the grouping of connected pixels exhibiting similar features. This concept significantly aids in reducing image redundancy and computational complexity for various subsequent tasks. Examples of such tasks include salient object detection [5] and 3D matching [6]. Currently, there are two main approaches to superpixel segmentation: hand-crafted feature-based algorithms and learned feature-based algorithms. Hand-crafted algorithms utilize features such as color and texture to group spatially adjacent pixels. Simple linear iterative clustering (SLIC) [7] clusters pixels based on their color similarity and proximity in the image plane. SLIC requires multiple iterations to obtain the final segmentation results. Linear spectral clustering (LSC) [8] maps image pixels to weighted points in ten-dimensional feature space by kernel functions. For learned feature-based algorithms, SCN [9] employs the U-net architecture to predict the association for each pixel. AINet [10] proposes an AI module to implant the corresponding grid features to the surroundings of the pixel. This module further perceives the relations between pixels and their neighbors. Learned feature-based algorithms are computation-intensive and memory-expensive, posing challenges in achieving ultra-low delay processing. Among the hand-crafted algorithms, SLIC is the most commonly used because of its high stability and fast segmentation speed [11]. Therefore, this paper adopts SLIC as the basic architecture for superpixel segmentation.

Several previous works have made efforts to accelerate SLIC. gSLIC [12] implements SLIC in GPU and achieves a six-fold speed improvement when compared to CPU implementations. FMSLIC [13] introduces a pixel-based search operation in the assignment step to remove data dependency. FMSLIC achieves a processing speed of 143 frames per second for input images. Despite these advancements in speed and memory efficiency, the iterative assignment and

Manuscript received June 29, 2023.
Manuscript revised September 30, 2023.
Manuscript publicized December 19, 2023.

[†] The authors are with the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu-shi, 808–0135 Japan.

[††] The authors are with Panasonic Connect Co., Ltd., Fukuoka-shi, 812–8531 Japan.

a) E-mail: liyuan3104@fuji.waseda.jp

DOI: 10.1587/transinf.2023EDP7128

update steps make these approaches challenging to process images at 1000 FPS with a processing speed under 1 ms. FP-SLIC [14] directly reduces the number of iterations to a very small number, but it compromises the algorithm's robustness. To address this challenge, and as an extension of our previous work [15], this paper proposes a hardware-oriented SLIC algorithm with its system-level hardware implementation. The contributions of this paper are summarized as follows:

1. Grid sample based temporal iteration is proposed. Instead of performing iterations within each image processing, the iterations are distributed into temporal domain. Each frame processes a single iteration. Grid sample information is incorporated as initialization data to enhance the stability of superpixels.
2. Selective label propagation based pipeline architecture is proposed to eliminate data dependency between adjacent pixels and achieve a fully pipelined system. All the possibilities of label propagation are calculated simultaneously and final results are selected based on the label propagation.
3. The proposed architecture has been implemented on an FPGA to develop a high frame rate and ultra-low delay system. Extensive experiments have been conducted to thoroughly validate both the algorithm and hardware performance of the system.

The rest of this paper is organized as follows. Section 2 reviews the related works. The proposed methods and implementation details are presented in Sect. 3. Experimental results and analysis are presented in Sect. 4. Finally, Sect. 5 makes a conclusion.

2. Related Works

2.1 Hand-Crafted Feature-Based Algorithm

ERS [16] formulates superpixel segmentation as an optimization problem. It aims to maximize the entropy rate of a random walk on the image graph. ARWN [17] utilizes the gradient of all pixels to assist in seed selection. This strategy allows each pixel to have the possibility of being chosen as a seed. However, the ultra-low delay implementation of these algorithms on FPGAs faces challenges due to their reliance on global processing. SLIC [7] calculates the Euclidean distance in color and proximity between each pixel and the superpixel center. It employs a local search range and involves simple calculations. LSC [8] is based on SLIC structure. LSC extends the search range to the entire image and maps pixels to weighted points in a ten-dimensional feature space to improve performance. ETPS [18] proposes a coarse-to-fine segmentation approach. This approach starts from a square grid and gradually reduces it to a one-pixel grid. SEEDS [19] also utilizes a grid-based initialization and iteratively reassigns boundary pixels using color and boundary histograms. However, The coarse-to-fine architecture of ETPS and SEEDS requires significant hardware resources to handle the different grid sizes. In the pursuit of accelerating SLIC, SNIC [20] introduces a priority queue to avoid iterative steps. However, constructing a priority queue requires saving the entire image. Additionally, the access between memory and processing core leads to a long delay during implementation.

2.2 Acceleration of SLIC

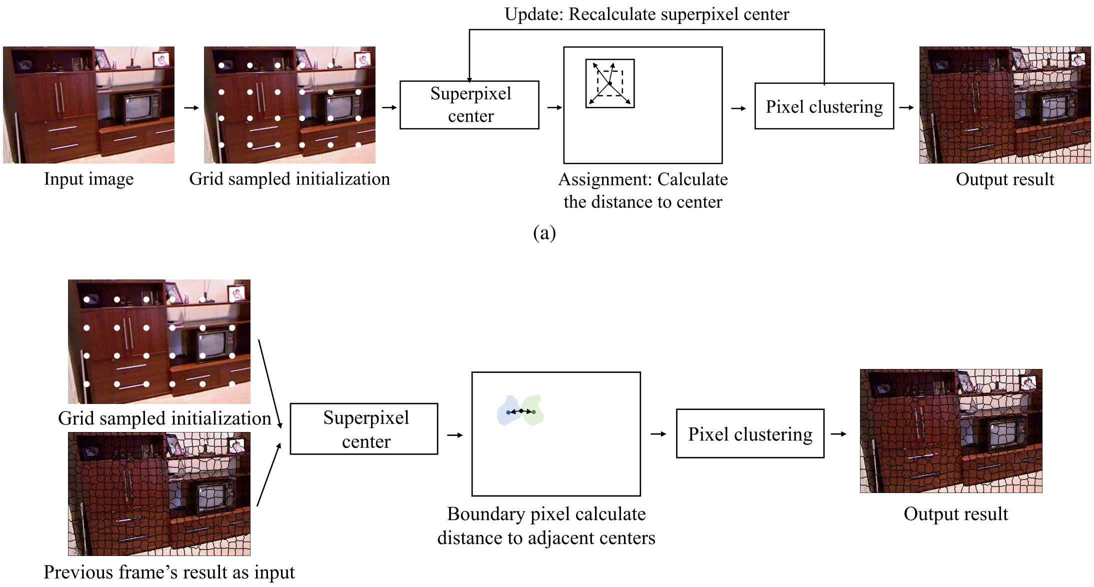
Due to the relatively low computational requirements and local calculations in SLIC, several works have attempted to accelerate SLIC and implement it on FPGA platforms. Khamaneh *et al.* [21] proposes a memory-efficient SLIC architecture. Only the label of each pixel and RGB color information are stored in memory. An RGB–to–CIELAB convertor is employed in the assignment step. The proposed SLIC algorithm achieves a frame rate of 24 fps for a camera with a resolution of 300x400 pixels. Mighani *et al.* [13] introduces a structure that replaces the cluster-based search operation with the pixel-based process in the assignment step. By eliminating the data dependency between the assignment and update steps, simultaneous execution is enabled. The proposed architecture processes 143 fps for an input image of 300x400 pixel resolution. Although the previous works have improved memory efficiency and execution time, the processing speed still falls short of achieving ultra-low delay. The iterations within each image processing result in frequent reading and writing of labels to memory, and lead to a long processing delay.

To address the issue of iterations, FP-SLIC [14] adopts a strategy of reducing the number of iterations. By limiting the iterations to just 2 times and implementing a fully pipelined FPGA architecture, the system's processing delay for an image with a resolution of 481x321 reaches 3.86 ms. However, this delay still does not meet the requirement of ultra-low delay applications, and the reduction of iteration times comes at the cost of robustness. Additionally, as the number of iterations increases, there is a substantial rise in both hardware resource utilization and processing delay. Further research is needed to develop a high frame rate and ultra-low delay system for SLIC superpixel segmentation.

3. Proposed Method

3.1 Grid Sample Based Temporal Iteration

The comparisons between the structure of SLIC and the proposed system are shown in Fig. 1. In SLIC method, initial superpixel centers are obtained by sampling image pixels at a regular grid. The grid size is determined based on the image size and the desired number of superpixels. In the assignment step, each pixel searches in a limited region. The distances to the superpixel centers within this region are calculated. The distance metric combines both CIELAB color and proximity Euclidean distance. Pixels are assigned to the superpixel with the smallest distance. Once every pixel is assigned to superpixels, superpixel centers are recalculated


Update: Recalculate superpixel center
Input image
Grid sampled initialization
Superpixel center
Assignment: Calculate the distance to center
Pixel clustering
Output result
Grid sampled initialization
Previous frame's result as input
Superpixel center
Boundary pixel calculate distance to adjacent centers
Pixel clustering
Output result

(a)

(b)

Fig. 1 Comparisons between the structure of SLIC and proposed method. (a) Original SLIC structure; (b) Proposed structure.

in the update step. Assignment and update steps operate iteratively to obtain the final output result. To reach the high frame rate and ultra-low delay, iteration is a critical problem.

The proposed grid sample based temporal iteration is illustrated in Fig. 1 (b). By exploiting the high similarity between adjacent frames in high frame rate video, the iterations are distributed across the temporal domain. The proposed structure utilizes the output of the previous frame as input for the process of the current frame. The output of the previous frame serves as the penultimate iteration result, while the output of the current frame is obtained through only a single iteration processing.

The utilization of temporal iteration enables the achievement of ultra-low delay; however, it introduces instability to the superpixels due to the absence of initialization information. This instability becomes especially apparent when the camera moves. It causes the superpixels at the edge of the image to either lose or gain pixels. Moreover, camera movement causes certain superpixels to vanish, while others grow significantly larger than the expected size. To address this issue, grid sample initialization is added to temporal iteration process. The operation is defined by the following equations.

$$cor_{cur} = \lambda \cdot cor_{ini} + (1 - \lambda) \cdot cor_{tem}, \tag{1}$$

$$\lambda = \min(A, 2^{A-\varepsilon}), \tag{2}$$

$$A = \left\lfloor \left| 10 \times \left(1 - \frac{size}{size_{exp}}\right) \right| \right\rfloor. \tag{3}$$

In the equations, cor_{cur}, cor_{ini}, cor_{tem} denote the coordinate of the current frame's input superpixel center, grid sampled initialization, and the center obtained from the previous frame's output, respectively. The multiplication by 10 in the calculation of A is performed to ensure an integer operation. The variable $size$ represents the number of pixels contained in the superpixel. $size_{exp}$ represents the expected size. ε is the parameter to control the ratio between the amount of initialization information and the previous output information. When the size of the current superpixel is close to the expected size, less initialization information is added. However, if the size deviates significantly from the expected size, the inclusion of grid sample initialization becomes necessary to preserve the stability of the superpixel.

The proposed system takes high frame rate videos as input. With a frame rate of 1000 FPS, the difference between each adjacent frame is minimal. In the original SLIC algorithm, each pixel searches for the superpixel centers, which leads to low label consistency in subsequent processing. The proposed system is inspired by SEEDS to address this issue and leverages the strong temporal coherence in high frame rate videos. Instead of processing all pixels, only the boundary pixels of superpixels in the current frame are considered. To identify the boundary pixels, a 4x4 search block is employed. If a pixel in the second row and second column has a different label when compared to its neighboring pixels to the right or below it, these two pixels are designated as boundary pixels. The distances between the boundary pixel and these two superpixel centers are computed in the assignment step. These boundary pixels are moved to the superpixel with a smaller distance. By adopting this methodology, the pixels inside each superpixel remain stationary during processing. It results in a high label consistency in the proposed system.

Additionally, the original SLIC algorithm uses only color and proximity Euclidean distances in its distance calculation. However, relying solely on the information of superpixel centers results in less compact superpixels. A compactness coefficient is proposed. The proposed distance metric is formulated as shown in Eq. (4). E represents the Euclidean distance, E_c is the color Euclidean distance and E_p is the proximity Euclidean distance. These distances are calculated the same as the original SLIC algorithm. CO refers to compactness. S and P represent the size and perimeter of the superpixel. The parameter γ determines the degree of influence that the compactness coefficient has in the overall distance calculation.

$$E = E_c + E_p \times CO, \tag{4}$$

$$CO = 1 + \gamma(\frac{S}{(\frac{P}{4})^2} - 1)^2. \tag{5}$$

The calculation of the compactness coefficient is based on the ratio of the superpixel's size to its perimeter divided by four squared. Dividing by 4 simulates the perimeter of a square divided by 4, which is equivalent to calculating the side length of the square. A CO value closer to 1 indicates a more square-like shape, signifying a higher level of compactness. In this situation, color and proximity distances contribute equally to the overall distance calculation. However, as the compactness deviates further from 1, the proximity distance becomes more influential in the final distance calculation. Because the boundary pixel only moves to the superpixel with a smaller distance, compactness coefficient results in more compact superpixels.

3.2 Selective Label Propagation Based Pipeline

In the assignment step, boundary pixels compute distances to both adjacent superpixel centers and are assigned to the center with the smaller distance. To identify boundary pixels, a 4x4 search block is employed. If the pixel in the second row and second column has a different label compared to its neighboring pixels on the right or below it, these two neighboring pixels are regarded as boundary pixels. Consequently, the pixel in the second row and second column, the pixel to its right, and the pixel below it may potentially change their labels during the assignment step. Importantly, the pixel in the third row is only involved in the computation for the subsequent line of the image. The delay of distance calculation is considerably lower than the time required for transmitting pixels in one line. Therefore, any potential label change of the pixel in the third row does not adversely affect the system's pipeline. However, if the pixel in the third column undergoes a label change during the current block calculation, it impacts the computation of the next block. Waiting for the completion of the current block results in a long delay. To ensure timely processing of the camera's pixel stream and achieve full pipelining in the proposed system, it is crucial to eliminate the data dependency between adjacent blocks.

An example of this problem is shown in Fig. 2 to provide a clearer illustration of this issue. Labels 0 and 1 are used as examples to represent different labels. Different colors are employed to provide a clearer visual distinction between these labels. Within the current block n's 4x4 calculation, the boundary pixels inside the red lines have the possibility to move either to the superpixel labeled as 0 or the one labeled as 1. If the right pixel remains labeled as 1, there are no boundary pixels in the second row in the next block n+1 calculation. However, if the right pixel changes to the label 0, the movement of boundary pixels still needs to be calculated in the next block n+1 calculation. In summary, due to the dependency on the current calculation's results, the next block requires different computations and yields distinct results. As a consequence, the next calculation has to wait for the completion of the current calculation.

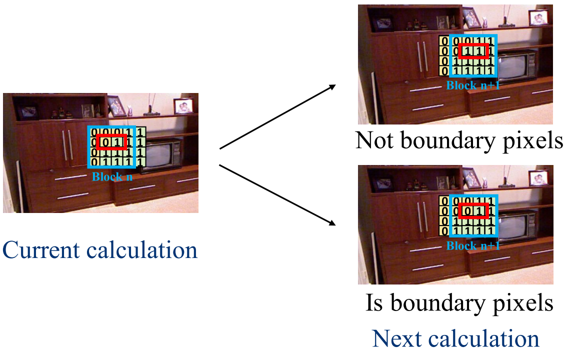
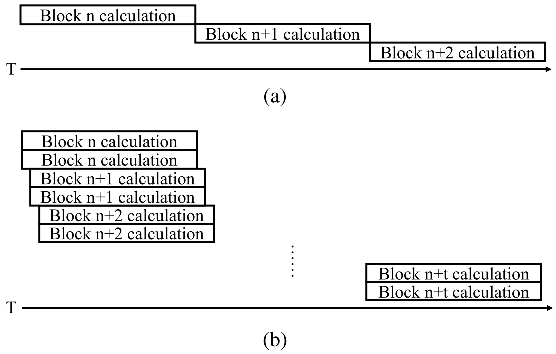


Fig. 2 Example of data dependency between adjacent blocks.



(a)

(b)

Fig. 3 Conceptual difference. (a) Original boundary pixels movement calculation; (b) Proposed selective label propagation based pipeline.

The data dependency between adjacent blocks causes long delays and poses challenges for the timely processing of high frame rate images. To address this issue, a selective label propagation based pipeline architecture is proposed. The conceptual difference is shown in Fig. 3. In each block calculation, boundary pixels are searched and moved to the superpixel with a smaller distance. Regardless of the label obtained from the previous block's calculation, all possible label configurations for the second row and second column of the current block are calculated in parallel. These results are stored in registers. After completing the calculation of the previous block, the calculation of the current block is selected based on the the label of previous one. The labels for the entire frame are selective based on label propagation from the first to the last block. The proposed structure is able to deal with input pixel stream from camera sensing

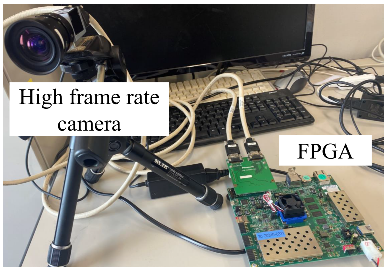


Fig. 4 Components of the high frame rate and ultra-low delay system for realworld applications. High frame rate camera is BASLER acA2000-340, FPGA is Xilinx Zynq UltraScale+ MPSoC ZCU104.

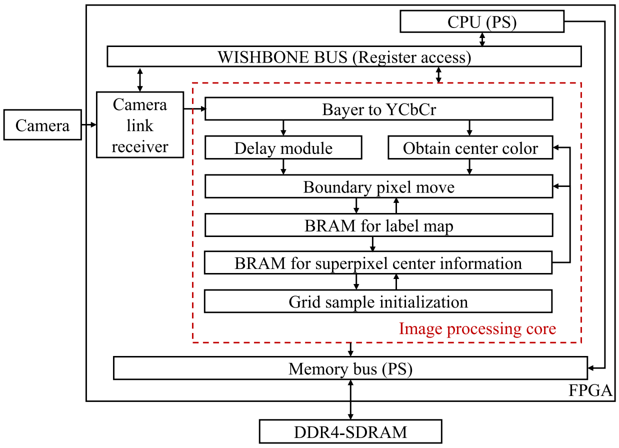


Fig. 5 Hardware structure of the proposed SLIC superpixel segmentation system.

promptly. It enables a fully pipelined architecture to achieve ultra-low delay at the same time.

3.3 Hardware Implementation

The proposed system is implemented in the hardware system depicted in Fig. 4. It comprises an industrial high frame rate camera and an FPGA. The industrial camera captures video data and transmits it to FPGA in a pixel stream. The superpixel segmentation system is implemented in the image processing core on the FPGA, as illustrated in Fig. 5. The processing system (PS) is implemented on one chip with programmable logic. The pixel stream from the camera is converted into a parallel pixel stream with a frequency of 100 MHz using the camera link receiver. This parallel pixel stream is the input to the image processing core. The output result of the image processing core is the label assigned to each pixel. It is transmitted to PS for further post-processing via WISHBONE BUS. Additionally, DDR4-SDRAM as an external memory can be employed for post-analysis.

Details in image processing core will be explained. The process of converting from RGB to CIELAB involves multiple divisions and exponential operations. These calculations demand substantial memory and computational resources in hardware design. In order to avoid these resource-intensive mathematical computations while still retaining the perceptual lightness channel, YCbCr color space is employed. Upon receiving the pixel stream from the camera link receiver, the pixels are converted to YCbCr format for subsequent processing. Obtain center color module utilizes the information of the previous frame's superpixel centers, which is stored in BRAM. When the coordinates of an input pixel match those of a center, its color information is saved for the assignment calculation. To ensure sufficient center color information, a delay of 20 lines is added to the pixel stream through a delay module. Within the boundary pixel move module, the pixel stream undergoes a transformation into a 4x4 block for the color distance computation. Boundary pixel move module takes as input the coordinates and color information of the superpixel center, as well as the superpixel's size and perimeter from BRAM. Additionally, a 4x4 block of labels from memory is provided as input to this module. Within this module, all possible calculations are performed in parallel. Selective label propagation based pipeline enables a pipelined boundary pixel move module, resulting in a fully pipelined hardware structure in the proposed system.

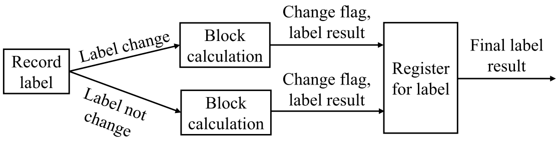


Fig. 6 Details of boundary pixel move module.

The details within boundary pixel move module are depicted in Fig. 6. Within this module, the previous labels of input pixels are first recorded and the possible label is calculated. By incorporating this information, the 4x4 label block is turned into two blocks. One represents labels where changes occur in the second row and second column due to the previous block calculation, and the other where labels remain unchanged. These two blocks calculate parallelly and output the change flag and this block's label result. The change flag indicates whether the label in the second row and third column changes or not. These outputs are stored in the register. Upon completing computations for the previous block, the final label result for the current block can be selectively determined based on the change flag. As a consequence of the pipelining of boundary pixel move module facilitated by the proposal, the output results are likewise pipelined for subsequent processing steps.

The label results obtained from boundary pixel move module are stored in the BRAM for label map. A ping-pong BRAM structure is utilized to enable synchronous reading of the previous frame's label map and writing of the current frame's label map. After finishing this frame's reading and writing, the functions of these BRAMs are exchanged in preparation for the next frame. Concurrently, the output label results are utilized to compute the current frame's superpixel center information. The center information is then stored in BRAM. Upon processing the last pixel of the current frame, the center information is sent to grid sample initialization module. After adding the initialization, the center information is then stored in BRAM for the next frame's processing. The whole system is fully pipelined in accordance with the input pixel stream.

4. Experimental Results

4.1 Algorithm Evaluation

4.1.1 Dataset and Evaluation Metrics

Because this paper is primarily focused on the fields of factory automation and robotics, the experiments are conducted on both factory assembly line dataset and the indoor scenes dataset. To simulate the factory assembly line scene, component images from the Halcon example images [22] are used to generate this dataset. To simulate indoor scenes for robotics, images from the NYUV2 dataset [23] are used. NYUV2 dataset shows varying indoor scenes of private apartments and commercial accommodations. The videos in both datasets have a resolution of 500x340. Each dataset contains four different motion patterns including horizontal translation, vertical translation, rotation, and scale change. Examples of the datasets are shown in Fig. 7.

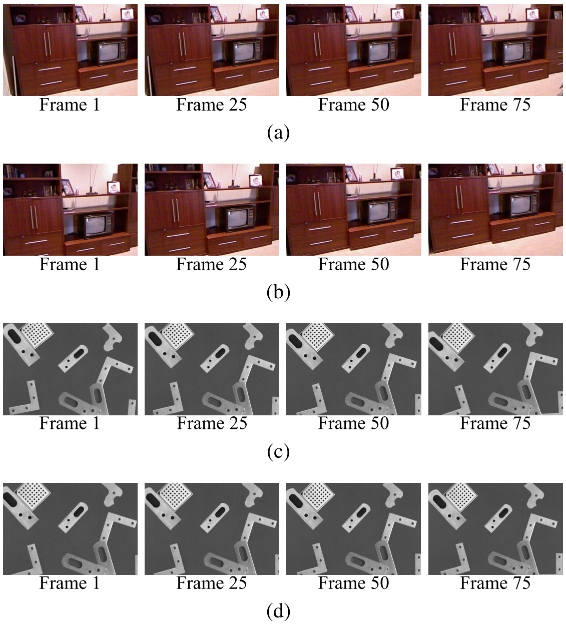


Fig. 7 Examples of factory assembly line dataset and indoor dataset. (a) Horizontal translation; (b) Vertical translation; (c) Rotation; (d) Scale. Horizontal and vertical translation datasets move at 1 pixel/frame; Rotation dataset rotates at 0.1 degree/frame; Scale dataset scales at 0.001 times/frame.

To evaluate the segmentation results objectively, three common metrics are used in this paper. Boundary Recall (BR) is used to evaluate boundary adherence. Undersegmentation Error (UE) is used to evaluate leakage or bleeding of a superpixel with groundtruth. Achievable Segmentation Accuracy (ASA) is used to evaluate the upper bound on the achievable segmentation accuracy using superpixel as a pre-processing step. Compactness (CO) [24] is employed to evaluate the visual quality of algorithms. Moreover, label consistency (LC) [25] is employed to evaluate the stability of superpixels across consecutive frames in a video sequence. LC shows how well superpixels track parts of objects, and it plays a crucial role in subsequent tasks such as classification. When label consistency is low, it may result in unstable or flickering classification outcomes [26].

4.1.2 Evaluation Results and Analysis

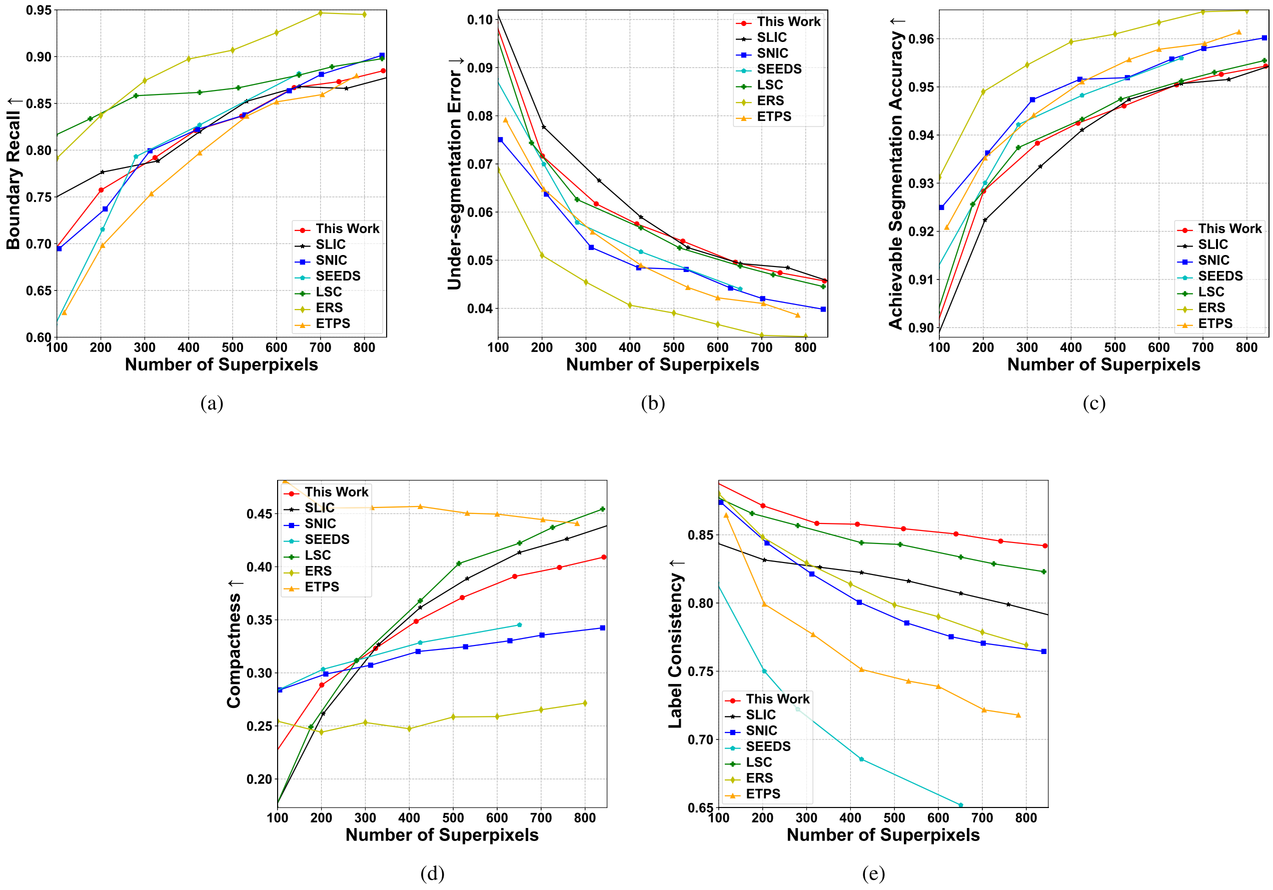
The evaluation results based on the mentioned evaluation metrics are presented in Table 1 and Table 2. In the experiments, ε and γ are set to 4. The expected superpixel number is set to 400. The proposed system maintains comparable values of BR, UE, and ASA for both datasets when compared with SLIC. While the boundary pixels move along with the motion and result in a decrease in CO, the utilization of temporal information in the proposed system leads to an average

Table 1 Evaluation results on indoor datasets.

	Horizontal translation		Vertical translation		Rotation		Scale change	
	SLIC	Proposed	SLIC	Proposed	SLIC	Proposed	SLIC	Proposed
BR ↑	0.820	0.825	0.818	0.807	0.814	0.814	0.813	0.812
UE ↓	0.059	0.057	0.056	0.057	0.064	0.063	0.064	0.062
ASA ↑	0.941	0.943	0.944	0.943	0.936	0.937	0.936	0.938
CO ↑	0.362	0.350	0.374	0.360	0.382	0.370	0.381	0.380
LC ↑	0.822	0.858	0.801	0.839	0.885	0.915	0.789	0.813

Table 2 Evaluation results on factory assembly line datasets.

	Horizontal translation		Vertical translation		Rotation		Scale change	
	SLIC	Proposed	SLIC	Proposed	SLIC	Proposed	SLIC	Proposed
BR ↑	0.986	0.990	0.992	0.979	0.988	0.994	0.986	0.991
UE ↓	0.022	0.018	0.020	0.020	0.027	0.024	0.027	0.024
ASA ↑	0.978	0.982	0.980	0.980	0.973	0.976	0.973	0.976
CO ↑	0.526	0.445	0.530	0.457	0.530	0.465	0.530	0.468
LC ↑	0.862	0.877	0.862	0.880	0.922	0.943	0.834	0.847



(a) (b) (c)

(d) (e)

Fig. 8 Quantitative evaluation results. (a) Boundary recall; (b) Under-segmentation error; (c) Achievable segmentation accuracy; (d) Compactness; (e) Label consistency.

Table 3 Comparisons with deep learning algorithms in indoor dataset.

	Horizontal translation			Vertical translation			Rotation			Scale change		
	SCN	AINet	Proposed	SCN	AINet	Proposed	SCN	AINet	Proposed	SCN	AINet	Proposed
BR ↑	0.892	0.892	0.873	0.927	0.903	0.882	0.919	0.903	0.881	0.916	0.908	0.875
UE ↓	0.035	0.036	0.047	0.031	0.034	0.045	0.040	0.043	0.052	0.041	0.043	0.053
ASA ↑	0.965	0.964	0.953	0.969	0.966	0.955	0.960	0.957	0.948	0.959	0.957	0.947
CO ↑	0.369	0.350	0.399	0.373	0.353	0.404	0.369	0.348	0.417	0.369	0.349	0.421
LC ↑	0.843	0.823	0.845	0.806	0.786	0.824	0.948	0.933	0.917	0.781	0.762	0.784

increase of more than 2.4% in LC when compared to SLIC.

To demonstrate the effectiveness of the proposed methods, comparisons with other well-known algorithms are conducted. Quantitative experiments are performed using the horizontal translation of the indoor dataset as an example. The comparisons between the proposed method and SLIC, SNIC, SEEDS, LSC, ERS, ETPS are shown in Fig. 8. Among all the algorithms, ERS shows superior performance on BR, UE, and ASA. However, due to its global processing and complex calculations, implementing ERS as an ultra-low delay system is challenging. As illustrated in the figure, although the proposed method is designed for an ultra-low delay system, its performance is comparable to that of other algorithms. Moreover, the proposed method shows the highest performance in terms of label consistency.

Comparisons with deep learning methods are also conducted on indoor datasets. The pre-trained modules from SCN [9] and AINet [10] are utilized in the experiments. To ensure fairness, the expected superpixel number of the proposed system is set as 750. Results are shown in Table 3. Deep learning methods achieve superior segmentation performance at the expense of huge computational amounts and memory costs. Notably, the proposed method achieves higher LC when compared to deep learning methods in most cases.

4.1.3 Ablation Study

To demonstrate the effectiveness of compactness coefficient, experimental results for different values of γ are presented in Table 4. These experiments encompass both indoor and factory assembly line datasets. Horizontal translation is se-

Table 4 Ablation studies of different influence degrees of compactness coefficient.

Dataset		0	1	2	3	4	5	6
Indoor	BR ↑	0.844	0.833	0.833	0.823	0.825	0.816	0.817
	UE ↓	0.058	0.058	0.057	0.059	0.057	0.058	0.058
	ASA ↑	0.942	0.942	0.943	0.941	0.943	0.942	0.942
	CO ↑	0.315	0.333	0.337	0.344	0.350	0.353	0.354
	LC ↑	0.853	0.859	0.860	0.858	0.858	0.858	0.854
Factory assembly line	BR ↑	0.989	0.986	0.983	0.989	0.990	0.985	0.984
	UE ↓	0.019	0.019	0.020	0.019	0.018	0.019	0.020
	ASA ↑	0.981	0.981	0.980	0.981	0.982	0.981	0.980
	CO ↑	0.440	0.447	0.446	0.446	0.445	0.443	0.438
	LC ↑	0.876	0.880	0.879	0.878	0.877	0.878	0.875

Table 5 Hardware performance and resource utilization of the proposed system.

Item		Hardware performance
Resource utilization	# LUT	177618 (77.09%)
	# LUTRAM	3281 (3.22%)
	# FF	227166 (49.30%)
	# BRAM	188 (60.26%)
	# DSP	210 (12.15%)
Performance	Frequency	100 MHz
	Delay per frame	0.985 ms

lected as an illustrative example. Images in indoor datasets tend to feature more irregular shapes when compared to those in the factory assembly line dataset. When γ is set to 0, it signifies the absence of compactness coefficient. When γ is configured as 4, CO of indoor dataset is improved by 11%. In contrast, for the factory assembly line dataset, where superpixels exhibit more regular shapes, the improvement in CO is 1%. For both datasets, as CO increases, the other evaluation metrics including UE, ASA, and LC maintain values similar to those observed in the situation without the use of the compactness coefficient. These results highlight the superior performance of the compactness coefficient, particularly when dealing with superpixels of more irregular shapes.

Nonetheless, it's important to recognize the existence of a trade-off between CO and BR. As CO increases, BR decreases for both datasets. Notably, setting γ to 6 results in a 3% reduction of BR in the case of the indoor dataset. This effect will be even more pronounced when dealing with superpixels exhibiting more irregular shapes. Furthermore, as γ becomes larger, the improvement in CO becomes less evident. In scenarios with regular-shaped superpixels, CO may even decrease with larger values of γ.

4.2 Hardware Evaluation

The effectiveness of the proposed hardware-oriented system is stated in the previous subsection. The processing speed and hardware resource utilization are other important issues because the ultimate target of this work is to develop a high frame rate and ultra-low delay SLIC superpixel segmentation system. The hardware synthesis and implementation are conducted on ZCU104 FPGA board using Vivado 2021.2. BRAM supports up to 512 superpixels. The results of the hardware evaluation are presented in Table 5.

For the hardware resource utilization, the BRAM utilization percentage exceeds 60% due to the storage of the whole frame's label map and superpixel center information. Additionally, a ping-pong BRAM structure is deployed for label maps. It causes an increase in BRAM utilization. LUT serves not only for combinatorial logic calculations but also for the delay module to receive the pixel stream. As a result, the LUT utilization reaches 77%.

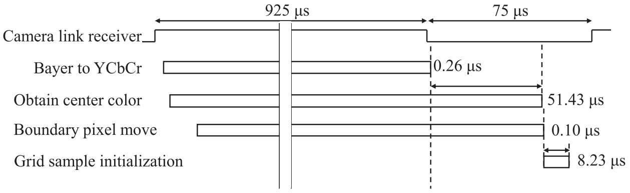


Fig. 9 Detailed timing flow of the proposed system.

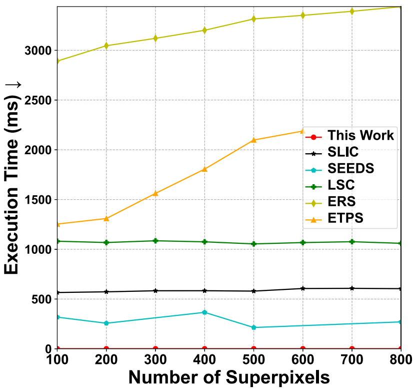


Fig. 10 Comparisons of execution times.

The processing delay from the sensor input to the whole frame result output is 0.985 ms/frame. The timing flow of the proposed system is illustrated in detail in Fig. 9. Obtain center color module requires a waiting period of 20 lines to ensure sufficient center color information for the current frame. It results in a delay of over 50 µs. In grid sample initialization module, the sequential processing of reading and writing center information in the BRAM introduces a delay of 8.23 µs.

The comparisons of execution times between the proposed system and other algorithms are illustrated in Fig. 10. SLIC, SEEDS, LSC, ERS, and ETPS are all provided by a superpixel benchmark [27]. The execution time of this work

Table 6 Comparison of the proposed system with other FPGA implementations of SLIC algorithm.

	[13]	[14]	[21]	This work
Iteration strategy	Remaining iterations	Reducing iterations	Remaining iterations	Temporal iterations
Delay	6.99 ms	3.86 ms	41.67 ms	0.985 ms
Image resolution	300x400	481x321	300x400	640x360

is the delay of image processing core. It is from the first-pixel sensor input to the whole frame result output. Notably, as the number of superpixels increases, the delay associated with the grid sample initialization also increases. However, even when the expected number of superpixels reaches 800, the delay remains remarkably low at 0.988 ms, and it still reaches the requirement of under 1 ms. This performance is significantly faster when compared to other algorithms.

The results presented in Table 6 illustrate the comparisons between the proposed system and other FPGA implementations of SLIC algorithm. By leveraging a temporal iteration strategy, iterations of SLIC algorithm are separated into the temporal domain, ensuring an ultra-low delay. The delay in this work is significantly lower when compared to other implementations of the SLIC algorithm on FPGA.

5. Conclusion

The system described in this paper is designed for superpixel segmentation with a high frame rate and low delay. Both the algorithm and hardware implementation have been presented in this paper. From an algorithmic perspective, iterations inside one image processing are separated into the temporal domain. Grid sample initialization and compactness coefficient are proposed for the stability of superpixel. To make the proposed architecture fully pipelined to reach ultra-low delay, selective label propagation based pipeline is proposed. All the possibilities of labels are calculated parallelly, and the final output results are selected through label propagation. The experiments prove that the proposed system achieves a real-time processing speed of 1000 FPS with a delay of less than 1 ms/frame. Moreover, the system's performance remains comparable to other well-known algorithms while achieving higher LC compared to state-of-the-art methods. For future work, it is important to further optimize the hardware resource utilization of the proposed method. Since superpixel segmentation serves as a pre-processing step for various applications, it is crucial to allocate sufficient resources for subsequent processing stages.

Acknowledgments

This work was supported by KAKENHI (21K11816).

References

[1] J.S. Rentmeister, M.H. Kiani, K. Pister, and J.T. Stauth, "A 120–330v, sub-μa, 4-channel driver for microrobotic actuators with wireless-optical power delivery and over 99% current efficiency," IEEE Symp. VLSI Circuits, pp.1–2, IEEE, 2020.

[2] T. Hu, R. Fuchikami, and T. Ikenaga, "High temporal resolution-based temporal iterative tracking for high framerate and ultra-low delay dynamic tracking system," IEICE Trans. Inf. & Syst., vol.E105-D, no.5, pp.1064–1074, 2022.

[3] S. Du, Z. Dong, Y. Li, and T. Ikenaga, "Straight-line detection within 1 millisecond per frame for ultra-high-speed industrial automation," IEEE Trans. Ind. Informat., vol.19, no.4, pp.5965–5975, 2023.

[4] S. Du, K. Gu, and T. Ikenaga, "Subpixel displacement measurement at 784 fps: From algorithm to hardware system," IEEE Trans. Instrum. Meas., vol.71, pp.1–10, 2022.

[5] Z. Zhou, Y. Guo, J. Huang, M. Dai, M. Deng, and Q. Yu, "Superpixel attention guided network for accurate and real-time salient object detection," Multimed. Tools. Appl., vol.81, no.27, pp.38921–38944, 2022.

[6] H. Wang, W. Zhou, X. Zhang, and X. Lou, "A block patchmatch-based energy-resource efficient stereo matching processor on fpga," IEEE Trans. Circuits Syst. I: Regul. Pap., vol.69, no.7, pp.2893–2905, 2022.

[7] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," IEEE Trans. Pattern Anal. Mach. Intell., vol.34, no.11, pp.2274–2282, 2012.

[8] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.1356–1363, 2015.

[9] F. Yang, Q. Sun, H. Jin, and Z. Zhou, "Superpixel segmentation with fully convolutional networks," Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.13964–13973, 2020.

[10] Y. Wang, Y. Wei, X. Qian, L. Zhu, and Y. Yang, "Ainet: Association implantation for superpixel segmentation," Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp.7078–7087, 2021.

[11] J.-H. Bae, G.-H. Yu, J.-H. Lee, D.T. Vu, L.H. Anh, H.-G. Kim, and J.-Y. Kim, "Superpixel image classification with graph convolutional neural networks based on learnable positional embedding," Appl. Sci., vol.12, no.18, p.9176, 2022.

[12] C.Y. Ren and I. Reid, "gslic: a real-time implementation of slic superpixel segmentation," University of Oxford, Department of Engineering, Technical Report, pp.1–6, 2011.

[13] M. Mighani and A. Khakpour, "Fmslic: Fast memory-efficient structure for implementation of slic on fpga," Circuits, Syst. Signal Process., vol.42, no.8, pp.5065–5078, 2023.

[14] A. Ghaderi, C. Ahlberg, M. Östgren, F. Ekstrand, and M. Ekström, "Fp-slic: A fully-pipelined fpga implementation of superpixel image segmentation," Euromicro Conf. Digit. Syst. Des. (DSD), pp.109–117, IEEE, 2022.

[15] Y. Li, T. Hu, R. Fuchikami, and T. Ikenaga, "Grid sample based temporal iteration and compactness-coefficient distance for high frame and ultra-low delay slic segmentation system," Int. Conf. Mach. Vis. Appl. (MVA), 2023.

[16] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.2097–2104, IEEE, 2011.

[17] H. Wang, J. Shen, J. Yin, X. Dong, H. Sun, and L. Shao, "Adaptive nonlocal random walks for image superpixel segmentation," IEEE Trans. Circuits Syst. Video Technol., vol.30, no.3, pp.822–834, 2020.

[18] J. Yao, M. Boben, S. Fidler, and R. Urtasun, "Real-time coarse-to-fine topologically preserving segmentation," Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.2947–2955, 2015.

[19] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool, "Seeds: Superpixels extracted via energy-driven sampling," Int. J Comput. Vis., vol.111, no.3, pp.298–314, 2015.

[20] R. Achanta and S. Susstrunk, "Superpixels and polygons using simple non-iterative clustering," Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.4651–4660, 2017.

[21] P.A. Khamaneh, A. Khakpour, M. Shoaran, and G. Karimian, "Real–time memory efficient slic accelerator for low–power applications,"

Multimed. Tools. Appl., vol.81, no.22, pp.32449–32467, 2022.
[22] Halcon, Available online: https://linx.jp/product/mvtec/halcon/.
[23] S. Song, S.P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.567–576, 2015.
[24] A. Schick, M. Fischer, and R. Stiefelhagen, "Measuring and evaluating the compactness of superpixels," Proc. Int. Conf. Pattern Recognit., pp.930–934, IEEE, 2012.
[25] J. Chang, D. Wei, and J.W. Fisher, "A video representation using temporal superpixels," Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp.2051–2058, 2013.
[26] O. Miksik, D. Munoz, J.A. Bagnell, and M. Hebert, "Efficient temporal consistency for streaming video scene analysis," Int. Conf. Robot. Autom. (ICRA), pp.133–139, IEEE, 2013.
[27] D. Stutz, A. Hermans, and B. Leibe, "Superpixels: An evaluation of the state-of-the-art," Comput. Vis. Image Underst., vol.166, pp.1–27, 2018.

Yuan Li received the B.E. degree in information engineering from Southeast University, Nanjing, China, in 2018, and the M.E. degree in engineering of information, production and systems, in 2019, from Waseda University, Kitakyushu, Japan, where she is currently working towards the Ph.D. degree. Her research focuses on hardware implementation of computer vision algorithms.

Tingting Hu received her B.E. degree in School of Automation from the Beijing Institute of Technology, China, in 2016, and her M.E. and Ph.D. degrees from the Graduate School of Information, Production, and Systems of Waseda University, Japan, in 2017 and 2023, respectively. She joined Panasonic Corporation, Japan, in 2018. Her current research interests are high frame rate and ultra-low delay vision systems.

Ryuji Fuchikami received his B.Sc. degree in Kyushu Institute of Technology, Japan, in 1999. He joined Panasonic Corporation (Formerly Matsushita Electric Industrial Co., Ltd.), Japan, in 2000. He had been undertaking research on the LSI design of video codec. His current research interests are high-speed visual systems.

Takeshi Ikenaga received his B.E. and M.E. degrees in electrical engineering and Ph.D degree in information computer science from Waseda University, Tokyo, Japan, in 1988, 1990, and 2002, respectively. He joined LSI Laboratories, Nippon Telegraph and Telephone Corporation (NTT) in 1990, where he had been undertaking research on the design and test methodologies for high performance ASICs, a real-time MPEG2 encoder chip set, and a highly parallel LSI system design for image understanding processing. He is presently a professor in the integrated system field of the Graduate School of Information, Production and Systems, Waseda University. His current interests are image and video processing systems, which covers video compression (e.g. VVC, SCC), video filter (e.g. super resolution, high-dynamic range imaging), and video recognition (e.g. sports analysis, ultra-high speed and ultra-low delay vision system). He is a senior member of the Institute of Electrical and Electronics Engineers (IEEE), a member of the Institute of Electronics, Information and Communication Engineers of Japan (IEICE) and the Information Processing Society of Japan (IPSJ).