Real-Time Multi-Object Tracking Based on Highly Parallel Image Segmentation and Pattern Matching

Tetsushi Koide, Hans Juergen Mattausch, Takashi Morimoto, Hidekazu Adachi, Kosuke Yamaoka

Research Center for Nanodevices and Systems, Graduate School of Advanced Sciences of Matter, Hiroshima University
1-4-2 Kagamiyama, Higashi-Hiroshima 739-8527, Japan, e-mail: {koide, hjm}@sxsys.hiroshima-u.ac.jp

1. Introduction

The vision-based intelligent processing by associative memory-based systems with recognition and learning capability requires image segmentation of objects and feature extraction/modeling as front-end technologies. We are investigating and implementing new image segmentation and feature extraction/modeling methodologies suitable for associative memory-based systems. Furthermore, technologies for object tracking and image recognition are essential in research fields like robot vision and intelligent transport systems. In this paper, we present a currently developed multi-object tracking architecture, using our proposed image segmentation framework for real-time moving pictures [5, 8] and the pattern matching functionality of our associative memories [3, 4], to enable vision-based intelligent processing.

2. Multi-Object Tracking Algorithm

Several moving object tracking algorithms and architectures have already been proposed. Most of them are based on difference evaluation between the current image and a previous image or a background image [1, 9]. However, algorithms based on the difference of images have problems with following practical cases. (1) Still objects included in the tracking task exist. (2) Multiple moving objects are present in the same frame. (3) The camera is moving. (4) Occlusion of objects occurs. Our proposed algorithm for object tracking [7], based on image segmentation and pattern matching, aims at solving above problems.

In this algorithm we extract all objects from an input image by image segmentation. Next we extract simple object features and use these features to form standardize patterns for representing the objects. Then we compare the features of extracted objects in the current frame with those of extracted objects in the preceding frame by pattern matching. The most similar objects (i.e. the objects which have the smallest distance) between successive frames are judged to be corresponding objects. A coarse flow chart of the proposed algorithm is shown in Fig. 1. In spite of the motion condition of objects, the proposed algorithm is effective because each object's motion vector is determined and used as one of its features. Additionally, we can increase the number of extracted object features from segmentation results, so that detection and tracking accuracy can be improved to a suitable level.

3. Multi-Object Tracking Architecture

Fig. 2 shows the overall block diagram of the developed FPGA/ ASIC implementation architecture. This architecture roughly consists of 4 blocks. The first block is the image segmentation cell-network in which all objects of the frame are extracted. The second block is the feature extraction block in which object features for each segmented object are calculated using the image segmentation results. The third block is the pattern matching block in which the most similar object is searched among the reference data from the previous frame. The fourth block is the estimated position calculation block in which the estimated position of each object in the next frame is calculated.

For the segmentation part we exploit a previously developed cell-network based on a digital image segmentation architecture [6, 8] (Fig. 3). The image segmentation cell-network implements a region-growing algorithm and has the structure of a two-dimensional array of image segmentation cells corresponding to the pixels of an input image. By taking advantage of the cell-network, we can access the segmentation result of each cell in parallel in x-direction and in y-direction. This is done in the feature extraction block where the width of each segmented object is calculated from the cell-network data which is outputted in parallel into y-direction and where the height and the area of each object are calculated from the cell-network data which is outputted in parallel into x-direction. Then the determined object-features are transmitted to the pattern matching block so as to search the most similar object among the reference data in the previous frame. In the estimated position calculation block, the estimated position of the current input object in the next frame is calculated from the positions of the matched object and the input object. Then the estimated position is stored in the pattern matching block as one feature of the input object's reference pattern for matching with the objects of the next frame.

Due to the sequential nature of the segmentation by region growing, we can apply pipeline processing, as shown in Fig. 4, to interleave the processing steps of image segmentation, feature extraction and pattern matching. As compared to the case of completing the frame's segmentation before advancing to feature extraction, higher processing speed of the complete algorithm is achieved, so that more objects can be tracked in real time.

4. Performance Evaluation of Proposed Architecture

For verifying the effectiveness of the proposed object tracking algorithm, we tested a sample picture sequence with 80x60 pixels per frame, consisting of four successive frames (30fps) as shown in Fig. 5. The sample sequence is an example, which includes multiple moving objects and also the occlusion effect among objects. Note that the object labels are explicitly shown in the pictures. The table in Fig. 5 shows Manhattan distances between the objects of successive frames. From the table in Fig. 5, we can confirm correct matching between objects in successive frames and thus confirm the validity of the proposed algorithm. Furthermore, we also have confirmed the effectiveness of the proposed algorithm for many other difficult cases such as rapid direction of movement changes by object collision, rotating complex objects or non-rigid objects like walking humans.

Fig. 6 shows the graph which plots the number of segmented regions vs. the processing time for three kinds of image size, VGA, QVGA and 80x60 pixels. The process of pattern matching with the sequential implementation requires much time. But, as shown in Fig. 4, we can shorten the total amount of time by executing image segmentation, feature extraction and pattern matching in a processing pipeline. In this way, about 187, 220, and 255 objects for VGA, QVGA, and 80x60 pixel image size, respectively, can be handled in a real-time tracking application (30fps) under the condition that the operating frequency is 20 MHz. Futhermore Fig. 6 shows, that as the image size becomes larger, the effect of pipeline processing on processing time becomes more significant.

5. Conclusion

We have proposed a multi-object tracking architecture for video pictures, based on image segmentation and pattern matching of the segmented objects between frames in a simple feature space. The suitability of the proposed algorithm was verified by simulation. Then we have proposed an FPGA / ASIC implementation architecture, realizing this algorithm for real time object tracking. The relative simplicity of this tracking algorithm promises that an FPGA implementation is possible and already sufficient for real-time applications with a few moving objects. We have also showed that it is sufficient for the tracking to use the simple Manhattan distance. Thus, VLSI implementation of the algorithm is possible by using our developed architectures for image segmentation [6, 8] and a fully parallel associative memory for high-speed minimum Manhattan distance search [2, 10], both of which have been already realized as

VLSI circuits. By using our associative memory as the pattern matching block, the number of objects which can be processed in real-time will be increased.

The next steps in our research effort towards the development of a prototype multi-object tracking system are : an FPGA implementation of the prototype system, and a VLSI circuit development based on our image segmentation and associative memory technologies.

Acknowledgments: The test-chips in this study have been fabricated in the chip fabrication program of VLSI Design Education Center (VDEC), the University of Tokyo in the collaboration with Rohm Corporation and Toppan Printing Corporation. Part of this work was supported by a Grant-in-Aid for Young Scientists (B) (No.16700184), Ministry of Education, Culture, Sports, Science and Technology, Japanese Government.

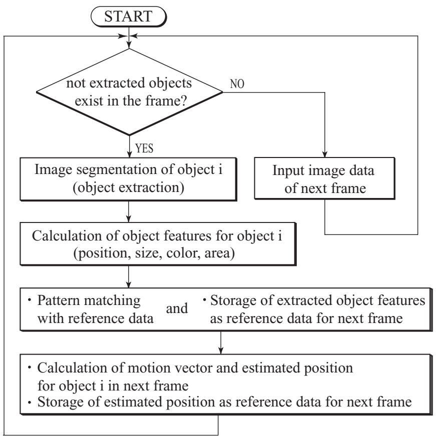


Figure 1: Flowchart of the proposed multi-object tracking algorithm based on image segmentation and pattern matching.

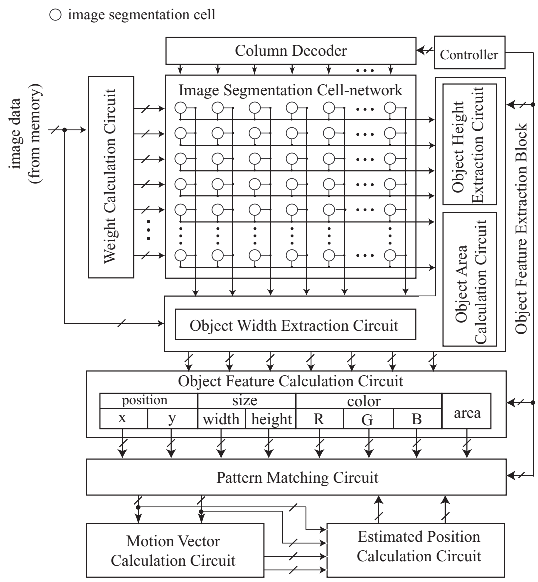


Figure 2: Block diagram of the proposed FPGA / ASIC implementation architecture.

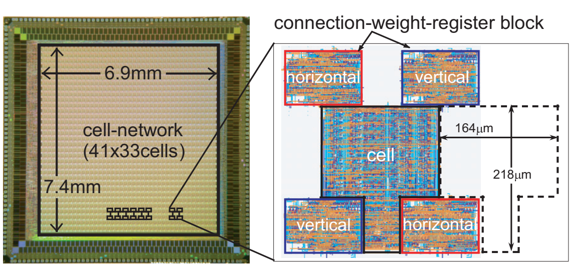


Figure 3: Die photo of the image segmentation cell-network including 41x33 cells designed in a 0.35um 3-metal CMOS technology.

References

[1] H. Kimura, et al., Ext. Abst. SSDM2002, pp. 240-241, 2002.
[2] T. Koide, et al., Ext. Abst. SSDM2004, pp. 360-361, 2004.
[3] H. J. Mattausch et al., IEEE JSSC, vol. 37, 2, pp.218-227, 2002.
[4] H. J. Mattausch et al., Symp. on VLSI Circuits, pp. 252-255, 2002.
[5] T. Morimoto et al., IEICE Trans. E87-D, 2, pp. 500-503, 2004.
[6] T. Morimoto et al., Ext. Abst. SSDM2004, pp. 138-139, 2004.
[7] T. Morimoto et al., Proc. ISCAS2005, pp.3215-3218, 2005.
[8] T. Morimoto et al., IEE Proc. Circuits, Devices & Systems, in press, 2005.
[9] S. W. Seol, et al., Proc. Int. Tech. Conf. on Circuits/Systems, Computers and Communications, pp. 260-263, 2003.
[10] Y. Yano, et al., Ext. Abst. SSDM2002, pp. 254-255, 2002.

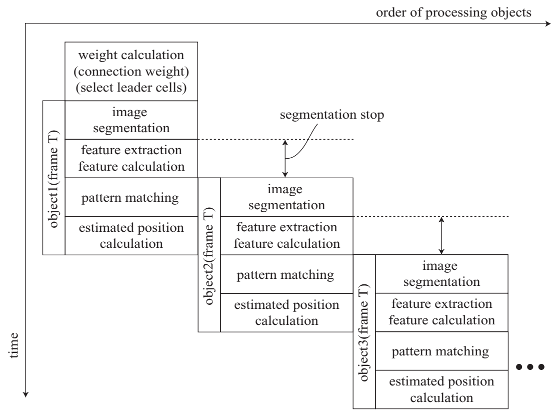
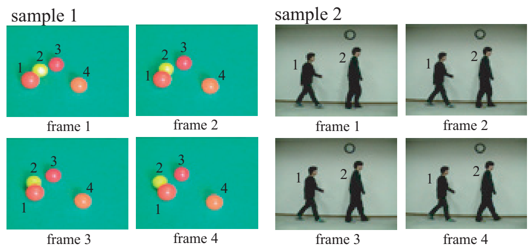


Figure 4: Process flow of the FPGA / ASIC implementation architecture when using pipline processing.



(a) The Manhattan distance between succesive frames for sample 1

	(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	**1**	19	10	10
(2,2)	18	**4**	19	21
(2,3)	12	18	**1**	11
(2,4)	13	23	12	**2**

	(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	**2**	17	10	13
(3,2)	19	**6**	19	26
(3,3)	10	19	**2**	15
(3,4)	10	25	14	**3**

	(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	**2**	17	12	10
(4,2)	19	**2**	23	27
(4,3)	10	21	**2**	14
(4,4)	11	24	13	**1**

(b) The Manhattan distance between succesive frames for sample 2

	(1,1)	(1,2)
(2,1)	**1**	12
(2,2)	12	**1**

	(2,1)	(2,2)
(3,1)	**2**	13
(3,2)	11	**2**

	(3,1)	(3,2)
(4,1)	**2**	10
(4,2)	12	**0**

Figure 5: Successive image sequences for algorithm simulation (80x60).

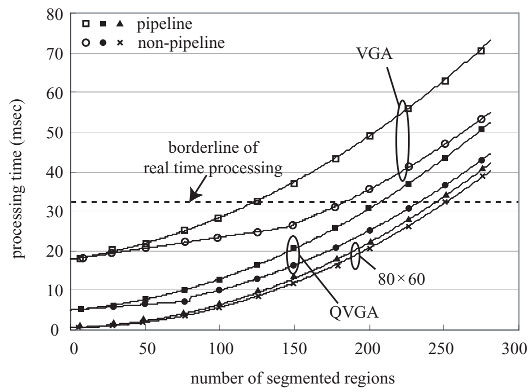


Figure 6: Relationship between the number of segmented regions and the estimated processing time.

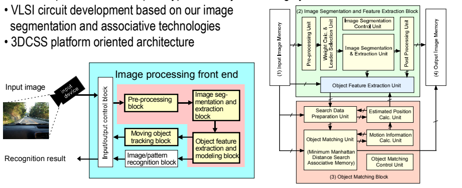
Real-Time Multi-Object Tracking Based on Highly Parallel Image Segmentation and Pattern Matching

NTIP
Hiroshima University
Research Center for Nanodevices and Systems (RCNS)
System Design and Architecture Research Division

Tetsushi Koide, Hans Juergen Mattausch,
Takashi Morimoto, Hidekazu Adachi, Kosuke Yamaoka
Research Center for Nanodevices and Systems, Hiroshima University

Image Processing Front End for Associative Memory-Based Systems

Research Target:
- Image segmentation and feature extraction for associative memory based systems
- Real-time image segmentation architectures (cell-network-based, block-scan-based)
- A fully parallel minimum distance search associative memory for object tracking
- Multi-object tracking based on image segmentation and pattern/object matching
- FPGA implementation of a prototype multi-object tracking system
- VLSI circuit development based on our image segmentation and associative technologies
- 3DCSS platform oriented architecture



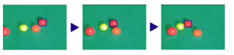
(a) A block diagram of image segmentation and feature extraction for real time application

(b) A block diagram of multi-object tracking based on image segmentation and pattern matching

1

Introduction: Multi-Object Tracking

Object tracking in video pictures is of great interest in e.g.
- computer vision
- navigation systems
- surveillance systems



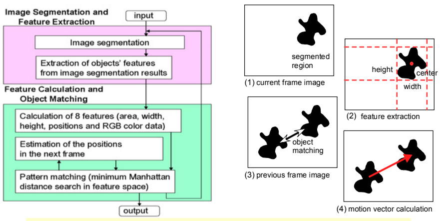
Conventional approaches, based on the image difference, contain problems in
- multiple objects tracking
- simultaneous tracking of still and moving objects
- the case of a moving camera
- the case of occlusion of objects occurs

Proposed novel approach to object tracking, based on
- image segmentation,
- pattern matching,

realizes an efficient, simple, and stable tracking of multiple moving/still objects

2

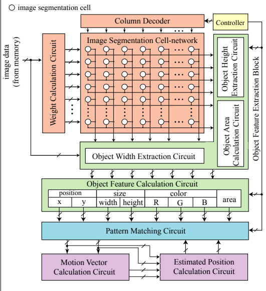
Proposed Multi-Object Tracking Algorithm



- The algorithm works well for pictures including multiple moving/still objects, occlusion of objects, rotating objects, and non-rigid objects
- If mistracking occurs at some frames (e.g. by reason of occlusion), the algorithm can recover correct tracking after a couple of frames
- Using object feature extraction and pattern matching techniques, the algorithm can be extended to object recognition

3

Proposed Multi-Object Tracking Architecture



(1) Image segmentation cell-network
All objects of the frame are extracted

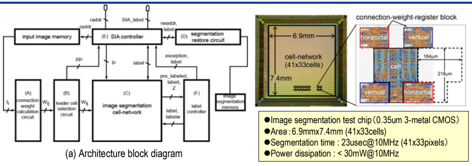
(2) Feature extraction block
Object features for each segmented object are calculated using the image segmentation results

(3) Pattern matching block
The most similar object is searched among the reference data from the previous frame

(4) Estimated position calculation block
The estimated position of each object in the next frame is calculated

4

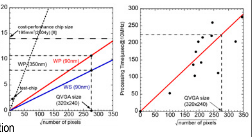
Image Segmentation Cell-Network Architecture



- Image segmentation test chip (0.35um 3-metal CMOS)
- Area : 6.9mmx7.4mm (41x33cells)
- Segmentation time : 23usec@10MHz (41x33pixels)
- Power dissipation : < 30mW@10MHz

(a) Architecture block diagram

- Digital color image segmentation algorithm based on region growing approach
- Fully pixel parallel architecture based on cell-network
- Development of low power real-time color image segmentation chip architecture
- Low power operation with boundary-active-only (BAO) scheme
- Proposal of a block-scan based image segmentation architecture for large-scale image segmentation



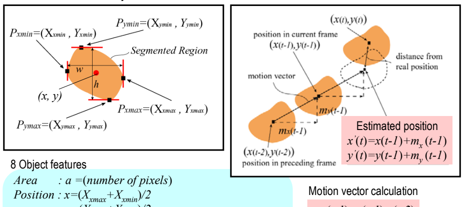
(b) Estimation of chip size and processing time

5

Feature Extraction of Segmented Object

Extraction of object features

Position estimation in the next frame



8 Object features

Area : a = (number of pixels)

Position : $x = (X_{xmax} + X_{xmin})/2$, $y = (Y_{ymax} + Y_{ymin})/2$

Size : $w = X_{xmax} - X_{xmin}$, $h = Y_{ymax} - Y_{ymin}$

Color : $R = [R(P_{xmax}) + R(P_{xmin}) + R(P_{ymax}) + R(P_{ymin})]/4$
$G = [G(P_{xmax}) + G(P_{xmin}) + G(P_{ymax}) + G(P_{ymin})]/4$
$B = [B(P_{xmax}) + B(P_{xmin}) + B(P_{ymax}) + B(P_{ymin})]/4$

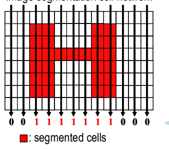
Estimated position
$x'(t) = x(t-1) + m_x(t-1)$
$y'(t) = y(t-1) + m_y(t-1)$

Motion vector calculation
$m_x(t-1) = x(t-1) - x(t-2)$
$m_y(t-1) = y(t-1) - y(t-2)$

The estimated position is exploited instead of the extracted position for pattern matching

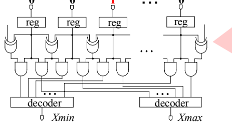
6

Object Feature Extraction Block



- Object width, height, and area are calculated by taking advantage of two-dimensional array structure of image segmentation cell-network.
- Status signals of segmented cells are transmitted to basic feature extraction circuit in parallel.

- Status signal
  1 : column belonging to the current segmented region
  0 : not belonging to the current segmented region



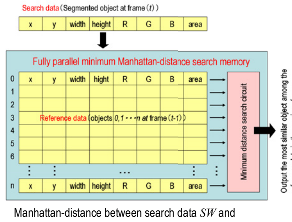
Width extraction circuit for the current segment

Basic Feature Extraction Circuit function
- Boundaries between 1 (column belonging to segmented region) and 0 are detected. (boundary is "1")
- These boundary information are transformed to the column index in the two decoder (Xmin, Xmax)

7

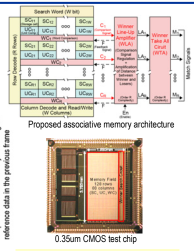
Pattern Matching with Manhattan-Distance Search Memory

Find the most similar object among the reference data in the previous frame



Manhattan-distance between search data SW and reference data REF

$$D_{Manhattan} = \sum_{i=1}^{W} |SW_i - REF_i|$$



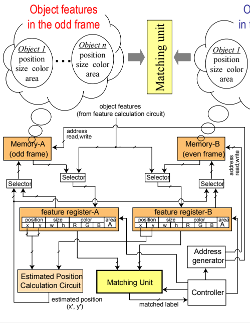
Proposed associative memory architecture

0.35um CMOS test chip

- ✓ Minimum-distance search time: < 240ns
- ✓ Power dissipation : < 260mW at 10MHz (34.7mW/mm²)
- ✓ Performance : 170GOPS (20GOPS/mm²)
- ✓ Search time in practical applications such as image compression: < 150ns

8

Pattern Matching Block for an FPGA Implementation



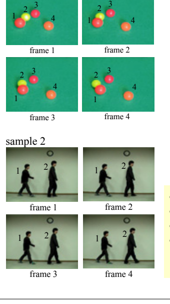
Reference Memory
- Storage of extracted object features.
  odd frame's data : Memory-A
  even frame's data : Memory-B

Matching Unit
- Calculation of Manhattan-distance between input object (n-th frame) and reference object (n-1-th frame) in the feature space.
- The reference object having smallest distance is selected to the most similar object.

Estimated Position Calculation Circuit
- Calculate position in the next frame using position information of matched object.
- Estimated position is stored to the reference memory to apply to the pattern matching of the next frame.

9

Experimental Verification for Object Tracking Algorithm



(a) The Manhattan distance between succesive frames for sample 1

	(1,1)	(1,2)	(1,3)	(1,4)
(2,1)	1	19	10	10
(2,2)	18	4	19	21
(2,3)	12	18	1	11
(2,4)	13	23	12	2

	(2,1)	(2,2)	(2,3)	(2,4)
(3,1)	2	17	10	13
(3,2)	19	6	19	26
(3,3)	10	19	2	15
(3,4)	10	25	14	3

	(3,1)	(3,2)	(3,3)	(3,4)
(4,1)	2	17	12	10
(4,2)	19	2	23	27
(4,3)	10	21	2	14
(4,4)	11	24	13	1

(b) The Manhattan distance between succesive frames for sample 2

	(1,1)	(1,2)
(2,1)	1	12
(2,2)	12	1

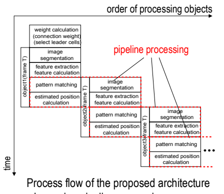
	(2,1)	(2,2)
(3,1)	2	13
(3,2)	11	2

	(3,1)	(3,2)
(4,1)	2	10
(4,2)	12	0

- Sample picture sequence with 80x60 pixels per frame (30fps)
- All objects correctly match with Manhattan-distance measure
- The use of Manhattan distance is sufficient for object tracking
- Effective for many other difficult cases such as rapid direction of movement changes by object collision, rotating complex objects or non-rigid objects like walking humans

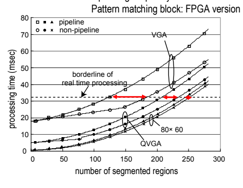
10

Number of Segmented Regions vs. Processing Time



Process flow of the proposed architecture when using pipeline processing

Operating frequency : 20MHz
Pattern matching block: FPGA version



- Pipeline processing to interleave the processing steps of image segmentation, feature extraction and pattern matching.
- As the image size becomes larger, the effect of pipeline processing on processing time becomes most significant.

	Number of segmented regions in real time (33msec)		improved ratio
	non-pipeline	pipeline	
80x60	247	254	+2.8%
QGGA	209	241	+15.3%
VGA	127	187	+47.2%

11

Conclusion

- Low Power Real-time Image Segmentation
  - Low-power region-growing architecture with real-time capability is developed and tested
  - VLSI integration for large image sizes with conventional CMOS technology verified by test chip designs
- Multi-Object Tracking using Image Segmentation and Pattern Matching
  - An algorithm capable of multi-object tracking, even for the moving camera case, developed and tested
  - A VLSI architecture for tracking-system realization developed and under verification in FPGA

Future Work

- Development of a Prototype Multi-Object Tracking System
  - An FPGA implementation of the prototype system
  - A VLSI circuit development based on our image segmentation and associative memory technologies
  - Adaptation of the architecture to 3DCSS platform

12