

High Performance Feature Transformation Architecture based on Bag-of-Features in CAD system for Colorectal Endoscopic Images

Koki Sugi ¹⁾, Tetsushi Koide ¹⁾, Anh-Tuan Hoang ¹⁾, Takumi Okamoto ¹⁾, Tatsuya Shimizu ¹⁾, Toru Tamaki ²⁾, Bisser Raytchev ²⁾, Kazufumi Kaneda ²⁾, Yoko Kominami ³⁾, Shigeto Yoshida ³⁾, Shinji Tanaka ³⁾

1) Research Institute for Nanodevice and Bio Systems Hiroshima University

2) Graduate School of Engineering, Hiroshima University

3) Graduate School of Biomedical Sciences, Hiroshima University

1-4-2 Kagamiyama, Higashi-Hiroshima, Japan

e-mail : { sugi-kouki, koide } @hiroshima-u.ac.jp

Abstract - Our research target to a computer-aided diagnosis (CAD) system for colorectal endoscopic images with narrow band imaging (NBI) magnification, which identifies a pathology type from local feature in the NBI endoscopic image. We propose a high speed feature transformation for CAD system by using Manhattan distance calculation and on the fly normalization method. A high performance and low cost algorithm for multiple Scan Window (SW) processing for FPGA is also introduced. The proposed high speed feature transformation can complete the transformation processing within 380 msec on a real time Full HD NBI endoscopic image.

I. Introduction

With the increase in the number of colorectal cancer patients, systems which support a doctor's diagnosis have been researched. The computer-aided diagnosis (CAD) system for colorectal endoscopic images with narrow band imaging (NBI) magnification [1] has already been proposed [2]. The proposed CAD system identifies 3 types of endoscopic image: Type A, Type B, and Type C3 in Fig.1 using D-SIFT feature and support vector machine (SVM). Currently, it is implemented in software and is able to identify the center region, is 120 x 120 pixel size of a Full HD input image at 14.7 fps. This software processing takes about 20 minutes to scan all of 120 x 120-pixel regions with 10 pixel pitch in a full high definition (Full HD) (1920 x 1080) image. Further improvement in speed is essential for realization of high performance Full HD image CAD system. So it is necessary to improve the software based algorithm to be suitable for hardware implementation. Our proposed system is expected to have latency within 1 second and have processing speed (throughput) within 1 ~ 5 fps. This paper describes the hardware implementation and improvement in feature transformation. The other processing in Fig.2 can be referred in reference [3], [4], and [5]. In the feature transformation, most of the processing is used on a distance

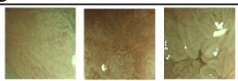


Type A (normal cells)	Microvessels are not observed or extremely opaque.	
Type B (tubular adenomas)	Fine microvessels are observed around pits, and clear pits can be observed via the nest of microvessels.	
Type C3 (cancer)	Pits via the microvessels are invisible, irregular vessel diameter is thick, or the vessel distribution is heterogeneous, and avascular areas are observed.	

Fig.1. Narrow Band Imaging (NBI) magnification findings [1].

comparison between a reference vector and an input feature vector. We introduce our proposed method that changes the distance computation from Euclidean distance to Manhattan distance to reduce the computation complexity and we discuss the tradeoffs between the accuracy and the time of the proposed method in Section 4. We also show the implementation of the feature transform with /without pipeline on FPGA to improve the feature transformation processing time on Section 5.

II. Identification Algorithm Based on the NBI Magnification

A. Narrow Band Imaging (NBI)

Narrow Band Imaging (NBI) is a special light endoscope technology newly developed in the field of digestive organ endoscope [1]. This technique emphasizes to blood vessels using the wavelength, which is easy to be absorbed to hemoglobin.

B. The Identification Algorithm Based on the NBI Magnification Findings

Fig.3 shows the identification algorithm of NBI magnification [2] used in our CAD system. The algorithm consists of a bag-of-features (BoF) representation of local features. In feature extraction, an input image is processed as Scan Window (SW). The local feature quantities are extracted at all key points, at which the feature extraction performed in the SW. We use the Dense Scale-Invariant Feature Transform (D-SIFT) that takes key points to dense. In the learning phase, the features obtained from the learning images of each type are clustered and the center of each cluster is saved as a Visual Word (VW). Next, the feature extracted from the input SW image is compared with the VWs of each type and a Visual Word histogram is made by voting for the nearest VW. In the testing phase, the system classifies the testing SW image by comparing the histogram of the learning phase with the histogram of the testing SW image, and displays the result. In this research, the testing image identifies 3 types of endoscopic image (Type A, Type B, and Type C3 in Fig.1). The hardware oriented feature extraction [3], which is modified from the D-SIFT of VLFeat Library [6] is used for the feature extraction and the Support Vector Machine (SVM) [4] which is modified from the LIBSVM Library [7] are applied for 3 types identification.

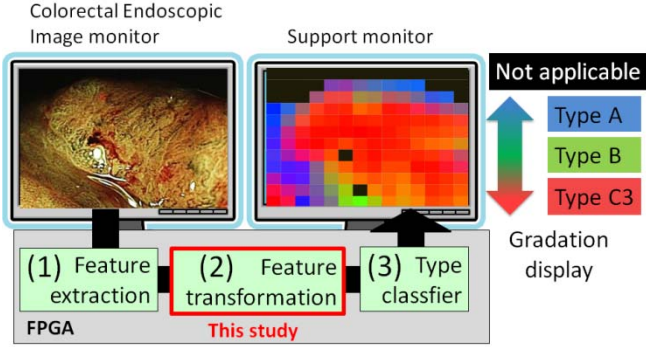


Fig.2. The proposed CAD system overview.

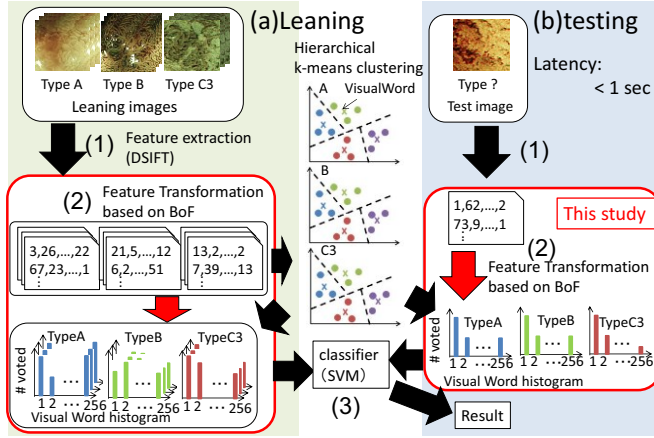


Fig.3. Bag-of-features (BoF) algorithm.

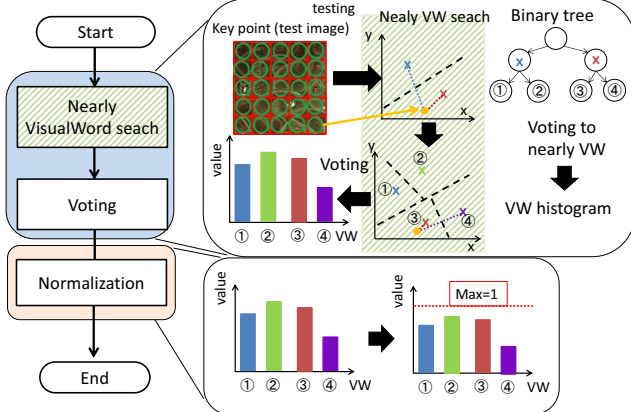


Fig.4. Feature transformation algorithm based on the Visual Word histogram.

III. Feature Transformation Algorithm

The feature transformation algorithm is used in learning (Fig.3(a)) and testing phases (Fig.3(b)). First, at the learning phase, the features of all the key points are extracted from a training image dataset and are used for the hierarchical k-means clustering method [8]. In our implementation, k is selected as 2^8 , and so, each clustering step will divide the feature set to two sub-sets. The hierarchical 2-means method repeats 8 times with the generated sub-set, making $2^8=256$ VWs creating 8-level binary search tree for each type. Then, three VW histograms are created by voting all the feature

vectors extracted from the testing image to the nearest VW of each type among A, B, and C3 as shown in Fig.3(a)(2).

Fig.4 explains the outline of the feature transformation algorithm. A D-SIFT feature vector of each key point that is extracted as shown in Fig.3(a)(1) is compared with the representative feature vector in each node of the binary tree to find the nearest VW. This process repeats 8 times, until the input feature vector reaches to the leaf node, which is the sub-cluster (or VW in other word) that that key point belong to. Hence, each input key point is voted on one of 256 VWs for each type. Since we have 3 types, all key points of the testing image will be voted on three 256-dimension VW histograms as shown in Fig.3(a)(2). Depend on the number of key points in the testing image, the VW histogram has different range of the maximum and the minimum. Hence, the histogram is normalized to the range $[0, 1]$ by equation (1). The VW of i^{th} dimension is defined to VW_i , the normalized VW'_i is as follows.

$$VW'_i = \frac{VW_i}{\sqrt{\sum_{i=1}^{256} VW_i^2}} \quad (1)$$

Then, the three 256-dimension histograms are combined together, making a 256-dimension \times 3 types = 768-dimension VW histogram. This histogram is then used in Fig.3(a)(3) for testing phase.

The same processing is done with the testing image by using the same VWs that are used in learning phase. As a result, we get a 768-dimension VW histogram of the testing image.

In the paper, we discuss the tradeoff of the distance computation between Euclidean distance and Manhattan distance in the distance comparison for feature vector classification. It helps to reduce both processing time and resources for VW histogram creation. In straight forward implementation, the computation of the denominator of equation (1) must wait until the VW histogram computation completes. We introduce a on the fly computation method for denominators of equation (1) to reduce the processing time in the hardware implementation. As a result, the waiting time for denominator computation in each dimension can be removed. We also discuss the architecture without pipeline or with pipeline.

IV. Simulation Experiments

A. Simulation method

We have performed several simulation experiments to verify the accuracy of the CAD system using Manhattan distance in feature vector classification and that using Euclidean distance computation. The number of VWs of each type is 256. The dataset consists of 1260 real endoscopic images, is used as the learning and testing data as shown in Fig.1 (image size are about $100 \times 300 \sim 900 \times 800$ pixel). They are created from the original images taken by Hiroshima University Hospital endoscope specialist by trimming the domain where the structure of each type is observed. The number of images of each type is also shown in Fig.5. In our proposed hardware oriented CAD system, the feature vector dimension is reduced from 128 to 64 as shown

	Type A	Type B	Type C3	All
Dataset NBI	420	420	420	1260

Fig.5. The dataset for Colorectal Endoscopic images of NBI Magnification findings.

on [3]. The SVM is also improved for resource saving by using 16-bit fixed point arithmetic [4]. The learning (a) and testing (b) phases in Fig.3 are considered with using Euclidean and Manhattan distances for distance comparison in near VW searching process, making 4 combinations as shown in Table 1. In the simulation, the accuracies of these 4 combinations are verified. The 10 fold-Cross validation is used for the simulation. In which, the data set is divided into ten sub-sets, and nine of them are selected as training images in training process. The remaining one sub-set is used as images for testing. The training and testing processes occurs for 10 times with different sub-sets. The final testing result is the average of the results of these 10 training-testing times. The true positive and precision rates are used for accuracy comparison among the 4 combinations. The true positive is an indicator of the accuracy of the identification, and the precision rate is a measure of deviation of the identification. They are defined in the following equations (2) and (3).

$$True\ Positive(i) = \frac{Posi_Num(i)}{Img_Num(i)} \times 100\ [%] \quad (2)$$

$$Precision\ Rate(i) = \frac{Posi_Num(i)}{Disc_Num(i)} \times 100\ [%] \quad (3)$$

i	Type A, Type B, Type C3
$Img_Num(i)$	Number of all sheets of the image data of i
$Disc_Num(i)$	Number of sheets of the image data discriminated from i
$Posi_Num(i)$	Number of sheets of the image data correctly discriminated from i

B. Result

The simulation results are shown in Fig.6. The columns in graphs (a) and (b) show the true positive and precision rate of the 4 combinations in Table 1. Form the results of Fig.6, the highest identification rate is obtained by EUEU, which means Euclidean distance is used in both learning and testing phases. The second one is EUMH that uses Euclidean distance in learning phase and Manhattan distance in testing phase. The difference of true positive in Type A is very small, and that of Type B and Type C3 are about 6% and 4%, respectively. The precision rate is decreased for Type B and Type C3 in EUMH. So the accuracy in identification of Types B and C3 is a little decreased in EUMH. However, our proposed Computer Aided Diagnosis (CAD) system can display the probability of the classification result by the gradation map in a support monitor as shown in Fig.2 [4]. Therefore, it is not a problem because we can show the estimates probability by the gradation map even if the identification of three types gives a probability as (0.1, 0.4, 0.5). So, we consider that the Manhattan distance in classifier is effective.

Table 1. Simulation combinations of distance metric.

		Leaning	
		Euclidian(EU)	Manhattan(MH)
Testing	Euclidian(EU)	EUEU	MHEU
	Manhattan(MH)	EUMH	MHMH

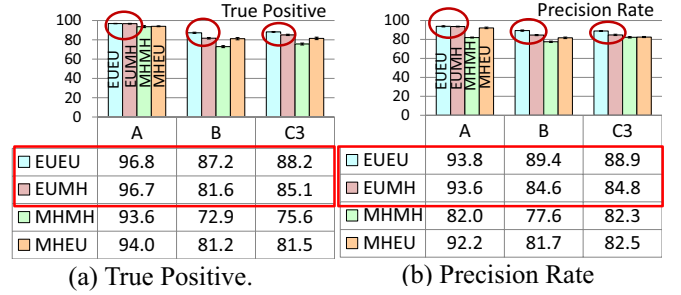


Fig.6. Simulation result for each learning and testing combination.

V. The Proposed Feature Transformation Architecture

A. Branch processing block

The branch processing block searches the nearest VW for the current inputted feature vector in the distance metric. The distance comparison procedure of the input feature vector and the 2 representative feature (left and right) vectors at each level of the binary search tree is shown in Fig.7. At each level, result of the distance comparisons of left and right feature vectors are used to determine the next searching node on the binary tree (Fig.8). If the $distance$ difference $d_j \leq 0$, then the left child node is selected for the next searching, otherwise $d_j \geq 0$, the right child node is selected. The searching result at each level is represented by one bit for left or right. The combination of those results from 8 levels gives a 8-bit result, represents the VW number closed with the input feature vector among the 256 VWs given at the learning phase. Fig.9(a) shows the block design of the distance comparison block.

Let f_j be the input feature vectors and l_j and r_j are defined as the representative feature vectors for left and right child nodes, respectively. Then the Euclidean distance is defined as $distance_{EU}(l_j, f_j) = (l_j - f_j)^2$ and the Manhattan distance is calculated $distance_{MH}(l_j, f_j) =$

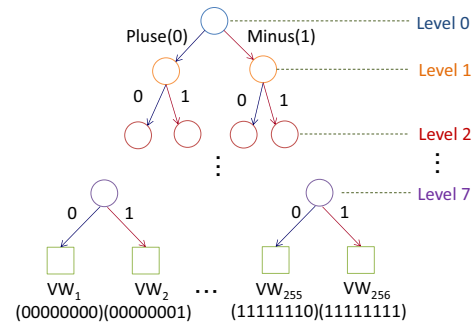


Fig.7. The nearest VW search in the binary tree.

$$distance(l_j, f_j) = (l_j - f_j)^2 \text{ or } |l_j - f_j|$$

$$\Delta VW_{left} = distance(l_j, f_j)$$

$$\Delta VW_{right} = distance(r_j, f_j)$$

$$d_j = \Delta VW_{right} - \Delta VW_{left}$$

$$d_j < 0 \quad \text{right} \quad d_j \geq 0 \quad \text{left}$$

f_j : input feature vector of j^{th}
 l_j : representative vector of left cluster of j^{th}
 r_j : representative vector of right cluster of j^{th}

Fig.8. Calculation at each level of binary tree.

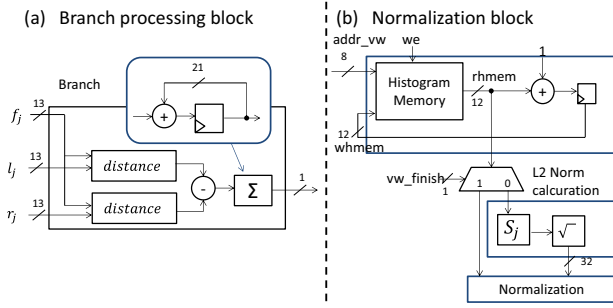


Fig.9. Processing blocks. (a) Branch processing block
(b) Normalization block

$|l_j - f_j|$, respectively.

B. On the fly normalization computation

The Normalization block computes the L2 Norm ($\sum_{i=1}^{256} VW_i^2$) in the denominator of equation (1). The conventional method requires the normalization process waits until the calculation of this L2 Norm is finished. We improve the Normalization block to be able to calculate the L2 Norm during VW histogram creation by using partial sum.

Let M be the total number of feature vectors which will be voting in current input image.

Let $N_j(VW_i)$ be the number of voting for the feature vector VW_i ($1 \leq i \leq 256$) at the j^{th} voting. We define the partial sum of the L2 Norm for all VW_i at the j^{th} voting as follows.

$$S_j = \sum_{i=1}^{256} N_j(VW_i)^2$$

When the $j + 1^{th}$ voting is performed and the f_{j+1} is voted the Visual Word VW_k the partial sum of the L2 Norm for all VW_i at the $j + 1^{th}$ voting can be calculated as following manner.

$$\begin{aligned} S_{j+1} &= \sum_{i=1}^{256} N_{j+1}(VW_i)^2 \\ &= \sum_{i=1, i \neq k}^{256} N_j(VW_i)^2 + (N_j(VW_k) + 1)^2 \\ &= \sum_{i=1}^{256} N_j(VW_i)^2 + (2N_j(VW_k) + 1) \\ &= S_j + 2N_j(VW_k) + 1 \end{aligned} \quad (4)$$

From the equation (4), S_{j+1} can be calculated the partial sum S_j and the previous number of voting of $N_j(VW_k)$.

Because $N_0(VW_i) = 0$ ($1 \leq i \leq 256$), we can easily obtain $S_M = \sum_{i=1}^{256} N_M(VW_i)^2$ by the above recurrent relation. So the partial sum can be calculated with the addition and shift operation on the fly of the voting. This is very suitable for hardware implementation without waiting time. The improved architecture is shown in

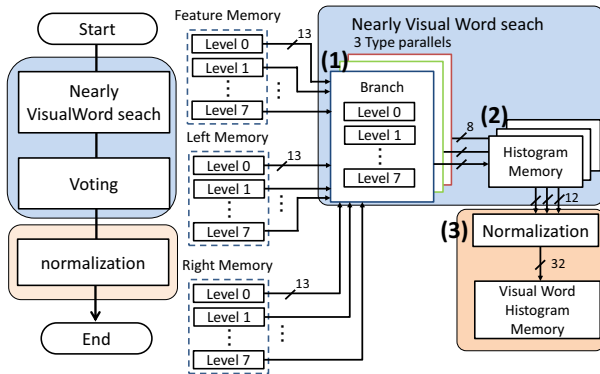


Fig.10. The block diagram of the proposed architecture.

Fig.9(b). The previous voted number $N_j(VW_i)$ of the current Visual Word VW_i for f_j is read from the VW histogram memory and $N_j(VW_i) + 1$ is stored to the memory. At the same time, $2N_j(VW_i) = N_j(VW_i) \ll 1$ is calculated by the left shift operation. In this way, we can obtain S_{j+1} by the previous partial sum on the fly. On the fly L2 Norm computation reduces 256 clocks for each SW in the image.

C. Feature transformation Architecture

The block diagram of the proposed high speed feature transformation architecture, which includes the above Branch processing and Normalization blocks are shown in Fig.10. First, the input feature vector is transformed to the VW histogram by searching for the near VW and voting to corresponding element in the Branch processing block (a). Next, the VW histogram is normalized in the Normalization block (b), before sending to the SVM classifier.

VI. Discussion on FPGA Implementation

We have implemented two distance metric (Euclidian and Manhattan) on FPGA, Altera Stratix IV (EP4SE530H35C2) device. In these implementations, the simple architecture without pipeline in branch processing and on-the-fly L2 Norm calculation is used. The three parallel visual transformations are implemented for each three sub-visual word, related with types A, B and C3. Their occupied resources and processing time are shown in Table. 2 for comparison. Estimation for pipeline implementation is also included. The DSP (Digital Signal Processing) in the FPGA is the dedicated block used to calculate the multiplication in high speed. The type classifier (SVM), which received feature transformation result in Fig.3, can increase its processing speed by using multiple DSP block in parallel. Hence, reducing the number of DSP blocks used in feature transformation leaves more DSP resources for the critical SVM module. Manhattan distance implementation saves 12 DSP blocks compared with Euclidian distance, and so the number of SVM can be increased. The Manhattan distance implementation can be speed up the processing frequency up to 113 MHz compared with the 84 MHz of Euclidian implementation. Table 3 shows the number of key points extracted from the Full HD (1920 x 1080) image. The image is raster scanned at 60-pixel interval for each Scan Window (SW) size (60, 120, 180, and 240), which is corresponding to the input image in the above discussion. The raster scanned area has overlap as shown in Fig.11. We reuse the voting result of overlapped areas, and so reduce the number of key points that need to be voted. Table 4 shows the processing time of feature transformation and the number of key point voted in this algorithm. Without reusing the overlap, number of key points that must be voted for SW 240x240-pixel size is over one million. The voting process will take over one second to complete. This time reduces to 381 msec for feature transformation using Manhattan distance and 513 msec for that using Euclidean distance if voting result of

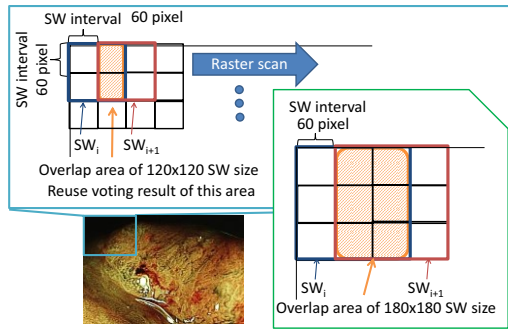
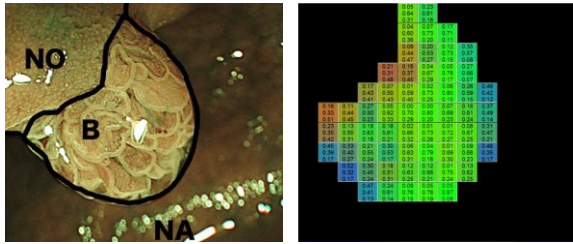


Fig.11. Reuse area of overlap.



(a) Input endoscopic image (b) Gradation map
 NO : Normal, NA : Not applicable
 Type A Type B Type C3 defocus region, white area

Fig.12. An example of a gradation map for real endoscopic image [9].

overlapped area is reused. It confirmed that real-time processing for each SW size is possible.

The improved normalization design with dynamic L2 Norm computation reduces an amount of 256 x the-number-of-SW (show in Table 3) clocks in processing time, equivalent to 147 thousands clocks in Full HD image scanning with SW size 60. It also reduces the required DSP blocks. The pipeline implementation architecture in Fig.10 would be about eight times faster than that shown in Table 4 but the number of ALUTs, register and DSP blocks occupied by branching module in Fig.10(1) also increases by 8 times. It makes total number of ALUTs, registers and DSPs in EU version increases from 3,163 (LUTs), 3,328 (registers), and 52 (DSPs) to 7,195 (ALUTs), 4,840 (registers), and 136 (DSPs) respectively. These increases in MH implementation are 4,032 (ALUTs), 1,512 (registers), and 0 (DSP). No increase in number of DSPs occurs due to the fact that no DSP is required for MH distance computation. The processing time decrease to 64 msec in Euclid distance, and 47 msec in Manhattan distance versions (Table 5). As a result, it is possible to improve the latency and the throughput by pipeline. This processing time is also 150 times faster than software implementation in Section 1.

VII. High-Speed Feature Transformation for Region Division

Fig.12 shows a reality image in diagnosis. It contains areas for type B, normal and not applicable. Their boundary can be in any shape, and so hard to detect by a fixed SW size. We propose a method that uses some SW sizes for the identification of pathological types with complex boundary. It

Table 2. Resource to be used.

	Non pipeline		Pipelined (estimate)	
	EU	MH	EU	MH
Stratix IV				
ALUT	3,163	3,163	7,195	7,195
Register	3,328	3,328	4,840	4,840
Total RAM bits	959,488	959,488	959,488	959,488
DSP block	52	40	136	40

Table 3. The number of key points for different SW sizes.

SW size	# of key points in SW(=M)	# of SW in Full HD image	Total # of key points
60x60	145	576	83,520
120x120	841	527	443,207
180x180	1,604	480	769,920
240x240	3,004	435	1,306,740

Table 4. Processing time without pipelining.

SW size	# of voting key points	latency (msec)		throughput (fps)@100MHz	
		EU	MH	EU	MH
60x60	83,520	509	378	1.96	2.65
120x120	83,781	510	379		2.64
180x180	83,819	511	379	1.95	2.62
240x240	84,204	513	381		

Table 5. Processing time with pipelining (estimate).

SW size	# of voting key points	latency (msec)		throughput (fps)@100MHz	
		EU	MH	EU	MH
60x60	83,520	63	47	15.87	21.27
120x120	83,781	63	47		
180x180	83,819	63	47	15.62	
240x240	84,204	64	47		

detects the pathological type and the boundary in each SW size and combines the results together using different weight for each size (Fig.13). If four SW sizes in Table 4 are sequentially raster scan the Full HD image, the non-pipeline architecture requires about 1,520 msec, while in Table 5 the pipeline architecture requires about 190 msec. Hence, the pipeline implementation meets the target of real time processing.

In particular, increasing SW sizes, the number of key points inside the SW is also increased. Therefore, the processing time of feature transformation is increased as the result of the increase in the number of key points that need to be processed using the binary searching tree. We are investigating a method that improves the processing time of VW histogram transformation for multi SW sizes. The proposed method creates VW histogram for bigger area by combining the VW histogram that is already created for smaller area, so the voting time can be reduced [10]. Fig.14 shows the conceptual diagram of the method. The feature extraction for all SW size in this study algorithm takes the key points at 5 pixel interval, so the extracted feature vector for key points in the small SW size is similar to that in bigger size at the same area. Hence, the VW histogram of big SW

that is made by combining the VW histograms of all small SWs inside is same with the VW histogram made by common feature transformation process for that big SW. As a result, it is possible to perform the process of VW histogram making for smallest SW size (reference SW size) only, then combines the VW histogram results in corresponding area to create VW histogram of other bigger size. Example for this computation method can be seen in Fig.14, where VW histogram of a big SW size (120x120 pixels) can be computed by adding VW histogram of the related four smaller references at size 60x60 pixels. Reference SW size can also make smaller or bigger, if overlap occur. Since the VW histogram creation occurs once only with reference SW size, the processing time of this method for VW histogram creation for multiple SW sizes is similar to that shown in Table 4 for smallest SW size (reference SW size = 60 pixel in Table 4). In other words, this method can process all SW sizes without the parallel computation for each one. This method decreases the number of SWs for identifying Full HD image. Therefore, the hardware architecture which uses this algorithm is 15 times smaller in key points than software implementation (in case of SW size 240x240 is used).

VIII. Summary

This paper shows that, the design using Manhattan distance for VW histogram computation achieves almost the same accuracy of that using Euclid distance. Usage of Manhattan distance architecture improves the frequency about 35% and reduces 12 DSP blocks. The processing time is improved about 130 msec. The improvement in normalization module reduces about 147000 clocks in the processing time NBI system. It is more effective to the algorithm that has more VW dimensions. We also introduced a VW histogram combination method for high speed feature transformation

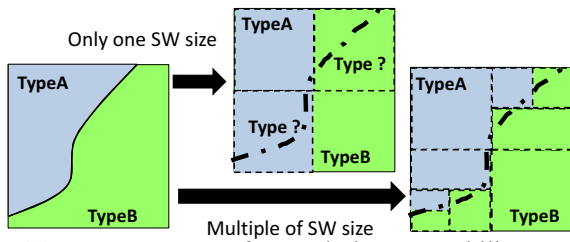


Fig.13.Improvement of resolution capability with multiple SW decomposition.

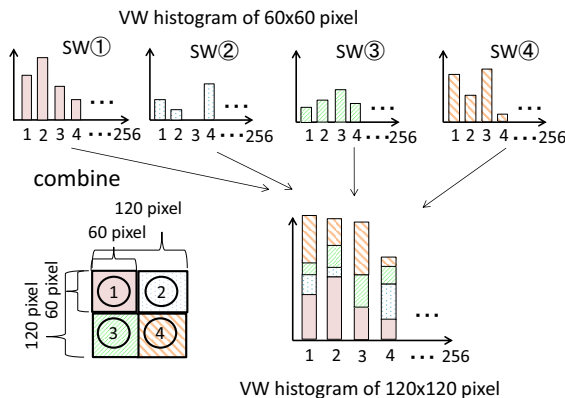


Fig.14. An implementation of the multiple SW decomposition with histogram combination.

for multiple SW sizes, in which the processing time is closed to that of the basic SW size with nearly no HW penalty. Pipeline implementation for nearest VW searching reduces latency of feature transformation to 64 msec for real time processing.

Acknowledgements

Part of this work was supported by Grant-in-Aid for Scientific Research (C) and Research (B) JSPS KAKENHI Grant Numbers 2459102 and 26280015, respectively.

References

- [1] T. Tamaki, et al., "Computer-aided colorectal tumor classification in NBI endoscopy using local features", medical image analysis, Vol. 17, No. 1, pp. 78-100, 2013.
- [2] H. Kanao, et al., "Narrow-band imaging magnification predicts the histology and invasion depth of colorectal tumors," Journal of Gastrointestinal Endoscopy, vol. 69, no.3, pp. 631-636, 2009.
- [3] T. Mishima, et al., "FPGA implementation of feature extraction for colorectal endoscopic images with NBI magnification," Proc. of the IEEE International Symposium on Circuits and Systems (ISCAS2014), pp.2515-2518, June 1-5 2014.
- [4] S. Shigemi, et al., "An FPGA implementation of support vector machine identifier for colorectal endoscopic images with NBI magnification," Proc. of the 28th International Conference on Circuits / Systems, Computers and Communications (ITC- CSCC2013), pp. 571-572, 2013.
- [5] T. Koide, et al, "FPGA Implementation of Type Identifier for Colorectal Endoscopic Images with NBIMagnification," Proceedings of the 12th IEEE Asia Pacific Conference on Circuits and Systems (APCCAS 2014), pp. 651-654, 17 - 20, Nov 2014.
- [6] A. Vedaldi, and B. Fulkerson, "Vlfeat: an open and portable library of computer vision algorithms," <http://www.vlfeat.org/>
- [7] Chin-Chung Chang, Chin-Jen Lin, "Livsvm – a library for support vector machins," <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [8] J. Nister, and H. Stewenius, "Scalable recongnition with a vocabulary tree", Proc. of the IEEE Computer Vision and Pattern Recognition (CVPR2006), pp. 775-781, 2006.
- [9] T. Okamoto, et al, "A Hierarchical Type Segmentation Algorithm based on Support Vector Machine for Colorectal Endoscopic Images with NBI Magnification," Proc. of Synthesis And System Integration of Mixed Information (SASIMI2015), in press.
- [10] K. Sugi, et al, "Consideration for Acceleration of Feature Transformation based on the Bag-of-Features for Colorectal Endoscopic Images," IEICE Tech. Rep., vol. 114, no. 302, CPSY2014-55, pp. 7-12, Nov. 2014.