High Speed Approximation Feature Extraction in CAD System for Colorectal Endoscopic Images with NBI Magnification

Tsubasa Mishima 1), Satoshi Shigemi 1), Anh-Tuan Hoang 1), Tetsushi Koide 1),
Toru Tamaki 2), Bisser Raytchev 2), Kazufumi Kaneda 2),
Yoko Kominami 3), Rie Miyaki 3), Taiji Matsuo 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
email: { mishima-tsubasa, shigemi-satoshi, anhtuan, koide } @ hiroshima-u.ac.jp

Abstract - In this study, we have proposed an improvement for feature extraction in computer-aided diagnosis (CAD) system for colorectal endoscopic images with narrow-band imaging (NBI) magnification [1]. Dense Scale-Invariant Feature Transform (D-SIFT) is used in the feature extraction [3]. It is necessary to consider a trade-off between the precision of the feature extraction and speedup by the FPGA implementation for processing of real time full high definition image. In this paper, we reduced the number of dimensions for feature representation in hardware implementation purpose.

I. Introduction

Colorectal cancer has been ranked as the third highest cause of cancer death following lung cancer and stomach cancer in Japan. Early detection of colorectal cancer is very important because if it is recognized at the first stage, it can usually be completely eliminated. However, there is a problem that the number of doctors who can diagnose colorectal endoscopy cancer is limited because it needs high specialty. With the increases in the number of colorectal cancer patients, systems which support a doctor's diagnosis have been researched currently. The computer-aided diagnosis system for colorectal endoscopic images with narrow band imaging (NBI) magnification [1] has already been proposed [2]. It identifies 3 types of endoscopic (Type A, Type B, Type C3) in Fig. 1 and Fig. 2. Those 3 types have relationship with normal mucosa, an adenoma, and advanced cancer, respectively. The implementation in software achieves a speed of 14.7 fps for a scan window (SW) of 120 x 120 pixels. In order to deal with the case in which two or more types are intermingled in one scanning window, multiple patches recognition is required. Hence, the processing time per frame when scanning in 10-pixel step is very long, which is (processing time / 1 SW) x (# of SWs) = (1 / 14.7 fps) x (17,557) ≈ 20 min. This study aims to an optimized algorithm, which is suitable for implementation on FPGA for a high performance computer-aided diagnosis system. In this paper, we introduce an improved algorithm of the brightness incline calculation method in a feature extraction for hardware implementation.

Section II explains the outline of the CAD system and Section III describes the improvement for hardware implementation of the feature extraction algorithm. In Section IV, the simulation results of the proposed algorithm are compared with the original implementation. The feature extraction architecture is proposed in Section V. Then, Section VI is the summary and conclusions.

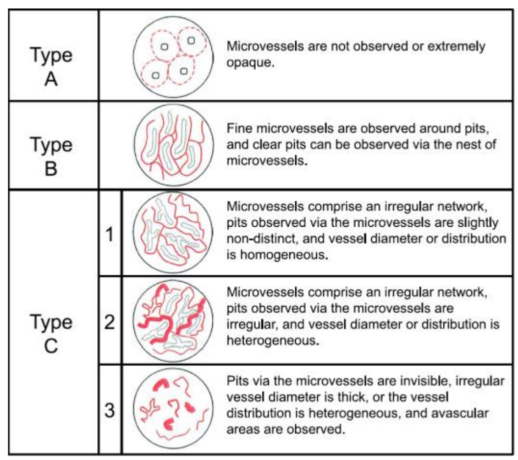
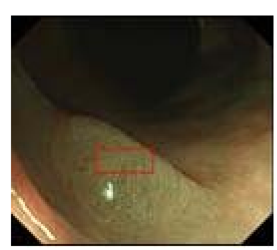


Fig. 1. Narrow Band Imaging (NBI) magnification findings [1].

(a) Type A. (b) Type B. (c) Type C3.



(d) Colorectal Endoscopic Image which includes Type A .

Fig. 2. Example of Colorectal Endoscopic NBI Images.

II. Software Implementation of Computer-Aided Diagnosis System

A. Narrow Band Imaging (NBI)

Narrow Band Imaging (NBI) is a special light endoscope technology newly developed in the field of digestive organ endoscope. This technique emphasized to blood vessels using the wavelength, which is easy to be absorbed to hemoglobin [2].

B. The Identification Algorithm Based on the NBI Magnification Findings

Summary of the algorithm is shown in Fig. 3. The algorithm consists of a bag-of-features (BoF) representation of local features. BoF applies a document search to an image and considers a feature vector achieved by extracting the feature of an image and distinguishes it by the appearance frequency. First, the features obtained from the images of each type are clustered and the center of each cluster is saved as Visual-Word (VW). Next, the features extracted from the input image are compared with the VWs of each type and a histogram is made from voting for the nearest VW. System classifies input image by comparing the histogram made from the images of each type with the histogram made from the input image, and displays a result. In this research, the Dense Scale-Invariant Feature Transform (D-SIFT) of Library VLFeat [3] is used for the feature extraction and Support Vector Machine (SVM) of Library LIBSVM [4] is used for type identification.

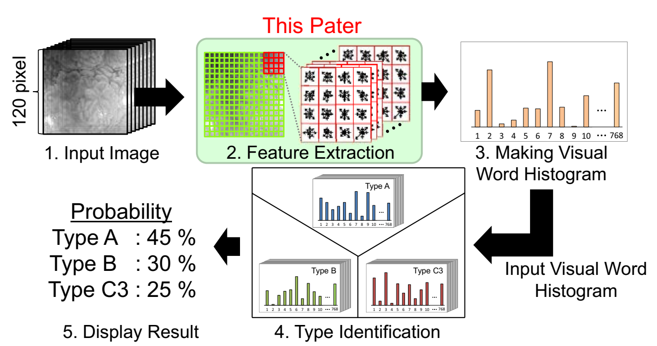


Fig. 3. Computer-aided diagnosis system for colorectal endoscopic images with narrow-band imaging (NBI) magnification.

C. Feature Extraction Algorithm

The D-SIFT is a technique that divides and computes the features of a picture in 128 dimensions from the gradient of luminosity value. By taking a lot of features, an image with little color change such as a large intestine endoscopic image can be identified. It handles with the change in scale of the image by calculating feature with multiple window sizes (scale). Figure 4 shows the original D-SIFT algorithm. At first input image is smoothed by using Gaussian Filter (GF) and the direction of a gradient and the intensity are calculated using the gradient of the luminosity value to x-direction and the gradient of the luminosity value to y-direction for each pixel. Next, the intensity of gradient of the pixels with the same direction within a certain area is convoluted and these values are weighting according to the distance from the central point to each block. Feature vectors of 128 dimensions are provided by normalizing those values. Step of feature point is 5 pixels. Two SWs of 20 and 28 pixels are used. The block sizes, which are the size of one block inside a SW in Fig. 4 are 5 and 7 pixels in this system.

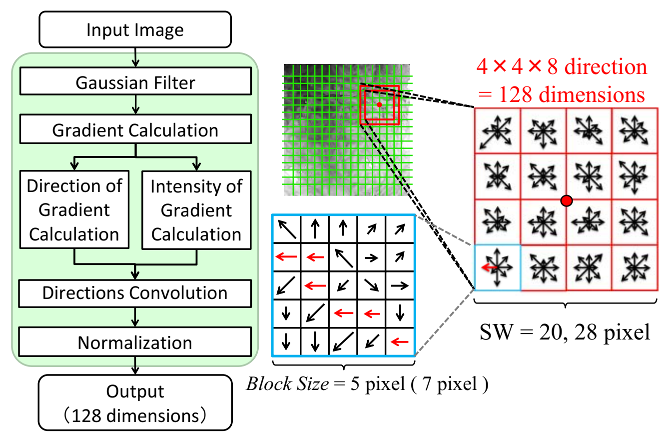


Fig. 4. The Original D-SIFT Algorithm.

III. High Speed Approximation for D-SIFT Algorithm for Hardware Implementation

A. Improvement of the Gaussian Filter Processing

The original GF computation is shown in equation (1). The speedup of Gaussian Filter (GF), in DSIFT is very important because it is applied to each pixel of Full HD image (1,920 x 1,080). There are some calculation problems in the original algorithm in equation (1), that is many floating point multipliers are needed, resulting to big processing time and big hardware size in straight forward GF implementation. Therefore we performed three following modification to solve these problems [5].

A.1. Unification of the GF Coefficient Independent with the Scale for Multiple Patches Recognition

Generally, GF coefficient varies with the value of the scale. GF calculation for each pixel is shown in equations (1), (2), and (3) with parameter shown in Table 1. The value of parameter σ and SW of the GF depends on the *Block Size*, in which, when the *Block Size* = 5 pixel, *SW* = 9 x 9 pixels and σ = 5/6, and when the *Block Size* = 7 pixel, the *SW* = 11 x 11 pixel and σ = 7/6. Two image smoothing processes are required for one input image. In our modification, we set σ and *SW* to a constant, making them independent with the *Block Size*. The required hardware size becomes about a half while the throughput keep the same.

$$GF(x,y) = exp(-\frac{x^2 + y^2}{2\sigma^2}) \quad (1)$$

$$Dst(x,y) = \sum_{i=x-a}^{x+a} \sum_{j=y-a}^{y+a} Img(i,j) \times GF(i,j) \quad (2)$$

$$a = \frac{SW - 1}{2} \quad (3)$$

Table 1. Parameter of Gaussian kernel.

x, y	Coordinate $(-a \le x, y \le a)$
σ	Parameter of Gauss kernel
GF (x, y)	Gaussian Filter coefficient at (*x, y*)
Img (i, j)	Luminosity value of input image at (*i, j*)
Dst (x, y)	Smoothed image value at (*x, y*)
SW	Scan window size of Gaussian Filter

A.2. Simplify GF by Power of Two Coefficients

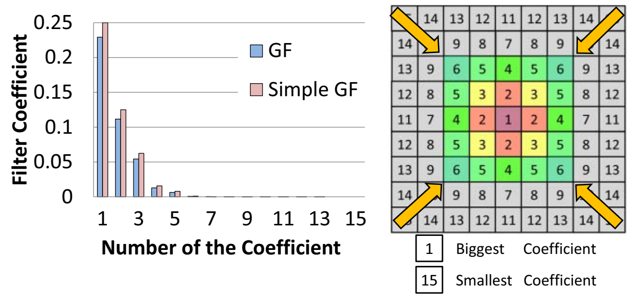
In the software implementation, GF processing is calculated using floating point coefficient. A direct hardware implementation faces with increases in processing time and circuit areas due to floating point multiplier. So, we modified those coefficients to power of 2 number to solve this problem. Hence, the floating point multipliers can be replaced by constant shifters. It significant shortens processing time and reduces area of hardware.

A.3. Reduction of Scan Window Size

Figure 5 shows the index of the coefficient of each pixel in the 9 x 9 GF matrix. Smaller index means bigger coefficient. The pixel at the center of Fig. 5 (b) has the biggest coefficient and indexed by 1, while those at the four corners, which are indexed by 15, have the smallest coefficient. There is a fact that the sum of coefficients with indexes bigger than 6 occupies only 0.3 % the overall. Hence, we can reduce the scan window size from 9 x 9 to 5 x 5 with very small penalty in accuracy. SW size can be reduced approximately 70 % by omitting the pixel with coefficient indexed bigger than 6. In this case, σ is fixed to 5/6. Coefficients of original GF and proposed Simple GF are shown in Table. 2.

B. Improvement of the Gradient Direction Calculation Process

Feature is computed based on the direction of a gradient and the intensity of luminosity. The gradient of the luminosity value to x-direction is defined as *Gx*, and the



(a) GF Coefficient graph (*Block Size* = 5 pixel). (b) Reduction of the SW size of GF.

Fig. 5. A coefficient number and graph of GF.

Table 2. Coefficient of GF and Simple GF.

# of the coefficient	1	2	3
GF	0.23	0.112	0.0543
Simple GF	$0.25 = 2^{-2}$	$0.125 = 2^{-3}$	$0.0625 = 2^{-4}$
# of the coefficient	**4**	**5**	**6**
GF	0.012865	0.0062621	0.0007221784
Simple GF	$0.015625 = 2^{-6}$	$0.0078125 = 2^{-7}$	$0.0009765625 = 2^{-10}$

gradient of the luminosity value to y-direction is defined as *Gy*. As shown in Fig. 6, the direction of a gradient and the intensity are calculated using *Gx* and *Gy* for each pixel. The most accurate method (original implementation) calculates the $Tan^{-1}(Gy/Gx)$ angle and the gradient intensity of luminosity before assigning them to the 8 directions as shown in Fig. 7. However, this calculation method is very complicated, and is not suitable for hardware implementation. In our implementation, the gradient of the luminosity value of the pixel are classified into 4 directions according to the sign of *Gx* and *Gy*, and also divided into two by comparison absolute values of *Gx* and *Gy* as shown in Fig. 8, equation (4), (5), and Table. 3, respectively. By that, the pixels are classified in 8 directions based on their signs and absolute values in x and y directions. The number of dimensions relies on the number of directions that a gradient is divided. We divide the gradient into 8, 4 and 2 directions for 3 implementations. The accuracy among them is then compared for the best implementation selection.

$$Tan(0) = 0 < \frac{|Gy|}{|Gx|} < Tan(\frac{\pi}{4}) = 1 \quad (4)$$

$$\Leftrightarrow \quad |Gy| < |Gx| \quad (5)$$

Table 3. Parameter of the Gradient direction calculation.

I (i, j)	Luminosity value at (*i, j*)
Gx	The gradient of the luminosity value to x-direction
Gy	The gradient of the luminosity value to y-direction
Dir	Value to express an gradient direction

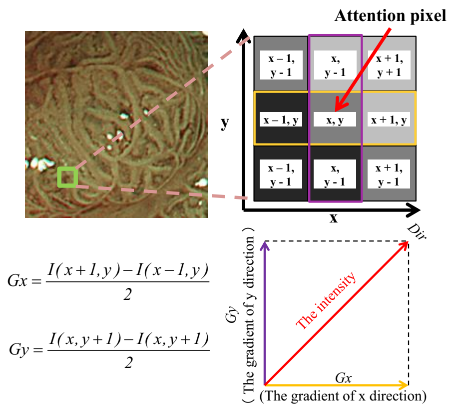


Fig. 6. Brightness gradient calculation method.

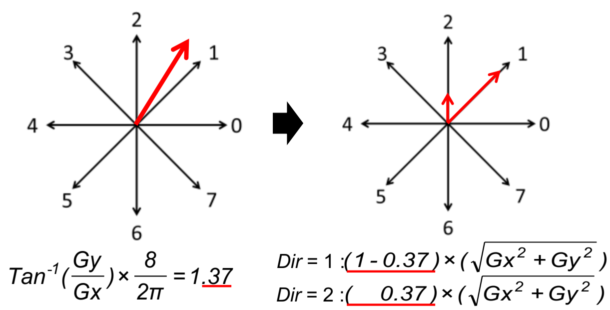


Fig. 7. Gradient direction calculation method using the arctangent.

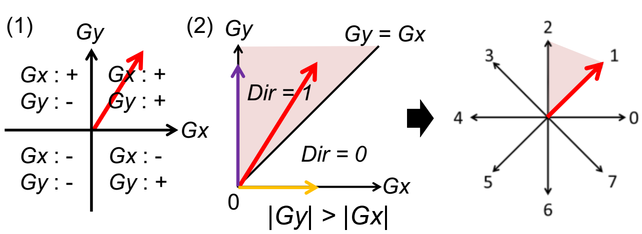
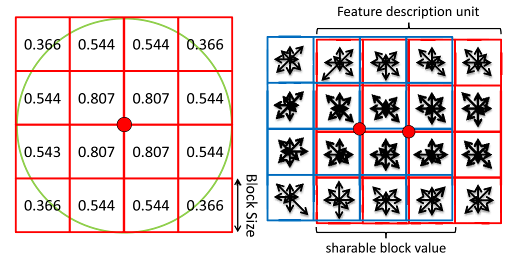


Fig. 8. Gradient direction calculation method by the suggestion technique.

C. Simplification of the Weight for Convolution Process and Convolution Features Sharing

The gradient intensity of each computed direction is convolved. This convolution occurs with all blocks in each feature description unit. The coefficient for convolution process relies on the distance from the central point to the corresponding block as shown in Fig. 9 (a). A feature description unit is generated by multiply the weight factor in Fig. 9 (a) with the corresponding 4 x 4 blocks. Hence, the feature of each block in a feature description unit is different from that in other feature description unit. By omitting this process, feature of all the blocks are similar regardless the feature description unit they belong to. Hence, those feature value are sharable among different units as shown in Fig. 9 (b).



(a) Weighting factor for blocks inside feature description unit. (b) Sharing of the block value.

Fig. 9. Improvement of the weighting process.

D. Normalization Replacement by Threshold Processing

The convolved block values are normalized in a unit of 16 blocks. This aims at obtaining the same features regardless of changing brightness, as shown in Fig. 10. However, normalization process needs multiplication, division, and square route computation with many inputs, as shown in equation (6) and Table 4. Moreover, the normalization will equalize the peak of the histogram of each type (Fig 11). In our implementation, the difference of the luminosity value was controlled by performing threshold comparison as shown in equation (7) and Table 4 at the output value of Gaussian Filter processing (Fig. 12). We omit the normalization process to reduce hardware size as well as shorten the computation critical path.

$$Dst_n = \frac{Tmp_n}{\sqrt{Tmp_1^2 + Tmp_2^2 + \cdots + Tmp_{128}^2}} \quad (6)$$

$$Img''(x, y) = \min\{Img'(x, y) >> 10, 255\} \quad (7)$$

Table. 4. The parameters of normalization.

n	The number of dimensions
Tmp_n	The value of dimension n before normalization
Dst_n	The value of dimension n after normalization
Img''	The value which smoothed the input image
Img'	The value which performed threshold processing

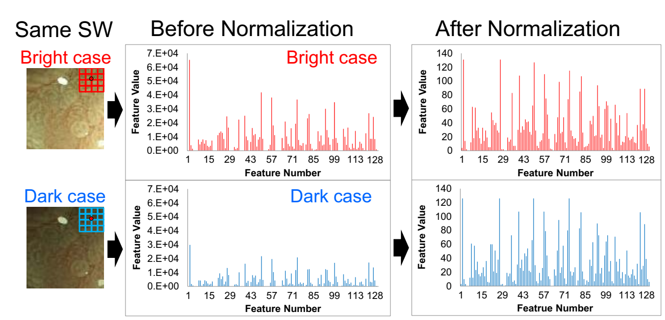


Fig. 10. Aims of normalization.

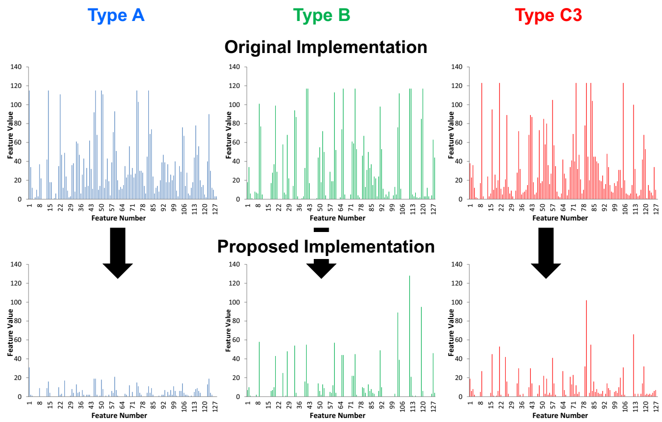


Fig. 11. The example of the feature histogram of each type.

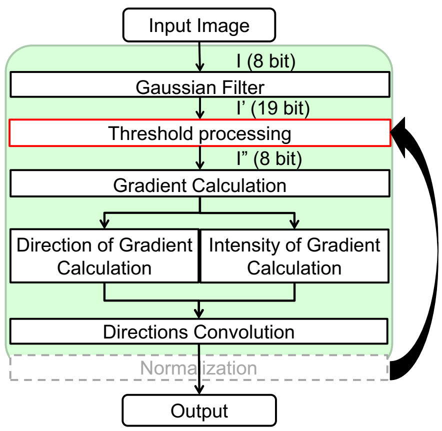


Fig. 12. Proposed threshold processing for normalization replacement.

IV. Simulation Result and Verification

A. Simulation Method

The proposed hardware oriented algorithm, which set the number of dimension to 128, 64, 32, and 16 by setting the direction of a gradient to 8, 4, 2, and 1, are compared with the original one by software simulation. The number of VWs is 768. The kernel 'easy_linear_probability' is used for SVM [4]. The Simple Gaussian Filter (SGF) proposed in Ref. [5, 6], which aims to hardware implementation is used for smoothing processing in D-SIFT. The parameters used in D-SIFT are shown in Table 5. The dataset is 1260 large intestine NBI expansion endoscopic images (about 100 x 300 ~ 900 x 800 pixel). They are created from the original images taken by Hiroshima University Hospital endoscope specialty by trimming the domain where the structure of each type is observed. The number of images of each type is shown in Table 6.

B. Evaluation Method

Equations (8) and (9) are used for accuracy evaluation. Parameters of accuracy evaluation are shown in Table. 7. True positive is used to determine whether identification of each type is carried out precisely and Precision rate is used to determine whether the distinguished type have deflection. The 10 fold-Cross validation method is used for the simulation. The data set is divided into ten sub-sets. 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 process occurs for 10 times with different sub-sets. Final testing result is the average of the results of these 10 training –testing times.

Table. 5. The parameters of D-SIFT [3].

Name	Value	Explain
Sizes	5, 7	Scales at which the dense SIFT features are extracted.
Fast	TRUE	Set to false to turn off the fast SIFT features computation.
Step	5	Step (in pixels) of the grid at which the dense SIFT features are extracted.
Contrast Threshold	0	Contrast threshold below which SIFT features are mapped to zero. The input image is scaled to have intensity range in [0,1] (rather than [0,255]) and this value is compared to the descriptor norm as returned.
Window Size	1.5	Size of the Gaussian window in units of spatial bins.
Magnif	6	The image is smoothed by a Gaussian kernel of standard deviation SIZE / MAGNIF.

Table. 6. NBI expansion endoscopic test images.

	Type A	Type B	Type C3
Number of Images	420	420	420

$$True\ Positive\,(i) = \frac{Posi\_Num(i)}{Img\_Num(i)} \times 100\ [\%] \qquad (8)$$

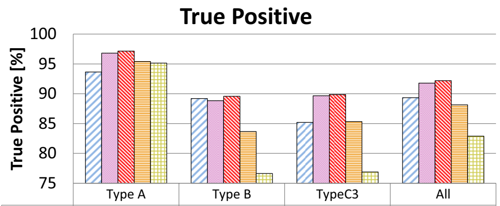
$$Precision\ Rate\,(i) = \frac{Posi\_Num(i)}{Disc\_Num(i)} \times 100\ [\%] \qquad (9)$$

Table. 7. Parameters of accuracy evaluation.

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

C. Simulation Result

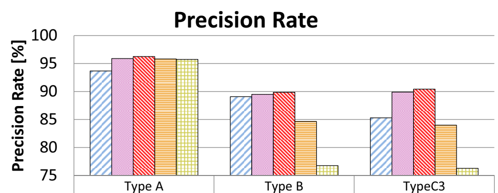
The simulation results in Fig. 13 (a) and (b) shows that the accuracy of the 64-dimensional algorithm is the best. In addition, the recognition accuracy of Type B and Type C3 decrease if the number of dimensions gets over 64. The 32-dimensional algorithm gets a worse result compares with the software implementation. The simulation results show that the 64-dimensional algorithm is the best for hardware implementation and be able to apply to high performance diagnosis system.



	Type A	Type B	TypeC3	All
Original	93.64	89.19	85.19	89.34
Pro (128)	96.81	88.83	89.67	91.77
Pro (64)	97.14	89.57	89.86	92.19
Pro (32)	95.40	83.67	85.33	88.13
Pro (16)	95.14	76.62	76.88	82.88

* Pro : Proposal, (#) : #-dimensions

(a) True Positive.



	Type A	Type B	TypeC3
Original	93.67	89.06	85.29
Pro (128)	95.87	89.49	89.88
Pro (64)	96.25	89.85	90.42
Pro (32)	95.82	84.65	83.99
Pro (16)	95.74	76.75	76.28

* Pro : Proposal, (#) : #-dimensions

(b) Precision Rate.

Fig. 13. Comparison results among various dimension D-SIFT algorithms and the original algorithm.

V. D-SIFT Hardware Architecture Overview

The hardware architecture of the proposal algorithm is shown in Fig. 14. In the proposed architecture, pipeline processing is realized by using FIFO. Moreover, by reducing the number of directions of the gradient from 8 to 4, the amount of memories is reduced by about 20%. In addition, pipeline processing is realized by performing block line gradient computation. Each pixel will get to the system for block line gradient computation before storing into block buffer. When the pixel in the next line of the same block comes, the corresponding intermediate block line gradient is read from the block buffer to continue the gradient computation for that block. As mentioned in Section II. C, two block sizes of 5 pixel and 7 pixel are necessary in our system. Hence, the Direction and Intensity of Gradient Calculation unit (c) and the Directions Convolution unit (d) in Fig.14 are duplicated for *Block Size* = 5 pixel and *Block Size* = 7 pixel processing in parallel.

VI. Summary and Conclusions

In this paper, the optimal number of dimension for the hardware implementation of the features extraction for a NBI intestine endoscopic image diagnostic support system was investigated. The simulation results show that the most excellent accuracy is achieved in the 64-dimensional algorithm. So we proposed the D-SIFT architecture based on this result. The proposed 64-dimension D-SIFT implementation reduces the amount of memories about 20 % as compared with conventional 128 dimensions implementation in software. The future work includes a hardware implementation of the proposed algorithm and an improvement in the processing speed for real time full HD image processing.

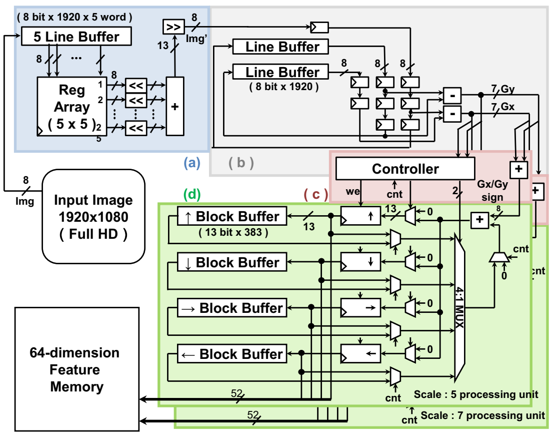


Fig. 14. The proposed D-SIFT architecture.
(a) Simple GF Processing unit. (b) Gradient Calculation unit.
(c) Direction and Intensity of Gradient Calculation unit.
(d) Directions Convolution units (scale : 5, scale : 7).

Acknowledgements

Part of this work was supported by Grant-in-Aid for Scientific Research(C) JSPS KAKENHI Grant Numbers 23560400 and 2459102, and Adaptable and Seamless Technology Transfer Program through Target-driven R&D, JST.

References

[1] 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.
[2] T. Tamaki, et al., “Computer-aided colorectal tumor classification in NBI endoscopy using local features, medical image analysis,” Available online 13 September 2012, ISSN 1361-8415, 10.1016/j.media.2012.08.003. 2012.
[3] http://www.vlfeat.org/
[4] http://www.csie.ntu.edu.tw/~cjlin/libsvm/
[5] T. Mishima, et al., "A Simple and Effective Hardware Oriented Feature Extraction Algorithm for Colorectal Endoscopic Images with NBI Magnification,” Proc. of the 28th International Conference on Circuits / Systems, Computers and Communications (ITC-CSCC2013), pp. 567-570, June 30-July 3, Yeosu, Korea, 2013.
[6] 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, June 30-July 3, Yeosu, Korea, 2013.