# Implementation of Generalized Hough transform for autonomous inspection systems

Jun Won Mun[1], Yune Seok Jang, Yoo Jun Nam and Jae Seok Kim[a]
Department of Electrical & Electronic Engineering, Yonsei University
E-mail: [1]mjw5554@yonsei.ac.kr

***Abstract* - Generalized Hough transform (GHT) is a well-known algorithm for image registration and applicable for various other applications. This paper presents a hardware design of autonomous inspection system based on GHT algorithm. In our design, we proposed a 2-stage GHT to overcome the disadvantages of GHT, such as high cost and requiring large computational memory capacity. We implement our modified GHT with Zynq7020 FPGA, and both hardware and software are co-designed with Vivado toolkit. Also, our system is also designed in ASIC. From the simulations and experimentations, we could achieve 65 fps with FPGA and 130 fps for the results with ASIC design when the inputs were VGA images. Our chip design requires 317.4K gates while consuming 135.9mW. In addition, the MATLAB simulation is carried out to show that our modified GHT is able to achieve comparable results with the conventional GHT.**

***Keywords*—Autonomous inspection, Hardware design, Slim Generalized Hough transform (SGHT)**

## I. INTRODUCTION

The generalized Hough transform (GHT) [1] is very effective image registration algorithm for various applications, *i.e*, industrial inspection, remote sensing, stereo matching, fingerprint matching and object tracking [2-4]. Compared with other techniques such as normalized cross correlation [5] and scale invariant feature transform [6], GHT is more robust to illumination change, clutter, occlusion, and noise. However, GHT is hard to implement in hardware due to its enormous memory requirements. Although several techniques are proposed to reduce memory requirements, they are not applicable when the edges are not sparse enough [10], [11].

As shown in Fig. 1, image registration in autonomous inspection can be assumed to be 2D image registration since the product and the camera have same geometric relation for both template and input images. In addition, the product always has similar edge distributions such as its boundary. In this paper, we propose simple yet effective GHT algorithm and its hardware implementation, which can be used in autonomous inspection. We introduce our modified GHT first, then hardware design will be described next. In the experiment results, both FPGA and ASIC implementation environments and results are presented.

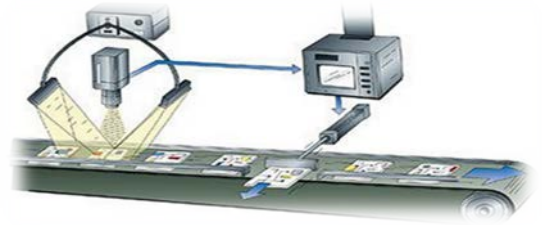

Fig. 1. Example figure of autonomous inspection.
Image courtesy of http://www.ecvv.com.

## II. PROPOSED GENERALIZED HOUGH TRANSFORM

Our proposed algorithm, so called slim GHT (SGHT), tackles drawbacks of conventional GHT. The GHT consists of two phases: online and offline. In *offline phase*, templates are registered in the reference table (so-called *R-table*) with gradient angle of edges and relative coordinate between edge pixel and reference point as shown in Fig. 2. In online phase, both position and rotation angle of template are obtained from the image of product by using R-table, which is constructed in offline phase. Each of pixel in search image is assumed that one of 2D transformed edges of registered template, and both displacement of reference point and orientation of template in search image are voted in accumulate matrix *acc* as shown in Fig. 3. Finally, the true template pose can be predicted which has maximum value in *acc* matrix. Since the matrix requires 3-dimensiona memory space, the memory requirement is enormous.

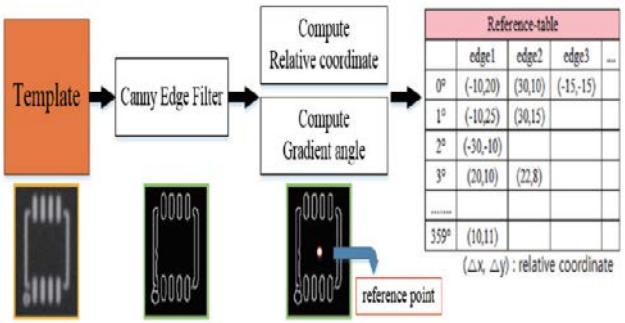


Fig. 2. Offline phase in GHT. Template is converted to R-table.

a. Corresponding author; jaekim@yonsei.ac.kr

Manuscript Received Feb. 22, 2018, Revised Mar. 13, 2018, Accepted Mar. 27, 2018

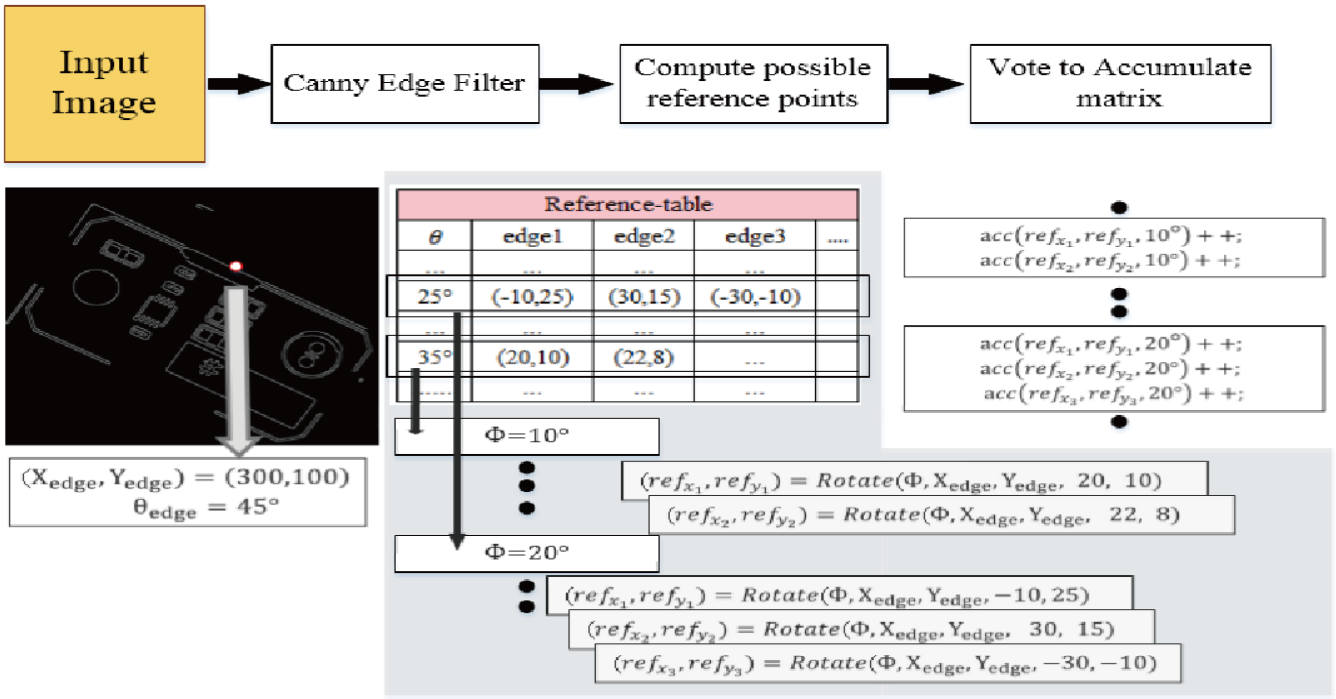


Fig. 3. Online phase in GHT. Possible template position $(\boldsymbol{ref}_x, \boldsymbol{ref}_y)$ and rotation angle $\Phi$ are voted in accumulate matrix.

Offline phase is not critical when considering real-time processing. Thus, now we focus on only for online phase. In the proposed SGHT, we employ 2-stage scheme, as rotation angle is firstly obtained with gradient angle histogram correlation and displacement is obtained in the second stage. The online phase block diagram is shown in Fig. 4.

1st stage: The key point of our proposal is not using template but reference image in this stage. In autonomous inspection, the product always has the same shape such as rectangle, and the template is a part of the product. Most of gradient angles in reference image are not changed even if the template has wrong shape or missing parts. In some cases, using reference image performs better than using just template. Thus, gradient angle histogram correlation calculation is simple but effective method in this application to obtain orientation. Generally, histogram of gradient angle can be obtained after visiting all edge pixels, and correlation is computed by shifting and multiplying for $n_r$ times, which $n_r$ is the number of possible orientation. Suppose the angle resolution is 1 degree, then $n_r$ is 360. However, in the hardware design, it could be computed in simple way with adder and LUT only. Suppose correlation vector is *corr* and gradient angles of reference and search image are $\theta_r^i$ and $\theta_s$, respectively. $hist[\theta_r^i]$ is the gradient angle histogram of reference image with degree of $i$. Then, correlation vector can be constructed as follows:

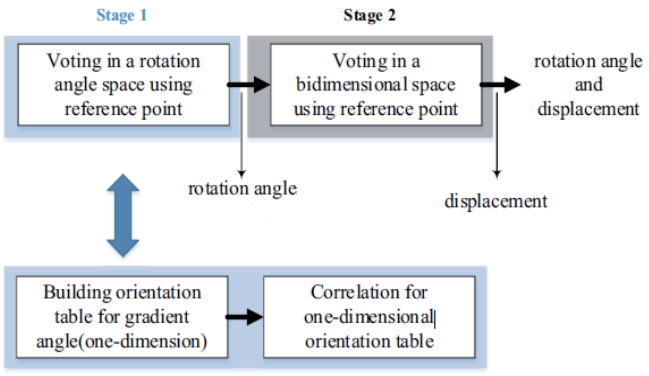


Fig. 4. Block diagram for proposed SGHT

Procedure 1: Calculating correlation of gradient angle

```
Begin
  Initialize corr
  For all edge pixels of search image
  For i=1 to n_r
        mod = modulo(θ_s-θ_r^i,360)
        corr[mod] =corr[mod]+hist[θ_r^i]
  EndFor
End
```

2nd stage: In this stage, the accumulate matrix for voting space is 2-dimensional matrix since orientation is obtained in the previous stage. Moreover, edge pixels are now assumed to be one of the template image's pixel without 2D transformation. Thus, the memory requirement is as same as image resolution. In our proposal, we further reduce accumulate matrix size while preserving displacement accuracy by employing double voting with coarse-to-fine [12]. Although coarse-to-fine method is a simple and effective way to reduce memory, it is inevitable to avoid the degradation of accuracy in spatial domain. As described in our previous work in [12], adaptive weight and double voting schemes are used. For the adaptive weight scheme, the increasing value for each vote is adaptively determined depending on the ratio of the number of gradient angle. For the double voting scheme, the modular of calculated possible position is voted to neighborhood pixels, and finding the maxima from the entire subsampled accumulator matrix and

the neighborhood pixels centered to subsampled maxima. According to [12], the memory requirement and computational complexity is reduced more than 99% and 83% compared with GHT [1], respectively. Note that this achievement is not valid in general 2D template matching; autonomous inspection application is the special case which can be assumed that the products have common shape without complex environments.

## III. HARDWARE BLOCK DESCRIPTION

In this section, detail descriptions for several main blocks in online phase are presented. To make the system for real-time processing, both two stages are processed concurrently.

In Fig. 5, top block diagram of our SGHT is shown. The volume of data is described for the case of the VGA input images. In our design, the input image is a grey image. In the first sub-block, canny edge detector [7] subtracts edge pixels from the input image, then the gradient angle and relative coordinate are produced. Then, the rotation angle is obtained from the next sub-block. In this block, gradient angle histogram of the reference image is stored in offline phase and correlation vector is constructed as described in Procedure 1. For each search image pixel, correlation vector is accumulated 360 times only for edge pixels. In our design, some buffer and 40 adders are used to compute orientation angle. In the next block, 2nd stage, voting process for obtaining displacement is carried out. In this sub-block, R-table information is used to compute the candidates of reference point coordinates. Then, candidates are accumulated in the accumulate matrix and index maximum entry is obtained by using 1 comparator.

In Fig. 6, top module of canny edge detector is shown. In the first module, the input image is filtered by smoothing operator, then an gradient value is calculated. Depending on the gradient values, mode is selected and the angle of each pixel is computed. Finally, non-maximum suppression methods determined whether the pixel is edge pixel or not. The output of this module is the gradient angle and coordinates of edge pixels.

Fig. 7 is a block diagram of the last step which decides the displacement of template. In the address generator block, all of the addresses of accumulate matrix are generated. The read address and initialize address are assigned to the accumulate matrix memory block, and initialized data is always 0 for the next search image. The matrix data is then come from the memory and the maximum value is found by using comparator. At the same time, the result of comparator decides the index of the maximum entry in the *acc*. Finally, position is determined after scanning all pixels of the image

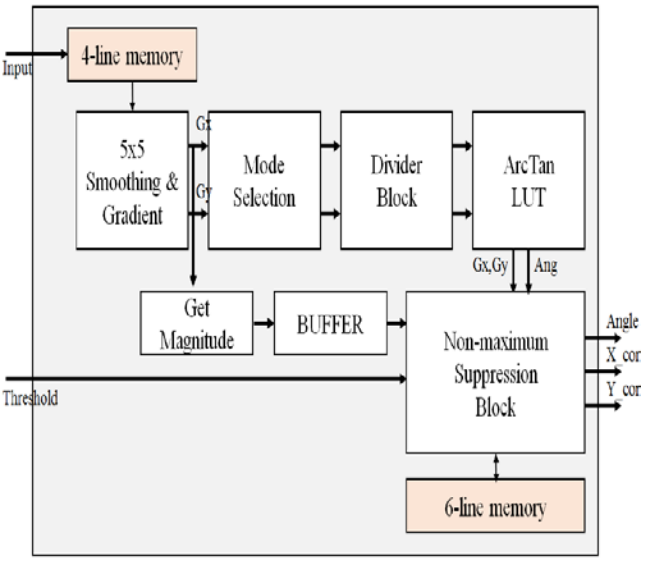


Fig. 6. Top block diagram for Canny edge detector.

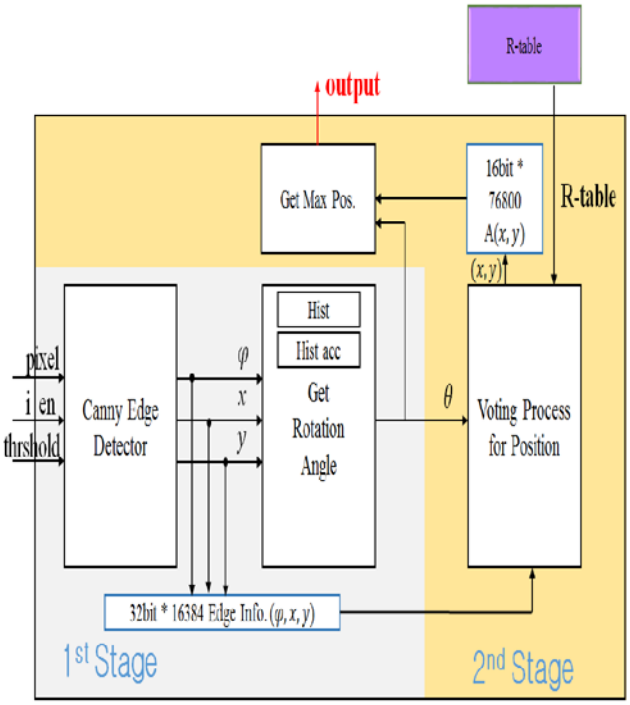


Fig. 5. Top block diagram for proposed SGHT.

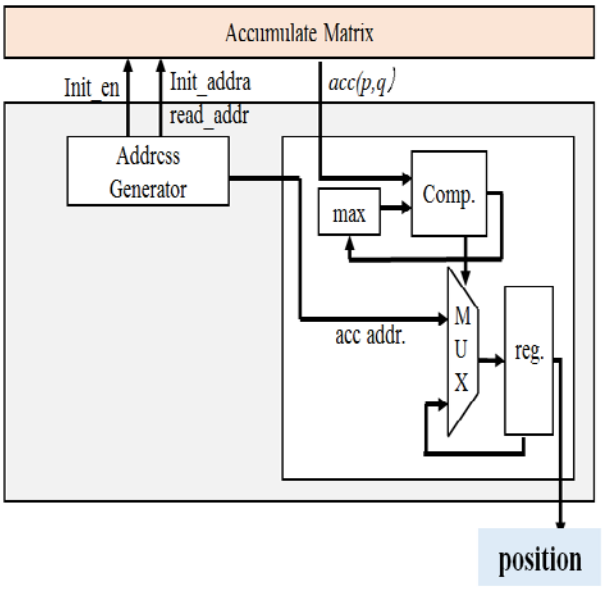


Fig. 7. Block diagram of the last step which decides the displacement

## IV. FPGA IMPLEMENTATION AND RESULTS

The proposed image registration system is synthesized and mapped on Zynq7020 field programmable gate array (FPGA) for both offline and online phases.

In Fig. 8, top block diagram of proposed FPGA design is shown. At first, the images are stored in DDR3 memory. In the offline phase, R-table and histogram of gradient angle are stored in the registers and block memory. And the accumulate matrix memory is initialized as 0. The R-table is constructed using ARM-Cortex-A9 in Zynq FPGA with SDK software.

In online phase, each of search image pixel from DDR3 memory passes through HP port per 1 clock by using CDMA IP core. Then, the output values of SGHT are stored in BRAM2. In our FPGA implementation, our operating clock frequency is set to 50MHz. For VGA images, we verify that SGHT is performed with 65 fps in Zynq7020 FPGA. The Utilization of post implementation is summarized in Table I.

In Fig. 9, timing diagram of our implementation is presented. The key point is that stage 1 and 2 are concurrently executed. As the result, 0.46 frame is delayed for the first output values, that is 0.03s in our design. After that, each of search image was processed in only 0.015s.

TABLE I.
Utilization of our FPGA implementation

| Item | Usage (%) |
|---|---|
| Flip-Flop | 15 |
| LUT | 69 |
| Memory LUT | 2 |
| BRAM | 87 |
| Max Frequency | 80.7MHz |

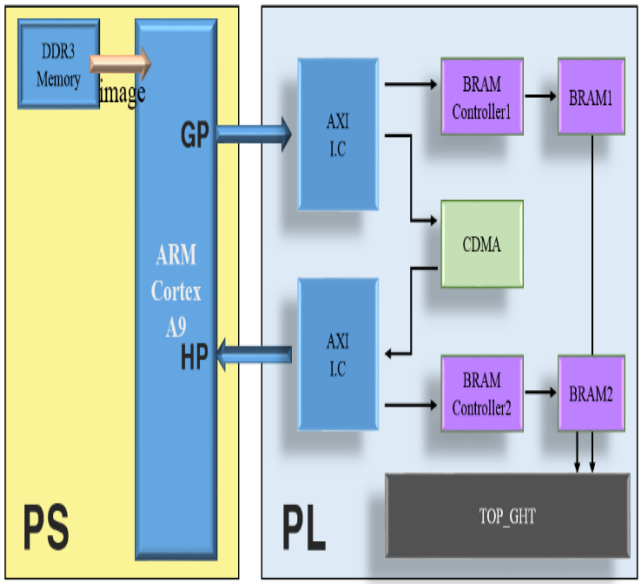


Fig. 8. Block diagram of the last step which decides the displacement.

## V. CHIP IMPLEMENTATION RESULTS

The proposed SGHT for autonomous inspection is synthesized and implemented through CMOS 65nm technology. In our design, we only implemented offline phase of SGHT on chip since the template information should be changed depending on the products. In our FPGA implementation, online phase is performed by ARM core in PS. As synthesize results, gate counts, power, cell area and total area are presented in Table II.

Fig. 10 and 11 show the photos of the implemented chip. The chip is implemented using the 65nm CMOS technology and packaged with 208-pin LQFP. The chip size is 4mm by 4mm, and the supply voltage is 3.3V and 1.2V for I/O and core each.

We compared our design with several existing implementations [8], [9]. In our knowledge, there are no ASIC implementations in the current literature, thus only CUDA and FPGA design are compared here. In addition, we assumed that the input image resolution is 640 by 480 and scale is always 1 that is scale invariant case. In [8], FPGA implementation of GHT is presented with efficient version of GHT [10]. In their work, canny operator subtracts edges and gradient angle is computed by using orientation table.

TABLE II.
Synthesize results of SGHT with CMOS 65nm technology

| Item | Description |
|---|---|
| Operating Clock Frequency | 100MHz |
| Gate Count | 317.4K |
| Power | 135.9096mW |
| Area | 0.679mm$^2$ |
| Used Cell | 95,574 |

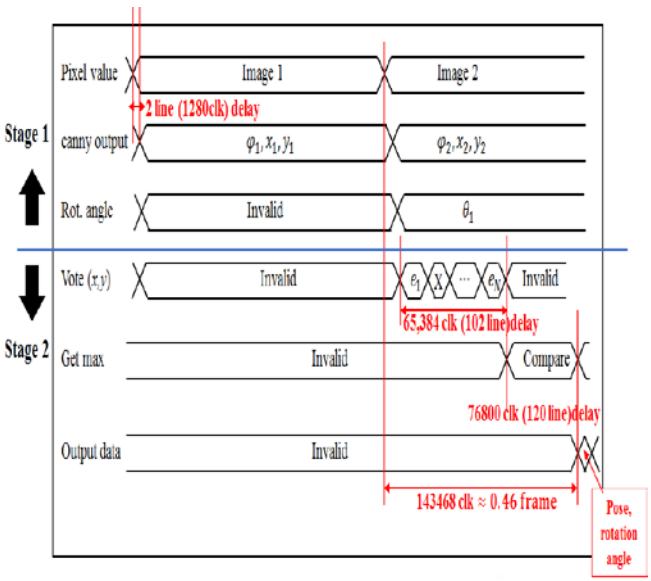


Fig. 9. Timing diagram of the proposed SGHT.

TABLE III.
Comparison of the proposed system with previously implemented with FPGA

| **Item** | **[8]** | **[9]** | **Proposed** |
|---|---|---|---|
| Device | 3s1000ft2566-4 | GPU | xc7z020clg484-1 |
| Input resolution | 44x36 | 200x150 | 640x480 |
| Phase | Online phase | Online phase | Online and offline phases |
| Flip Flop | 3858 | - | 15014 |
| LUT | 3062 | - | 36470 |
| Memory LUT | 2841 | - | 275 |
| I/O | 85 | - | 41 |
| Fps **(when VGA)** | **40** | **~3** | **65** |

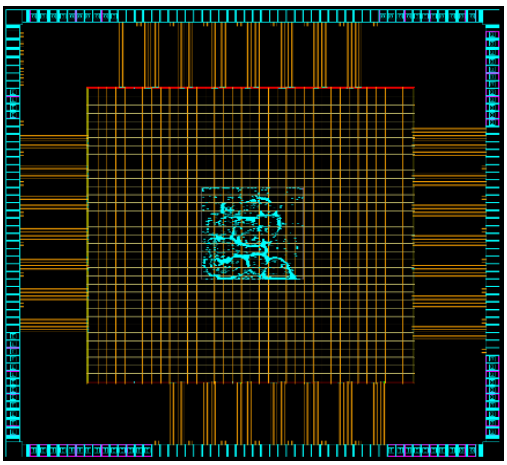

Fig. 10. Layout of our SGHT chip design

Fig. 11. Die photo of our SGHT chip design.

Then, pair of edges are used to calculate histogram correlation. Thus, 2D space correlation is required to obtain orientation. Note that in our proposal, 1D correlation is enough to obtain orientation. In [8], input image resolution is set to 44 by 36 and higher resolution would be reduced in the preprocessing. In our experiment results, reducing resolution was not good method for reducing memory requirement and computational complexity, since the accuracy of the displacement and orientation became worse. The device of [8] is SPARTAN-3 Starter Board from Digilent provided with a XC3S1000 FT256 speed grade 4 chip. The implementation results are compared in Table III. For the last row, fps is computed when the input image is VGA.

## VI. SIMULTATION RESULTS

In this section, we briefly compared the proposed algorithm results with GHT in MATLAB. In Fig. 12, images for reference and templates are shown. In this experiments, search images are produced by rotating reference image from 0 degree to 359 degree, and weak Gaussian noise is added for each images.


Fig. 12. (a) Reference image, (b)-(f) templates. Image courtesy of Baumer.

Comparison of the SGHT and GHT for Fig. 12 is described in Table IV. The number of test images is 360 that all images have different orientation. Orientation accuracy is measured when the orientation result is exactly same with true orientation and displacement error is measured by summation of absolute difference between correct position and produced results. Experimental results showed that SGHT is much accurate compared with GHT.

TABLE IV.
Orientation accuracy and displacement error for Fig. 12.

| | item | (b) | (c) | (d) | (e) | (f) |
|---|---|---|---|---|---|---|
| **[1]** | Orient.acc(%) | 100 | 98.89 | 75.56 | 100 | 100 |
| | Disp. Err(pixel) | 148 | 30 | 42 | 8 | 9 |
| **SGHT** | Orient.acc(%) | 100 | 100 | 100 | 100 | 100 |
| | Disp. Err(pixel) | 148 | 28 | 22 | 8 | 9 |

## VII. CONCLUSION

In this paper, we propose simplified GHT image registration technique which is applicable in autonomous inspection. By employing 2-stage strategy, memory requirement, which is the main drawback of GHT, is reduced greatly. We design our system with both FPGA and ASIC design. In FPGA, Zynq7020 is selected to design whole procedure of inspection with HW/SW co-simulation architecture. We achieve 65fps and 130fps for FPGA and ASIC design, respectively. These results seem much faster than other implementation works. As the synthesize results with 65nm CMOS technology, our system requires 317.4K gates.

## AKNOWLEDGMENT

This work was supported by the Technological Innovation R&D Program (S2342832) funded by the Small and Medium Business Administration (SMBA, Korea)" and supported by IDEC (IPC, EDA Tool, MPW).

## REFERENCES

[1] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," Pattern Recognition, vol.13, no.2, pp.111-122, 1981.
[2] Z. Wang, H Yang, W. Li, Z. Yin, "Super-Resolving IC Images with an Edge-Preserving Bayesian Framework," *IEEE Trans. Semicond. Manuf.*, vol.27, no.1, pp.118-130, Feb. 2015.
[3] J. Xu, X. Sun, D. Zhang, and K. Fu, "Automatic detection of inshore ships in high-resolution remote sensing images using robust invariant generalized Hough transform," *IEEE Geosci. Remote Sens. Lett.*, vol.11, no.12, pp.2070-2074, Dec. 2014.
[4] N. S. Vu, and A. Caplier, "Enhanced patterns of oriented edge magnitudes for face recognition and image matching," *IEEE Trans. Image Process.*, vol.21, no.3, pp.1352-1365, Mar. 2012.
[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol.60, no.2, pp. 91-110, 2004.
[6] A. J. Crispin, and V. Rankov, "Automated inspection of PCB components using a genetic algorithm template-matching approach," *Int. J. Adv. Manuf. Technol.*, vol.35, no.3, pp.293-300, Dec. 2007.
[7] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.8, no.6, pp.679-698, Nov. 1986.
[8] S. R. Geninatti, J. I. B. Benítez, and M. H. Calviño, "FPGA implementation of the generalized Hough transform," in Proc. *Int. Conf. Reconfigurable Computing and FPGAs*, 2009, pp. 172–177.
[9] R. Strzodka, I. Ihrke, and M. Magnor, "A graphics hardware implementation of the generalized hough transform for fast object recognition, scale, and 3D pose detection," in Proc. 12th *Int. Conf. Image Anal. Process*, 2003, pp. 188–193.
[10] Guil N.; Gonzalez J.M. and Zapata E.L.; "Bidimensional shape detection using an invariant approach," Pattern Recognition, vol. 32, pp. 1025-1038, 1999.
[11] R. K. K. Yip, P. K. S. Tam, and D. N. K. Leung, "Modification of hough transform for circles and ellipses detection using a 2-dimensional array," Pattern Recognition, vol.25, no.9, pp.1007-1022, Sep. 1992.
[12] Mun, J., & Kim, J. (2017, December). 2D template matching for automated inspection using generalized hough transform. In Research and Development (SCOReD), *2017 IEEE 15th Student Conference on* (pp. 461-466). IEEE.

**Jun Won Mun** received his B.S. degree in Electrical and Electronic engineering from Yonsei University, Seoul, Korea in 2013, and is currently pursuing a direct-entry Ph.D. degree at Yonsei University. His current research interests include template matching and SoC/VLSI design for image signal processing.

**Yune Seok Jang** received his B.S. degree in Electrical and Electronic engineering from Yonsei University, Seoul, Korea in 2015, and is currently pursuing a direct-entry Ph.D. degree at Yonsei University. His current research interests include template matching and SoC/VLSI design for image signal processing.

**Yoo Jun Nam** received his B.S. degree in Electrical and Electronic engineering from Yonsei University, Seoul, Korea in 2016, and is currently pursuing a direct-entry Ph.D. degree at Yonsei University. His current research interests include template matching and SoC/VLSI design for image signal processing.

**Jae Seok Kim** received his B.S. degree in electronic engineering from Yonsei University, Seoul, Korea, in 1977, an M.S. degree in electrical and electronic engineering from KAIST, Daejon, Korea in 1979, and a Ph. D degree in electronic engineering from Rensselaer Polytechnic Institute, NY, USA in 1988.From 1988 to 1993, he was a member of the technical staff at AT&T Bell Laboratories, USA. He was Director of VLSI Architecture Design Laboratory of ETRI from 1993 to 1996.He is currently a professor in the Electrical and Electronic Engineering Department at Yonsei University, Seoul, Korea. His current research interests include SoC design, high-performance VLSI digital signal processor design, and CAD S/W.