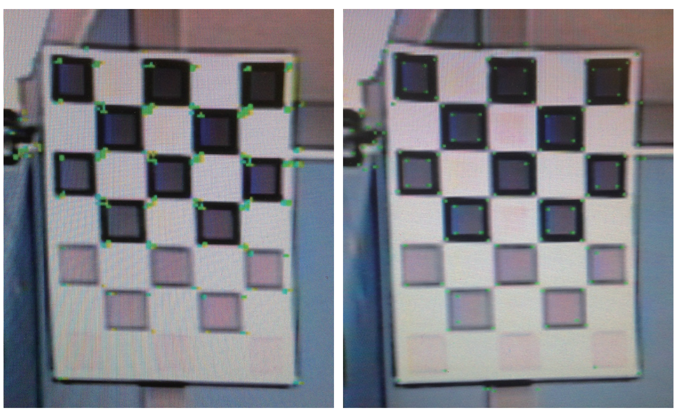
TABLE II
LOGIC CAPACITY AND UTILIZATION OF CORNER DETECTION SYSTEM

	Used	Available	Utilization
ALMs	2,694	41,910	6%
Block memory bits	721,572	5,662,720	13%
DSP blocks	57	112	51%

TABLE III
LOGIC DISTRIBUTION OF THE CORNER DETECTION SYSTEM

	Used	Ratio
ALMs		
Corner Detector	785.8	29%
LCD Controller	406.0	15%
Camera Controller	338.8	13%
Interconnect + Port	1,163.2	43%
Block memory bits		
Corner Detector	704548	98%
LCD Controller	8576	1%
Camera Controller	8448	1%
Interconnect + Port	0	0%
DSP blocks		
Corner Detector	49	86%
LCD Controller	7	12%
Camera Controller	1	2%
Interconnect + Port	0	0%



(a) Hardware implementation (b) Software implementation

Fig. 7. Result images of the corner detection system

Logic Module) in the table is a unit of the logic resource in Altera's FPGA device families. The VGA screen image during the execution is shown in Fig. 7(a). The corners are indicated by the green markers.

As described before, the software for the comparison runs on the ARM CPU. The software is developed under the Linux environment and implemented by using OpenCV 2.4.8 and mainly the cv::goodFeaturesToTrack() function. We configured each filter window with the size of 3×3, which is the same as in the hardware-implemented system. The VGA screen image during execution is shown in Fig. 7(b).

TABLE IV
PERFORMANCE OF THE CORNER DETECTION SYSTEM (3,000 FRAMES)

	Processing Time (sec.)	Average FPS
Software implemented	1189.21	2.52
Hardware implemented	18.40	163.03

The processing times for dealing with 3,000 frames are shown in Table IV. The hardware implementation achieves 64.63 times faster performance in term of FPS (frames per second) than the software implementation. The 163.03 FPS is good enough for the real-time processing.

VII. CONCLUSION

In this paper, we proposed an architecture of a hardware library that facilitates construction of image processing systems by reassembling block modules. The library provides various basic functions of image processing as primitive blocks. It is possible to realize various algorithms by flexibly combining these blocks. In order to achieve the faster and easier implementation of the image processing systems, we have prepared various useful primitive blocks which have a simple interface for connecting each other. We have constructed a real image processing system for the purposes of demonstrating the use of the library and evaluating the performance compared with that of the software implementation which runs on a general-purpose processor.

In the future, we will develop more image processing blocks and add them to the hardware library so that it will be able to adapt to various kinds of image processing systems. And also we will revise the architecture so that it has a more user-friendly interface for the configuration of the systems. This project is currently under development. We will make the hardware library public-accessible via Internet.

REFERENCES

[1] Z. Zhang, "Microsoft kinect sensor and its effect," *MultiMedia, IEEE*, vol. 19, no. 2, pp. 4–10, 2012.

[2] D. Mandal, J. Sankaran, A. Gupta, K. Castille, S. Gondkar, S. Kamath, P. Sundar, and A. Phipps, "An embedded vision engine (eve) for automotive vision processing," in *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, June 2014, pp. 49–52.

[3] C. Farabet, C. Poulet, and Y. LeCun, "An fpga-based stream processor for embedded real-time vision with convolutional networks," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, Sept 2009, pp. 878–885.

[4] (2015) Video and image processing suite megacore functions. Altera Corporation. [Online]. Available: https://www.altera.com/products/intellectual-property/ip/dsp/m-alt-vipsuite.html

[5] (2015) Video and image processing pack. Xilinx Inc. [Online]. Available: http://www.xilinx.com/products/intellectual-property/ef-di-vid-img-ip-pack.html

[6] ARM, "Amba 4 axi and ace protocol specification," 2013. [Online]. Available: http://www.arm.com/products/system-ip/amba-specifications.php

[7] "Recommendation itu-r bt.656-5," 2007. [Online]. Available: http://www.itu.int/rec/R-REC-BT.656/e

[8] "Recommendation itu-r bt.601-7," 2011. [Online]. Available: http://www.itu.int/rec/R-REC-BT.601/e

[9] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.