Fixed Homography–Based Real-Time SW/HW Image Stitching Engine for Motor Vehicles

Jung-Hee Suk, Chun-Gi Lyuh, Sanghoon Yoon, and Tae Moon Roh

In this paper, we propose an efficient architecture for a real-time image stitching engine for vision SoCs found in motor vehicles. To enlarge the obstacle-detection distance and area for safety, we adopt panoramic images from multiple telegraphic cameras. We propose a stitching method based on a fixed homography that is educed from the initial frame of a video sequence and is used to warp all input images without regeneration. Because the fixed homography is generated only once at the initial state, we can calculate it using SW to reduce HW costs. The proposed warping HW engine is based on a linear transform of the pixel positions of warped images and can reduce the computational complexity by 90% or more as compared to a conventional method. A dual-core SW/HW image stitching engine is applied to stitching input frames in parallel to improve the performance by 70% or more as compared to a single-core engine operation. In addition, a dual-core structure is used to detect a failure in state machines using rock-step logic to satisfy the ISO26262 standard. The dual-core SW/HW image stitching engine is fabricated in SoC with 254,968 gate counts using Global Foundry's 65 nm CMOS process. The single-core engine can make panoramic images from three YCbCr 4:2:0 formatted VGA images at 44 frames per second and frequency of 200 MHz without an LCD display.

Keywords: Image stitching, panoramic image, motor vehicles, real time, homography, vision SoC, ISO26262.

Manuscript received Jan. 29, 2014; revised Nov. 4, 2015; accepted Nov. 11, 2015.

This work was supported by the IT R&D program of MKE/KEIT (KI002162, Multi-camera based High Speed Image Recognition SoC Platform).

Jung-Hee Suk (corresponding author, jhsuk@etri.re.kr), Chun-Gi Lyuh (cglyuh @etri.re.kr), and Tae Moon Roh (tmroh@etri.re.kr) are with the Information & Communications Core Technology Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Sanghoon Yoon (shyoon11@keti.re.kr) is with the SoC Platform Research Center, KETI, Seongnam, Rep. of Korea.

I. Introduction

A panoramic image is a wide-view image synthesized with multiple consecutive images together on a common virtual planar surface, on a cylinder, or on a sphere, up to a full view of 360 degrees. The feature points of overlapped regions between the images are first extracted and matched; a homography is then estimated and represented in terms of a transformation matrix. Then, the images are warped onto the panorama surface using the estimated homography matrix (H-matrix) between the panorama surface and image coordinates [1]–[2]. Panoramic images provide users with wide scenes that cannot be captured by a single image from a normal camera. Thus, panorama synthesis overcomes the limitations of viewing angles and resolutions in normal cameras [3]. Traditional panoramic images have a single viewpoint, known as the "center of projection" [4]–[6]. Panoramic images can be captured by panoramic cameras, using special mirrors [7]–[8], by mosaicing a sequence of images from a rotating camera [9]–[10], or by mosaicing together images from a rotating pair of stereo cameras [11].

Panorama image systems have been popular in mobile cameras and PC environments. Many algorithms and commercial systems for image stitching have been reported [12]–[16]. Early panorama systems assumed fixed-camera motions, such as horizontal rotations with fixed angles, using user-constrained interfaces. This simplified the calculations of a transformation matrix, but the degrees of freedom to handle panoramic images were restricted [14]. In panorama algorithms, feature matching and transformation estimation are the most important procedures since images are spatially warped from any resulting transformations. Brown and Lowe exploited a descriptor-based feature, SIFT, to match image

correspondences and estimate arbitrary camera motions automatically [17]. Descriptor-based features such as SIFT [18] and SURF [19] improve the performance of automatic panorama synthesis. However, since feature detection and automatic feature matching carry a high computational load, these approaches are not suitable for systems with low computing power [3].

According to the increased attention paid to the safety of motor vehicles, the developments of vision SoCs such as eyeQ1 [20] and eyeQ2 [21] for driving assistance have recently become more active than ever. Many driving assistance systems based on these vision SoCs have been developed and their performance verified using real vehicles on actual roads [22].

In this paper, we design an efficient software (SW)/hardware (HW) image stitching engine for motor vehicles that can make panoramic images to enlarge the obstacle-detection distance and area for safety in real time with a small HW area. The remainder of this paper is organized as follows. The architecture of the proposed SW/HW image stitching engine is shown in Section II. Section III describes the SW algorithms to generate a fixed H-matrix. Section IV describes the efficient HW circuits used to calculate the warping algorithm and dual-core structure used to process input frames in parallel and detect any failures. The experimental results and demonstration are shown in Section V. Finally, we provide some concluding remarks in Section VI.

II. Proposed Architecture of Image Stitching Engine

To make the image stitching engine more efficient, we propose the following considerations. First, a telegraphic camera is superior to a pantoscopic one in recognizing objects at a distance. It provides much finer figures than a pantoscopic camera. Second, the H-matrix of the left and right images for image warping is not changed much because cameras are fixed to each other on the body frames of vehicles. Thus, the H-matrix is extracted only once from the initial frame of a video sequence. Third, an SW is suitable for calculating an H-matrix because one operation at the beginning of the system is sufficient when stitching all frames. Other processes such as warping and blending are computed using HW, except for H-matrix generation using SW. This can significantly reduce the HW area, such as the Gaussian function, gradient operation, keypoint descriptor, random sample consensus (RANSAC), and singular value decomposition (SVD). Fourth, we adopt a dual-core structure to detect a failure of the stitching engine to meet the ISO26262 standard and improve performance through parallel processing.

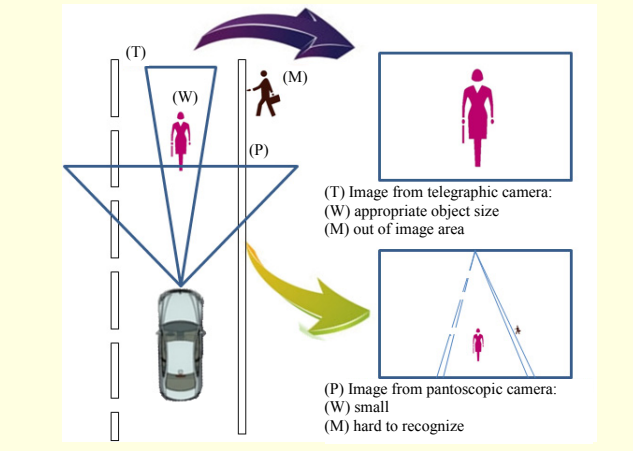


Fig. 1. Comparison of pantoscopic and telegraphic cameras.

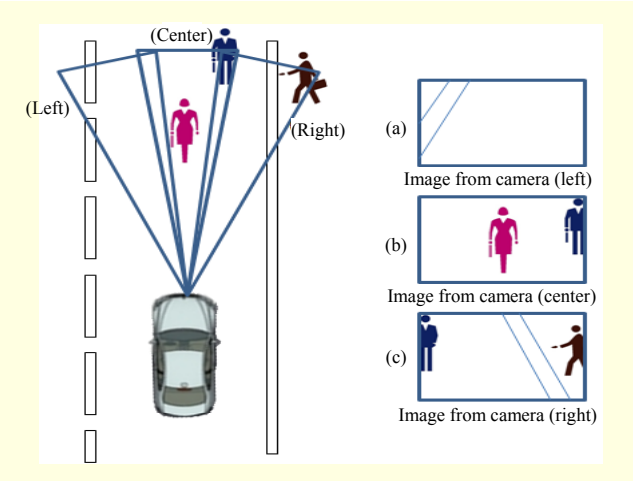


Fig. 2. Images from multiple telegraphic cameras.

1. Considerations for Motor Vehicle Cameras

As shown in Fig. 1, a pantoscopic camera can take wider images than a telegraphic camera, while the latter gives much finer resolution than the former when the objects are at a distance. To avoid crashing with an object, a telegraphic camera would be better than a pantoscopic camera, as shown in Fig. 1. On the other hand, objects across the road from the edge of a sidewalk, such as (M) in Fig. 1, cannot be detected using a telegraphic camera. Although high-resolution images can be used to observe an object at a distance in a wide area, we adopt multiple cameras to lower the aspect ratio of the observation area, as shown in Fig. 2. The man standing on the road in Fig. 2 spans the border of both images (b) and (c). While it is hard to detect these two half-bodies in each figure through vision-based object detection, the miss rate of the full body can be significantly decreased if they are unified through image stitching. To enhance the detection performance, an image stitching function is needed [23].

ETRI Journal, Volume 37, Number 6, December 2015
http://dx.doi.org/10.4218/etrij.15.0113.1120

2. Fixed-Homography Method for Warping Images

An H-matrix is extracted only once from the initial frame of a video sequence and is used to warp all input images without regeneration. This is reasonable since a vehicle's cameras are all fixed and their respective H-matrices are not changed much. There is, therefore, no reason to update each H-matrix at every frame. Actually, a blurring of the stitched images between frames occurs when using a H-matrix generated from every frame to warp the input images because a given H-matrix is selected randomly by the RANSAC algorithm with some unevenness for the same image. In addition, this is bad for both recognition accuracy and viewing stitched images. Using a fixed H-matrix, we can obtain three advantages — a decrease in the computational complexity for extracting the H-matrix at every frame, comfortable viewing images without blurring, and a removal of mismatch errors in the pixel position between frames for object recognition.

An H-matrix can also make only an affine transformation [24]–[25]. To reduce the HW complexity, the warping process using an H-matrix can be implemented using only linear operations, described in Section V.

3. SW/HW Co-design Architecture

We designed the image stitching engine as shown in Fig. 3. The SW engine is based on a 32-bit EISC microprocessor (MP) [26]. The HW engine consists of a warping module, blending module, and stitching controller. The warping module warps each image according to the H-matrix generated by the EISC processor. The blending module performs alpha blending, and the stitching controller controls all other modules. For efficient communications with the vision SoCs, the proposed engine supports an AXI interface. Control parameters such as the memory addresses used; the image size and format; and the blending area are set by the MP through the device driver. The proposed engine operates using a double-buffer which provide some marginal cycles for a frame since we assume a full-matrix AXI BUS structure [23].

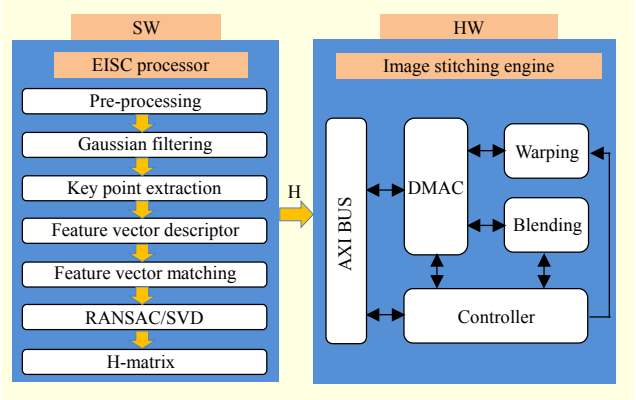


Fig. 3. SW/HW block diagram of image stitching engine.

4. Dual-Core Structure for ISO26262 Standard and Performance Improvement

A standard for the functional safety of road vehicles, ISO26262, was recently published [27]. We adopted lockstep blocks to detect a failure of the image stitching engine and dual-core structure to fulfill the requirements of this functional safety standard. When one of the engines belonging to the dual-core structure is out of order, only the other engine will operate normally instead of parallel processing. We can, therefore, obtain improvements in terms of functional safety and performance.

III. SW Algorithm Used to Generate Homography Matrix

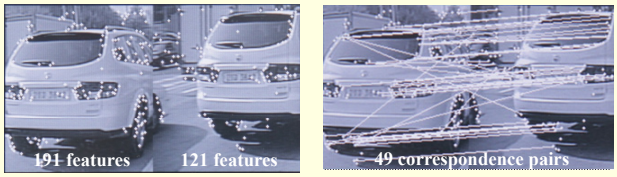
1. Feature Extraction

We first extract the feature points of input images through use of the SIFT algorithm. The major stages of the algorithm are as follows [18]:

1) Scale-space extrema detection: the first stage of extrema detection searches over all scales and image locations using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.
2) Keypoint localization: at each candidate location of the extremas, a detailed model based on the measures of their stability is used to select keypoints and to determine their location and scale.
3) Orientation assignment: one or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that have been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.
4) Keypoint descriptor: the local image gradients are measured at a selected scale in the region around each keypoint.

2. Correspondence Matching

We need to match the detected features using SIFT for all candidate features to find the best correspondences between stereo images from the left and right cameras [28]. Feature matching has been a bottleneck for real-time operation in feature-based methods [3]. Moreover, robust feature matching is required because the matching errors corrupt the H-matrix. The correspondence matching time is not effective for real-time processing, because a fixed H-matrix is generated only once. We heighten the matching accuracy using a nearest-



(a) (b)

Fig. 4. Correspondence matching result.

neighbor algorithm for all candidate features. Figure 4 shows the results of correspondence matching from a 32-bit EISC processor after extracting the feature points for 320 × 240 images. The number of left and right image feature points is 191 and 121, respectively. Additionally, the number of correspondence matching pairs is 49.

3. RANSAC and SVD

We use an iterative RANSAC method to estimate the best H-matrix that has minimum transformation errors for a stitched image. The RANSAC algorithm was first introduced by Fischler and Bolles [29] in 1981 as a method to estimate the parameters of a certain model starting from a dataset contaminated by large numbers of outliers [30]. When we estimate the H-matrix from the correspondence matching pairs, some errors (outliers) exist. The outliers are the correspondence matching pairs that have more errors above a threshold value for determining when the pairs fit the estimated H-matrix.

The major stages of RANSAC used to generate the H-matrix are as follows [30]:

1) Hypothesize: first, minimal sample sets (MSSs) are randomly selected from the input dataset, and the model parameters are computed using only the elements of the MSSs. At least four pairs of correspondences are necessary to estimate the H-matrix. Thus, our MSSs are the four pairs of correspondences, and the model parameters make up the H-matrix.
2) Test: in the second step, the RANSAC checks which elements of the entire dataset are consistent with the model instantiated with the parameters estimated in the first step. RANSAC terminates when the probability of finding a better ranked consensus set (CS) drops below a certain threshold.

Let $\mathbf{x}(\{d_1, \ldots, d_h\})$ be the parameter vector estimated using the dataset $\{d_1, \ldots, d_h\}$, where $h \geq k$ (k is the cardinality of an MSS). A model manifold matrix, $\mathbf{H}$, can be defined as

$$\mathbf{H}(\mathbf{x}) \overset{\text{def}}{=} \{d \in R^d : f_{\mathbf{H}}(d;\mathbf{x}) = 0\}, \tag{1}$$

where $\mathbf{x}$ is a parameter vector and $f_{\mathbf{H}}$ is a smoothing function whose zero-level set contains all points that fit model $\mathbf{H}$ instantiated with parameter vector $\mathbf{x}$. We define the error associated with datum d with respect to manifold $\mathbf{H}(\mathbf{x})$ as the distance from d to $\mathbf{H}(\mathbf{x})$. The distance function is the Euclidean norm,

$$e\big(d, \mathbf{H}(\mathbf{x})\big) = \min_{d' \in \mathbf{H}(\mathbf{x})} \sqrt{\sum_{i=1}^{n} (d_i - d_i')^2}. \tag{2}$$

The number of RANSAC iterations for estimating $\mathbf{H}$ is determined as in [3] as

$$T = \left\lceil \frac{\log \varepsilon}{\log(1-q)} \right\rceil. \tag{3}$$

Let q be the probability of sampling an MSS that produces an accurate estimate of the model parameters from dataset D. Here, q is usually set to 0.99. Consequently, the probability of picking an MSS with at least one outlier is $1–q$. If we construct h different MSSs, then the probability that all of them are contaminated by outliers is $(1-q)^h$. We would like to choose a large enough h (that is, the number of iterations) so that the probability $(1-q)^h$ is smaller than or equal to a certain probability threshold, ε (often called the alarm rate).

We use the SVD method to calculate the H-matrix described above from four pairs of correspondences. SVD is based on a theorem from linear algebra, which states that a rectangular matrix $\mathbf{A}$ can be broken down into the product of three matrices — an orthogonal matrix $\mathbf{U}$, a diagonal matrix $\mathbf{S}$, and the transpose of an additional orthogonal matrix; in this case, $\mathbf{V}$. The theorem is usually presented similar to [31] as

$$\mathbf{A}_{mn} = \mathbf{U}_{mn}\mathbf{S}_{nn}\mathbf{V}_{nn}^{\mathrm{T}}, \tag{4}$$

where $\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{I}$, $\mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$; the columns of $\mathbf{U}$ are orthonormal eigenvectors of $\mathbf{A}\mathbf{A}^{\mathrm{T}}$, the columns of $\mathbf{V}$ are orthonormal eigenvectors of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$, and $\mathbf{S}$ is a diagonal matrix containing the square roots of the eigenvalues from $\mathbf{U}$ or $\mathbf{V}$ in descending order.

After decomposition of matrix $\mathbf{A}$, its inverse is trivial to compute — if matrix $\mathbf{A}$ is a square, $N \times N$, then $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{S}$ are all square matrices of the same size. Because $\mathbf{U}$ and $\mathbf{V}$ are orthogonal, their inverses are equal to their transposes, and because $\mathbf{S}$ is diagonal, its inverse is a diagonal matrix whose elements are the reciprocals of elements S_j. From (4) the inverse of $\mathbf{A}$ is

$$\mathbf{A}^{-1} = \mathbf{V}_{nn} \cdot [\mathrm{diag}(1/S_j)] \cdot \mathbf{U}_{nn}^{\mathrm{T}}. \tag{5}$$

We define $\mathbf{H}$ to be a homography matrix for a transformation from the correspondence of images A and B. The matrix $\mathbf{H}$ can be calculated using the inverse of $\mathbf{A}$ as follows:

$$\begin{aligned} \mathbf{A}\mathbf{H} &= \mathbf{B} \\ \mathbf{H} &= \mathbf{A}^{-1}\mathbf{B} \\ \mathbf{H} &= \mathbf{V}_{nn} \cdot [\mathrm{diag}(1/S_j)] \cdot \mathbf{U}_{nn}^{\mathrm{T}}\mathbf{B}. \end{aligned} \tag{6}$$

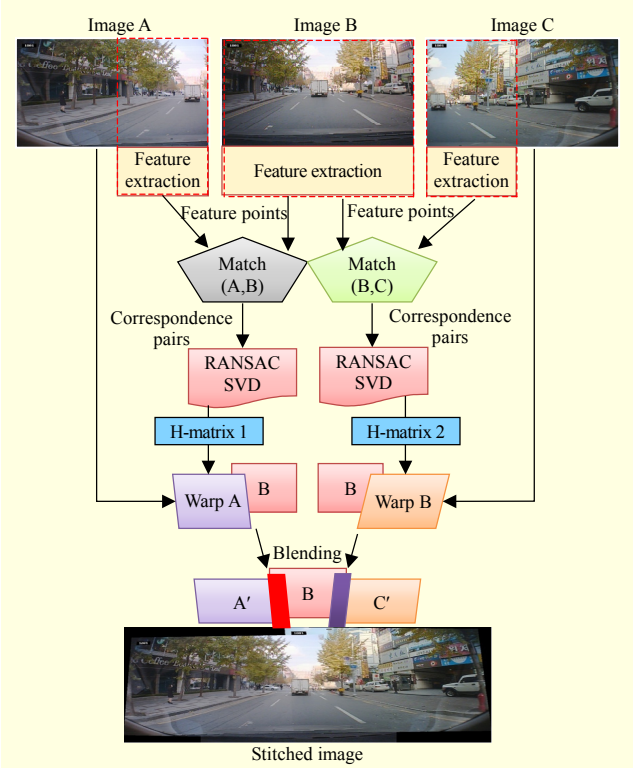


Fig. 5. Image stitching result by C-modeling program.

Table 1. SW operation time for H-matrix generation by 32-bit EISC processor for 200 MHz clock.

Task	Time	Features
Feature extraction for left image	10 s	764 features
Feature extraction for right image	9 s	484 features
Correspondence matching	2.3 s	196 pairs
RANSAC/SVD	217 ms	188 inliers
Total	21.517 s	N/A

Figure 5 shows the results of a stitched image from three VGA input images, and the method described above is used to make an H-matrix. In addition, Table 1 shows the SW operation time for the generation of an H-matrix using a 32-bit EISC process for a 200 MHz clock.

IV. Proposed HW Architecture for Real-Time Failure Detection

We designed the image stitching engine as shown in Fig. 6. The engine consists of a warping module, blending module, and stitching control module. The warping module warps each input image frame according to the H-matrix generated by the SW at the initial frame. For efficient communications with vision SoCs, the proposed engine supports an AXI interface.

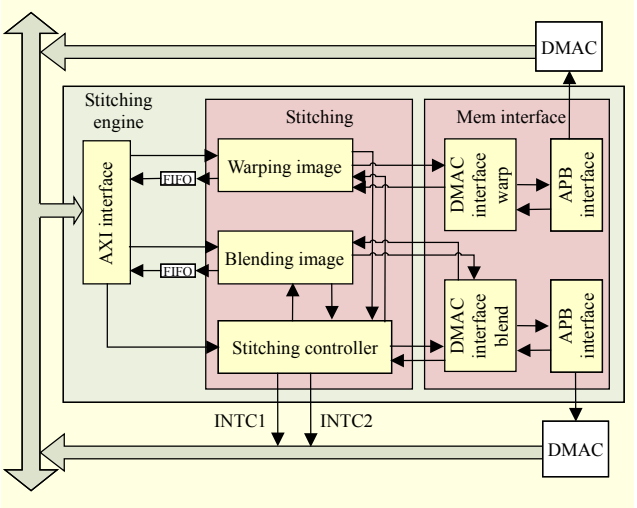


Fig. 6. HW block diagram of image stitching engine.

1. Fast Linear Warping Method and its Efficient HW Architecture

The homography matrix $\mathbf{H}_{ab}$ is a 3 × 3 matrix representing the relationship between pixel coordinates from two planes (*A* and *B*) taken from an input image (see Fig. 7). In the figure, *A* and *B* are the original and warped images of the input image, respectively, and $\mathbf{p}_a$ and $\mathbf{p}_b$ are a given pair of pixel coordinates, one from each respective plane. Pixel coordinates within the warped image can be calculated using matrix multiplications, as in (8) below, since we assumed an affine transform in Section II.

$$\mathbf{p}_b = \mathbf{H}_{ab}\mathbf{p}_a, \qquad \mathbf{p}_a = \mathbf{H}_{ab}^{-1}\mathbf{p}_b, \tag{7}$$

where

$$\mathbf{p}_a = \begin{bmatrix} x_a \\ y_a \\ z_a \end{bmatrix}, \quad \mathbf{p}_b = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}, \quad \mathbf{H}_{ab} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{31} & h_{31} \end{bmatrix},$$
$$\mathbf{p}_a = \begin{bmatrix} x_a \\ y_a \\ 1 \end{bmatrix}, \quad \mathbf{p}_b = \begin{bmatrix} x_b \\ y_b \\ 1 \end{bmatrix}, \quad \mathbf{H}_{ab} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

A transformed *x*- and *y*-coordinate, x_b and y_b, respectively, can be calculated from (8) and (9), and is given as (10) below.

$$\begin{bmatrix} x_b \\ y_b \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_a \\ y_a \\ 1 \end{bmatrix}, \tag{9}$$

$$\begin{aligned} x_b &= h_{11}x_a + h_{12}y_a + h_{13}, \\ y_b &= h_{21}x_a + h_{22}y_a + h_{23}. \end{aligned} \tag{10}$$

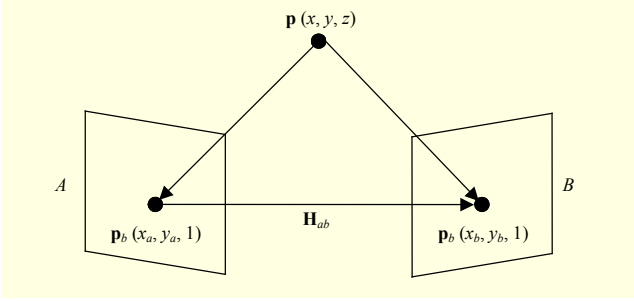


Fig. 7. Planar perspective projection relating homography and coordinate system transformation between two images.

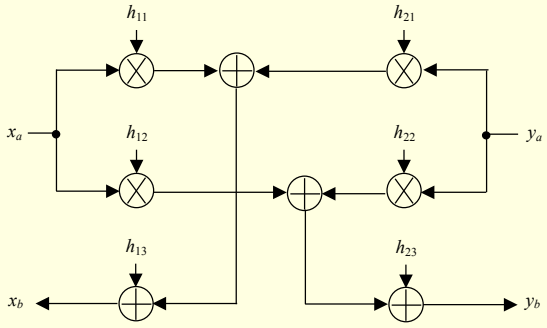


Fig. 8. HW architecture of address generator based on multiplier and adder for warping images.

To obtain a transformed position of an original pixel, four multiplications and four additions are needed, as shown in (10) and Fig. 8, which shows the general HW architecture of an address generator for warping images.

Although transforming a single pixel is not a difficult operation, the total number of multiplications/additions per frame is not a negligible quantity, since we should calculate a new pixel position for every pixel. When the 640 × 480 sized images are warped at every 1/30 s, the total number of multiplications per second exceeds 36 million since we should calculate, per frame, a new pixel position for every pixel [23].

$$\begin{aligned} x_{b,m,n} &= h_{11}x_{a,m} + h_{12}y_{a,n} + h_{13} \\ &= h_{11}\left(x_{a,m-1}+1\right) + h_{12}y_{a,n} + h_{13} \\ &= x_{b,m-1,n} + h_{11} \\ &= h_{11}x_{a,m} + h_{12}\left(y_{a,n-1}+1\right) + h_{13} \\ &= x_{b,m,n-1} + h_{12}, \\ y_{b,m,n} &= h_{21}x_{a,m} + h_{22}y_{a,n} + h_{23} \\ &= h_{21}\left(x_{a,m}+1\right) + h_{22}y_{a,n} + h_{23} \\ &= y_{b,m-1,n} + h_{21} \\ &= h_{21}x_{a,m} + h_{22}\left(y_{a,n-1}+1\right) + h_{23} \\ &= y_{b,m,n-1} + h_{22}. \end{aligned} \tag{11}$$

An transformed pixel position can be represented as ($x_{b,m,n}$, $y_{b,m,n}$), where m and n are integers denoting the x and y positions, respectively, and are ranged according to the image size. If the image size is $N \times M$, then m and n are ranged from 0 to $(N-1)$, and from 0 to $(M-1)$, respectively. Since m and n can be generated sequentially, we can obtain (11) from (10). As can be seen in (11), a transformed pixel position can be calculated by only two additions, whereas a conventional method needs four multiplications and additions. Thus, we can reduce the complexity of a transformed pixel point–generator based on a HW linear address generator as follows.

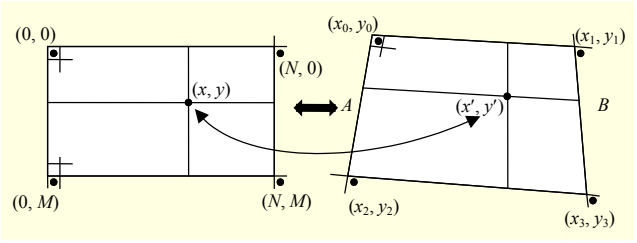


Fig. 9. Coordinate system for linear calculation.

We can calculate all coordinates of x and y through a linear operation without multiplication after generating only four transformed pixel positions, as shown in Fig. 9. Because we use a fixed H-matrix, the four transformed pixel positions are calculated only once for an initial state by an EISC processor. The linear equations for the transformation of a pixel position can be calculated as

$$\begin{aligned} A &= y_0\frac{(M-y)}{M} + y_2\frac{y}{M} = y_0 + (y_2-y_0)\frac{y}{M}, \\ B &= y_1\frac{(M-y)}{M} + y_3\frac{y}{M} = y_1 + (y_3-y_1)\frac{y}{M}, \\ y' &= A\frac{(N-x)}{N} + B\frac{x}{N} \\ &= A + (B-A)\frac{x}{N} \\ &= y_0 + (y_2-y_0)\frac{y}{M} + \left(y_1 + (y_3-y_1)\frac{y}{M} - y_0 - (y_2-y_0)\frac{y}{M}\right)\frac{x}{N} \\ &= y_0 + (y_1-y_0)\frac{x}{N} + (y_2-y_0)\frac{y}{M} + (y_3-y_2-y_1+y_0)\frac{xy}{NM}, \\ x' &= x_0 + (x_1-x_0)\frac{x}{N} + (x_2-x_0)\frac{y}{M} + (x_3-x_2-x_1+x_0)\frac{xy}{NM}, \\ T_{x0} &= x_0, \quad T_{x1} = (x_1-x_0)\frac{1}{N}, \quad T_{x2} = (x_2-x_0)\frac{1}{M}, \\ T_{x3} &= (x_3-x_2-x_1+x_0)\frac{1}{NM}, \\ T_{y0} &= y_0, \quad T_{y1} = (y_1-y_0)\frac{1}{N}, \quad T_{y2} = (y_2-y_0)\frac{1}{M}, \\ T_{y3} &= (y_3-y_2-y_1+y_0)\frac{1}{NM}. \end{aligned} \tag{12}$$

As shown in (12), we can calculate all transformed coordinates of x and y using a linear operation through eight constant

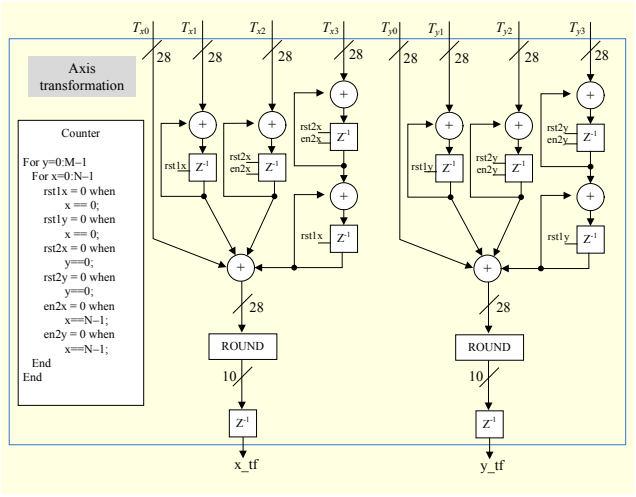


Fig. 10. Proposed HW architecture of address generator based on adder and register for fast linear warping method.

values, T_{x0}, T_{x1}, T_{x2}, T_{x3}, T_{y0}, T_{y1}, T_{y2}, and T_{y3}, generated from four initial transformed positions, such as (x_0, y_0), (x_1, y_1), (x_2, y_2), and (x_3, y_3), for example. The proposed HW circuit for generating the transformed coordinates of x and y is shown in Fig. 10 and is composed of ten adders, ten registers, two counters for (m, n), and two rounding operators without high-cost multipliers to reduce the computational complexity by 90% or more when compared with a conventional method.

2. Blending Algorithm

To make natural panoramic moving images, we blend each image with a *graph cut* [32] and alpha blending [33]. Whenever two registered images are overlapped, their differences are stored in a specified memory area to obtain a *cut-line*. According to the cut-line, which is decided by the software, alpha blending is then applied [23]. A cut-line decision algorithm can vary by designer. For designers who may want to use other algorithms, our engine also supports warped-only images for enhanced work using software. Whenever good algorithms are developed, they can be implemented using the MP in the vision SoC. The equation for alpha blending we adopted is shown below.

$$I_{\text{blend}} = \alpha I_{\text{left}} + (1-\alpha) I_{\text{right}}, \tag{13}$$

where I_{blend}, I_{left}, and I_{right} are the pixel values of blended, left, and right images, respectively, and α is a blending parameter that corresponds to the ratio of I_{left} over I_{right}. For example, the value of α ranges from 0 to 1, where 1 is used at the first pixel of the blended image on the left, and 0 is used at the last pixel of the blended image on the right. The blending algorithm computes the contribution of both images at each and every pixel and minimizes the effect of exposure variations [34].

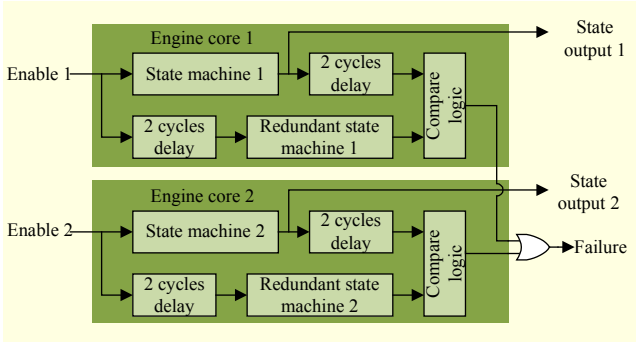


Fig. 11. Failure detection block based on rockstep.

3. Failure Detection

A standard for the functional safety of road vehicles, ISO26262, was recently published [27]. We adopted lockstep blocks to fulfill the requirements of this functional safety standard. A failure in the state machines in the image stitching engine may cause a problem in the SoC or system. Figure 11 shows a block diagram of the failure detection block, which is made up of a redundant state machine, two delay circuits, and a comparator based on a rockstep method for each engine core. The failure detection block of each engine compares the current states of the state machine and two-cycles-delay states by the lockstep block to detect any failures.

4. Dual Core–Based Architecture

A dual-core engine usually operates properly when stitching input images in parallel. When one of the dual-core engines is out of order, the other engine will operate normally instead of using parallel processing. Using a dual-core structure, we can obtain improvements in the functional safety and performance. A dual-core SW/HW image stitching engine, as shown in Fig. 12, is applied to stitch the input frames in parallel; thus, we can improve the performance by 70% or more as compared with a single-core operation (see Table 4).

V. Experimental Results

1. FPGA Implementation and Test Results

The proposed image stitching engine is controlled by an EISC processor with parameters such as the memory addresses used; the image size and format; and the blending area. In addition, this engine can be used for any image size, and it supports RGB, YCbCr 4:4:4, YCbCr 4:2:0, and YCbCr 4:2:2 image formats. The designed image stitching engine was verified on an FPGA board, as shown in Fig. 13. The specifications of the board and test results are summarized in Table 2.

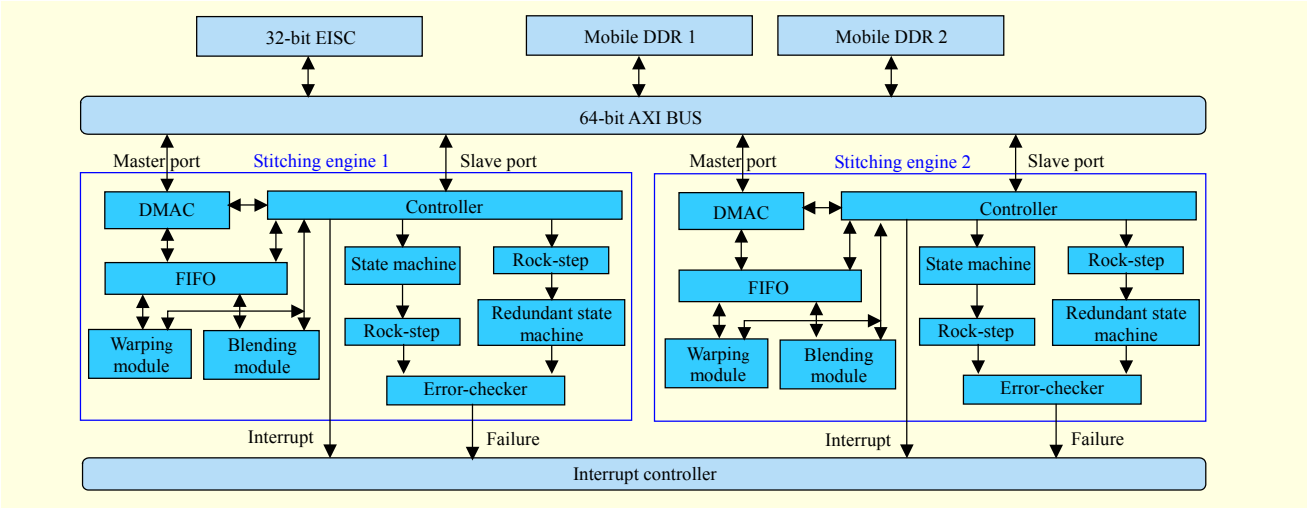


Fig. 12. Dual core–based architecture.

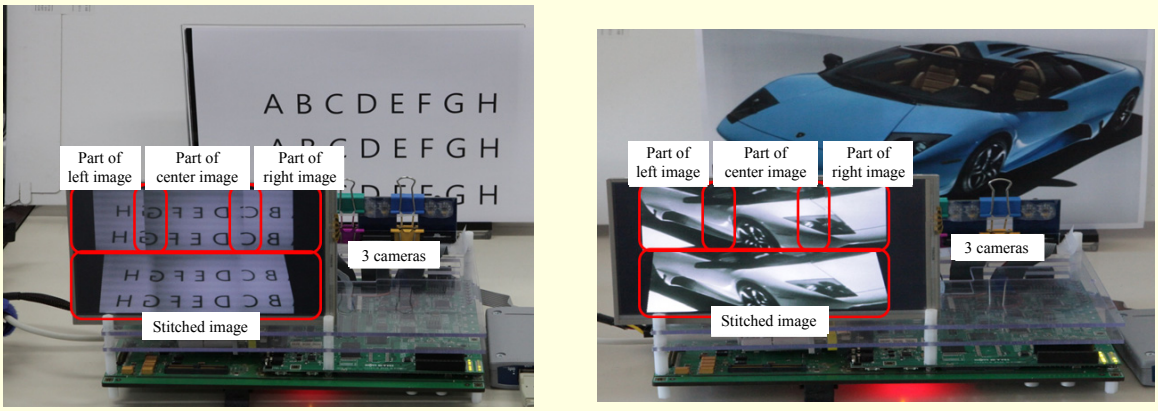


Fig. 13. Image stitching demonstrations using FPGA test board.

Table 2. Specifications of FPGA implementations and performance of image stitching.

Category	Features
Device	Altera Stratix IV EP4SGX530
Logic count	4,884 Combinational ALUTs 3,298 registers, 5,280 bits mem
Processor	32bit EISC from ADChips [26]
System bus	Full-matrix AXI (64 bit)
External memory	Mobile DDR
System clock	25 MHz
Performance	VGA 3 × 3 fps @ 25 MHz

Table 3. Synthesized results.

Category	Features	
Tool	Synopsys Design Compiler TM	
Process	65 nm (GF)	
Operating clock	Max 333 MHz	
Engine size	Core 1	245,220 μm^2 (127,514 gates)
	Core 2	245,104 μm^2 (127,454 gates)
Performance analysis	VGA 44 × 3 fps @ 200 MHz (single core)	

2. Single-Chip Implementation and Test Results

Table 3 shows the results of a gate-level synthesis of the image stitching engine and its performance analysis. We used Synopsys Design CompilerTM and a 65 nm Global Foundry process. We conducted place and route (PnR) and post-simulation processes with a 200 MHz clock-speed constraint for our design. According to our simulations after PnR, three YCbCr 4:2:0 formatted VGA images can be stitched at about a maximum of 44 fps for a 200 MHz main clock with a single-

Table 4. Comparison with other similar image stitching systems [38].

Systems	Test image resolution	Frame rate @ test image resolution	Test image resolution × fps	System clock (Mem. clock)	Implementation method	Cost (size)
Ladybug2 [35]	1,024 × 768 × 6	15 fps	70,778,880	N/A	SW, PC, video card	High
Ladybug3 [35]	1.6 k × 1.2 k × 6	6.5 fps	74,880,370	N/A	SW, PC, video card	High
FascinatE [36]	7 k × 2 k	25 fps	350,000,000	N/A	SW, PC, video card	Ultra high
Panoptic camera [37]	256 × 1,024	25 fps	6,553,600	212 MHz (SRAM, 212 MHz)	HW, 2 FPGA	Medium (2 × 35 mm × 35 mm, FPGA size)
Yuan Xu [38]	6 k × 720	15 fps	64,800,000	100 MHz (DDR 3,400 MHz)	HW, 1 FPGA	Low (31 mm × 31 mm, FPGA size)
This paper (single core)	640 × 480 × 3	44 fps	40,550,400	200 MHz (Mobile DDR, 100 MHz)	SW/HW, 1 ASIC	Ultra low (500 μm × 500 μm, stitching engine size)
This paper (dual core)	640 × 480 × 3	75 fps	69,120,000	200 MHz (Mobile DDR, 100 MHz)	SW/HW, 1 ASIC	Ultra low (2 × 500 μm × 500 μm, stitching engine size)

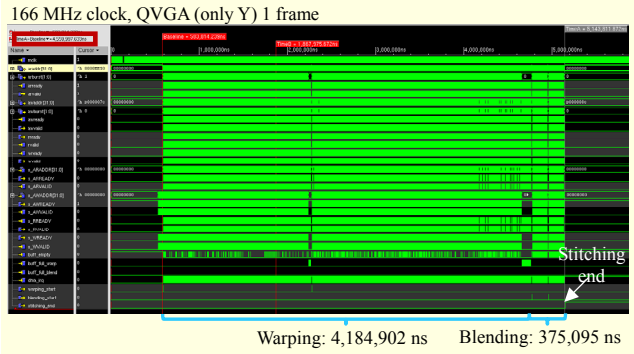


Fig. 14. Back-end simulation results in case of single core.

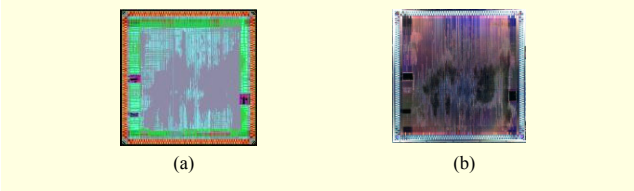


Fig. 15. Results of PnR and die photograph of SoC.

core operation without an LCD display. Figure 14 shows the back-end simulation results. Figure 15 shows the results of the PnR process and a die photograph of the SoC. Figures 16 and 17 show real-time image stitching demonstrations using an SoC test board.

3. Comparison with Other Systems

Table 4 shows the comparison among five similar image stitching systems. Ladybug2 and Ladybug3 are the systems from Grey Point [35]. Ladybug2 is a spherical video system

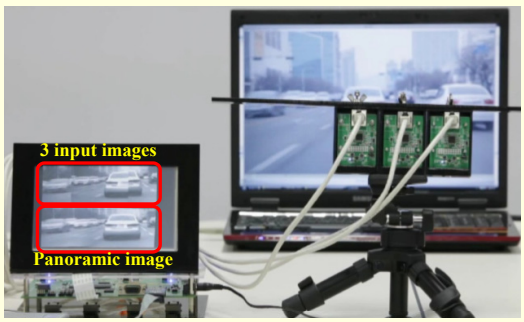


Fig. 16. Real-time image stitching demonstration using SoC test board and captured load images.

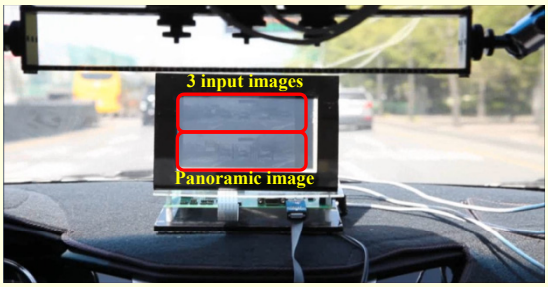


Fig. 17. Real-time image stitching demonstration with SoC test board in moving vehicle.

that can reach a resolution of 1,024 × 768 × 6 pixels at 15 fps. Ladybug3 has improved the resolution up to 1,600 × 1,200 × 6 pixels at 6.5 fps. They are high cost implementations based on a PC and its video card. A panoptic camera [37] can perform an overall resolution of 1,024 × 256 pixels at 25 fps. An EU-funded research project, FascinatE, [36] is performed on six high-definition (HD) cameras, resulting in an overall resolution

of 7,000 × 2,000 pixels at 25 fps. It is a piece of high-end broadcasting equipment based on a Cine Card PCI-card that supports up to 14 projectors per PC — the cost of which is extremely high. The compact version of the FacinatE system weighs about 16 kg. Yuan Xu's system [38] can provide a resolution of 6,000 × 720 pixels at 15 fps with one low-cost FPGA; the size of the FPGA is 31 mm × 31 mm, and the weight of the system is about 700 g.

The proposed single-core stitching engine can provide panoramic images from three VGA images at about a maximum of 44 fps for a 200 MHz clock with the smallest size of 500 μm × 500 μm. The dual-core stitching engine is applied to stitching input frames in parallel so we can improve the performance by 70% or more as compared with a single-core operation. In comparison with other systems, the proposed SW/HW stitching engine is of ultra-low cost and size, as well as being a high-performing and portable real-time system.

VI. Conclusion

In this paper, we proposed an efficient architecture of a real-time image stitching engine for a vision SoC of a motor vehicle. We adopt panoramic images from multiple telegraphic cameras to enlarge the detection distance and area for safety. We designed the engine using SW and HW based on a fixed homography for real-time processing within the environment of a moving vehicle. The proposed HW engine is based on a linear transform of the pixel positions to reduce the hardware complexity by more than 90%. In addition, using a dual-core structure, we can obtain improvements in functional safety and performance. The dual image stitching engines are fabricated in an SoC with 254,968 gate counts using Global Foundry's 65 nm CMOS process. The single engine can make panoramic images from three YCbCr 4:2:0 formatted VGA images at about a maximum of 44 fps for a 200 MHz clock without an LCD display. The engine performs well using an AXI-BUS-based vision SoC in real time. We expect that the proposed engine can be applied not only to driving assistance systems that have vision-based object detection ability and a heads-up display function, but also to image processing systems that have a panoramic view function, such as a digital camcorder or smartphone.

References

[1] R. Szeliski, "Image Alignment and Stitching: A Tutorial," Microsoft Research, Tech. Rep., MSR-TR-2004-92, Oct. 2004.

[2] R. Szeliski, "Video Mosaics for Virtual Environments," *IEEE Comput. Graph. Appl.*, vol. 16, no. 2, Mar. 1996, pp. 22–30.

[3] B.S. Kim, S.H. Lee, and N.I. Cho, "Real-Time Panorama Canvas of Natural Images," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, Nov. 2011, pp. 1961–1968.

[4] S. Mann and R.W. Picard, "Virtual Bellows: Constructing High Quality Stills from Video," *Proc. IEEE Int. Conf. Image Process.*, Austin, TX, USA, vol. 1, Nov. 13–16, 1994, pp. 363–367.

[5] S. Chen, "Quicktime VR: An Image-Based Approach to Virtual Environment Navigation," *Proc. SIGGRAPH*, New York, USA, Aug. 1995, pp. 29–38.

[6] Y. Xu, X. Li, and Y. Tian, "Automatic Panorama Mosaicing with High Distorted Fisheye Images," *Proc. Int. Conf. Natural Comput.*, Yantai, China, Aug. 10–12, 2010, pp. 3286–3290.

[7] S.K. Nayar, "Catadioptric Omnidirectional Camera," *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, San Juan, Puerto Rico, June 17–19, 1997, pp. 482–488.

[8] T. Kawanishi et al., "Generation of High-Resolution Stereo Panoramic Images by Omnidirectional Sensor Using Hexagonal Pyramidal Mirrors," *Proc. Int. Conf. Pattern Recogn.*, Brisbane, Australia, vol. 1, Aug. 16–20, 1998, pp. 485–489.

[9] S. Peleg and M. Ben-Ezra, "Stereo Panorama with a Single Camera," *IEEE Conf. Comput. Vis. Pattern Recogn.*, Fort Collins, CO, USA, vol. 1, June 23–25, 1999, pp. 395–401.

[10] F. Huang et al., "Animated Panorama from a Panning Video Sequence," *Int. Conf. Image Vis. Comput. New Zealand*, Queenstown, New Zealand, Nov. 8–9, 2010, pp. 1–8.

[11] S. Peleg, M. Ben-Ezra, and Y. Pritch, "Omnistereo: Panoramic Stereo Imaging," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, Mar. 2001, pp. 279–290.

[12] A. Ahmed et al., "Geometric Correction for Uneven Quadric Projection Surfaces Using Recursive Subdivision of Bézier Patches," *ETRI J.*, vol. 35, no. 6, Dec. 2013, pp. 1115–1125.

[13] S.J. Ha et al., "Panorama Mosaic Optimization for Mobile Camera Systems," *IEEE Trans. Consum. Electron.*, vol. 53, no. 4, Nov. 2007, pp. 1217–1225.

[14] S.J. Ha et al., "Embedded Panoramic Mosaic System Using Auto-Shot Interface," *IEEE Trans. Consum. Electron.*, vol. 54, no. 1, Feb. 2008, pp. 16–24.

[15] J.M. Seok and Y. Lee, "Visual-Attention-Aware Progressive RoI Trick Mode Streaming in Interactive Panoramic Video Service," *ETRI J.*, vol. 36, no. 2, Apr. 2014, pp. 253–263.

[16] D. Wagner et al., "Real-Time Panoramic Mapping and Tracking on Mobile Phones," *IEEE Conf. Virtual Reality*, Waltham, MA, USA, Mar. 20–24, 2010, pp. 211–218.

[17] M. Brown and D.G. Lowe, "Automatic Panoramic Image Stitching Using Invariant Features," *Int. J. Comput. Vis.*, vol. 74, no. 1, Aug. 2007, pp. 59–73.

[18] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, 2004, pp. 91–110.

[19] H. Bay et al., "SURF: Speeded-Up Robust Features," *Comput. Vis. Image Understanding*, vol. 110, no. 3, 2008, pp. 346–359.

[20] A. Shashua et al., EyeQ, Mobileye, 2010. Accessed Aug. 15,

2013. http://www.mobileye.com/technology/processing-platforms/eyeq/
[21] A. Shashua et al., EyeQ2, Mobileye, 2010. Accessed Aug. 15, 2013. http://www.mobileye.com/technology/processing-platforms/eyeq2/
[22] G.P. Stein, Y. Gdalyahu, and A. Shashua, "Stereo-Assist: Top-Down Stereo for Driver Assistance Systems," *IEEE. Conf. Intell. Vehicles Symp*., San Diego, CA, USA, 2010, pp. 723–730.
[23] J.-H. Suk et al., "An Efficient Architecture of Image Stitching Engine for Vis. SoC," *Proc. Workshop Image Processing Image Understanding*, Jeju, Rep. of Korea, Feb. 15–17, 2012, Index O-6.
[24] Homography, Wikipedia, 2013. Accessed May 30, 2014. https://en.wikipedia.org/wiki/ Homography
[25] Image Stitching, Wikipedia, 2008. Accessed Aug. 30, 2013. https://en.wikipedia.org/wiki/Image_stitching
[26] International Organization for Standardization (ISO), *ISO 26262 Road Vehicle – Functional Safety* Geneva, Switzerland: ISO, 2011.
[27] K.H. Kwon et al., EISC, Advanced Digital Chips Inc., 2012. Accessed June 15. http://www.adc.co.kr/technology/eisc/eisc.php
[28] H.S. Park et al., "In-Vehicle AR-HUD System to Provide Driving-Safety Information," *ETRI J*., vol. 35, no. 6, Dec. 2013, pp. 1038–1047.
[29] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. ACM*, vol. 24, no. 6, June 1981, pp. 381–395.
[30] M. Zuliani, "RANSAC for Dummies," Matlab draft, Jan. 2012.
[31] K. Baker, "Singular Value Decomposition Tutorial," tutorial paper, Mar. 2005.
[32] Y.Y. Boykov and M. Jolly, "Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images," *Proc. IEEE Int. Conf. Comput. Vis*., Vancouver, Canada, vol. 1, July 7–14, 2001, pp. 105–112.
[33] Alpha Compositing, Wikipedia, 2006. Accessed Aug. 30, 2013. https://en.wikipedia.org/wiki/Alpha_compositing#Alpha_blendin
[34] S. Ali and M. Hussain, "Panoramic Image Construction Using Feature Based Registration Methods," *Proc. Int. Multitopic Conf*., Islamabad, Pakistan, Dec. 13–15, 2012, pp. 209–214.
[35] Point Grey Research Inc. (2012, Dec. 7), Spherical Video System Ladybug2 and Ladybug3. Available: http://www.ptgrey.com
[36] O. Schreer et al., "Ultra-high-Resolution Panoramic Imaging for Format-Agnostic Video Production," *Proc. IEEE*, vol. 101, no. 1, Jan. 2013, pp. 99–114.
[37] A. Akin et al., "Enhanced Omnidirectional Image Reconstruction Algorithm and its Real-Time Hardware," *Euromicro Conf. Digital Syst. Des*., Izmir, Turkey, Sept. 5–8, 2012, pp. 907–914.
[38] Y. Xu et al., "High-Speed Simultaneous Image Distortion Correction Transformations for a Multicamera Cylindrical Panorama Real-Time Video System Using FPGA," *IEEE Trans. Circuits Syst. Video Technol*., vol. 24, no. 6, June 2014, pp. 1061–1069.

Jung-Hee Suk received his BS, MS, and PhD degrees in electronics engineering from Kyungpook National University, Daegu, Rep. of Korea, in 2001, 2003, and 2007, respectively. Since 2007, he has been with ETRI, where he is now a senior researcher. His doctoral research involved the H.264/AVC video codec algorithm. His current research interests include pattern recognition algorithms for smart devices, efficient architecture of SoC, multimedia codecs, motor control systems, and multiple outputs control algorithms of power management ICs.

Chun-Gi Lyuh received his BS degree in computer engineering from Kyungpook National University, Daegu, Rep. of Korea, in 1998 and his MS and PhD degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea, in 2000 and 2004, respectively. Since 2004, he has been with ETRI, where he is now a principle member of the research staff. His current research interests include vision SoC platforms for intelligent vehicles and digital integrated-circuit design.

Sanghoon Yoon received his BS, MS, and PhD degrees in electronic engineering from Hanyang University, Seoul, Rep. of Korea, in 1996, 1998, and 2008, respectively. Currently, he is a senior researcher of the research staff at the Korea Electronics Technology Institute, Seongnam, Rep. of Korea. His main research interests include vision and communication SoC platforms for intelligent vehicles and digital integrated-circuit design.

Tae Moon Roh received his BS, MS, and PhD degrees in electrical engineering & computer science from Kyungpook National University, Daegu, Rep. of Korea, in 1984, 1986, and 1998, respectively. Since 1988, he has been with ETRI, where he is now a principal researcher. He was engaged in the research of developing process technology for digital/analog CMOS IC and power IC, improving reliability of ultra-thin gate oxide, and evaluating hot carrier effects of MOSFETs. He studied low power digital circuits and multimedia SoCs with reconfigurable processors, vision SoC platforms for intelligent vehicles, and readout integrated circuits for ubiquitous sensor networks. His current interests are SiC power devices for hybrid electric vehicles and intelligent sensors for bio-health monitoring and health care systems.