Time Multiplexed VLSI Architecture for Real-Time Barrel Distortion Correction in Video-Endoscopic Images

P. G. S. Mythili[1] S. Vamsee Krishna[2]
[1]M. Tech(VLSI) [2]Associate Professor
[1, 2]SIETK, Puttur.

Abstract—A low-cost very large scale integration (VLSI) implementation of real-time correction of barrel distortion for video- endoscopic images is presented in this paper. The correcting mathematical model is based on least-squares estimation. To decrease the computing complexity, we use an odd-order polynomial to approximate the back-mapping expansion polynomial. By algebraic transformation, the approximated polynomial becomes a monomial form which can be solved by Hornor's algorithm. With the iterative characteristic of Hornor's algorithm, the hardware cost and memory requirement can be conserved by time multiplexed design. In addition, a simplified architecture of the linear interpolation is used to reduce more computing resource and silicon area. The VLSI architecture of this paper contains 13.9-K gates by using a 0.18 μm CMOS process. Compared with some existing distortion correction techniques, this paper reduces at least 69% hardware cost and 75% memory requirement.

Key Words: Barrel distortion correction, Hornor's algorithm, least-squares estimation, time multiplexed, video- endoscopic image.

I. INTRODUCTION

VIDEO-ENDOSCOPY plays an important role in clinical applications, such as gastrointestinal tract, respiratory tract, urology, gynecology, and surgical procedures. The cam- era with wide-angle lens (fish-eye lens) of clinical endoscope can capture a larger field of interior body in a single image. Such a system enhances the capability of images. However, its outcome shows barrel-type spatial distortion which causes the image regions farther from the distortion center to be compressed more than those near the center by a non-linear distortion. Thus, the outer regions of the image are observed smaller than their actual size.

Many researchers have proposed various mathematical models and algorithms to correct the endoscopic and wide-angle camera distorted images [1]–[10]. These studies provide novel technologies to correct the distorted images by some efficient algorithms. However, the proposed software solutions do not meet the high speed demand for real-time video- endoscopic applications. For example, a biomedical video-endoscopic real-time system is designed with a 1024 × 1024 wide-angle lens and it requires capturing 30 frames/s. By this specification, this system needs to correct more than 30
Mega (1024 × 1024 × 30) distorted pixels per second. Since each correcting procedure for each distorted pixel consumes more than a few instruction cycles, the software solutions are hard to meet the requirement of real-time video-endoscopic applications.

Several studies concerning very large scale integration (VLSI) architectures of real-time barrel distortion correction have been presented recently. Asari presented an efficient VLSI architecture to correct wide-angle camera images by mapping the algorithmic steps onto a linear array [11]. Pipelined architecture is presented in [12]. In this literature, high performance pipelined architecture for correction of barrel distortion in wide-angle camera images is proposed. A pipelined coordinate rotation digital computer (CORDIC) is used in this paper for improving the efficiency of coordinate transformation. In addition, a low-cost high-speed 21-stage pipelined VLSI architecture for barrel distortion correction is presented in [13]. Angle calculation elimination and odd-order back-mapping polynomial are used to improve performance and reduce hardware cost. The methods of the previously pro- posed technologies were based on the least-square estimation scheme presented by Asari [3].

In video-endoscopic applications, the barrel distortion correction circuit is integrated into medical instrument. Thus, since lower hardware cost and high performance become important parameters in most biomedical applications, prudent implementation should be considered. In this paper, we present a low-cost VLSI architecture for real-time distortion correction circuit. The correction mathematical model is based on least- squares estimation method [3], [11]. After analyzing the correcting procedure, we found the most computing resources are concentrated on coordinate transformation, back-mapping, and linear interpolation. To lower the computing resources, firstly we used an odd-order polynomial to approximate the back-mapping expansion polynomial. After which, the odd-order back-mapping and polar to Cartesian coordinate transformation can be combined into a new polynomial. Since the new combined polynomial is based on the vector magnitude square, a square root and an arc-tangent calculation can be efficiently removed.

To reduce more hardware, we used algebraic transformation to replace all the vector magnitude square by a new variable. It makes the new combined polynomial becomes a monomial form. Therefore, Hornor's algorithm [16] is able to efficiently evaluate the results of back-mapping and polar to Cartesian coordinate transformation. This approach not only greatly decreases the calculation complexity of back-mapping but also provide a flexible architecture for different designs by time multiplexed technique [17], [18]. To reduce more hardware costs and power consumption, we also implemented a low cost linear interpolation circuit by algebraic manipulation technique. It greatly eliminates the number of multipliers from eight to three. In this paper, we created a novel low- cost, low-power, and low memory requirement four-cycle time multiplexed distortion correction circuit for

biomedical video- endoscopic or wide-angle camera applications.

The organization of this paper is presented as follows. Section II introduces the proposed less complex distortion correction methodology. Section III describes the details of the proposed VLSI architecture. Section I V shows the comparisons and experimental results. And finally, in Section V, the conclusion is presented.

II. PROPOSED LOW-COMPLEXITY DISTORTION CORRECTION TECHNIQUE

In this section, the proposed low-complexity distortion correction technique based on least-square estimation method is presented. In the beginning of this section, a transformation mapping from Cartesian coordinate to polar coordinate for all pixels is introduced. Then, a simplifying procedure is introduced to decrease the computing resource of back- mapping and polar to Cartesian coordinate steps by eliminating the angle θ *and* reducing the square root operation for ρ. Then, Hornor's algorithm is adopted to reduce more computing resource of the simplified one. Finally, a basic algebraic manipulation is used to decrease the arithmetic resource of the linear interpolation.

A. Cartesian to Polar Coordinate Transformation

The first step of the proposed distortion correction technique is transforming all pixels in the distorted image space (DIS) onto the corrected image space (CIS). The distortion center is (u_c, v_c) in DIS and the correction center is (u_c, v_c) in CIS. In DIS, (u,v) is the Cartesian coordinate and (ρ, θ) is the polar coordinate. The distance ρ from distortion center (u_c, v_c) to an image pixel (u,v) and the angle θ between the pixel and distortion center are given by

$$\theta = \arctan\left(\frac{v-v_c}{u-u_c}\right) \quad (2.1)$$

Each image pixel (u, v) in DIS can be mapped into a new location (u, v) in CIS. The corresponding distance ρ and angle θ from the correction center (u_c, v_c) to (u, v) can be obtained by

$$\rho = \sqrt{(u-u_c)^2 + (v-v_c)^2} \quad (2.2)$$

$$\theta = \arctan\left(\frac{v-v_c}{u-u_c}\right) \quad (2.3)$$

The corresponding relationship between ρ and ρ can be defined by an expansion polynomial of degree N as

$$\rho = {}_{n-1}^{N}a_n(\rho)^n \quad (2.4)$$

$$\theta = \theta \quad (2.5)$$

where a_n is expansion coefficient which can be obtained by the least-squares estimation method [3]. Since it is assumed that the distortion is purely radial about the distortion center, the angle θ is the same as θ. The new location (u, v) in CIS can be calculated as

$$u = u_c + \rho cos\theta \quad (2.6)$$

$$v = v_c + \rho sin\theta \quad (2.7)$$

B. Back Mapping Procedure and Polar to Cartesian Coordinate Transformation

After transforming the image pixels from Cartesian coordinate to polar coordinate, the back-mapping module maps the pixel (ρ, θ) in CIS into the corresponding location (ρ, θ) in DIS. As presented in [3] and [12], the back-mapping procedure can be defined by a back-mapping expansion polynomial of degree N as

$$\rho = {}_{n-1}^{N}b_n \cdot \rho^n \quad (2.8)$$

$$\theta = \theta \quad (2.9)$$

where b_n is the back-mapping coefficient which can be obtained by the least-squares estimation method. The angles θ and θ are the same since the distortion is assumed to be purely radial about the distortion center.

C. Polynomial Approximating Analysis

The back-mapping expansion polynomial can be approximated to odd-order or even-order polynomial by (2.8) as

$$\rho = co_0\rho + co_1\rho^3 + co_2\rho^5 + co_3\rho^7 \ldots\ldots \quad (2.10)$$

$$\rho = ce_0\rho^2 + ce_1\rho^4 + ce_2\rho^6 + ce_3\rho^8 \ldots\ldots \quad (2.11)$$

Where $co_0, co_1, co_2, co_3 \ldots\ldots$ or $ce_0, ce_1, ce_2, ce_3 \ldots\ldots$ are odd order or even order coefficients of the odd-order or even-order back-mapping polynomial. To be able to analyze the qualities of the odd-order and even-order approximating polynomials, ten sample images, some of which were used as test images in previous papers dealing with image processing, were selected as a test set.

	Odd-Order		Even-Order	
Test Image	PSNR	Approximated Rate (%)	PSNR	Approximated Rate (%)
Lena	23.84	97.49	10.79	59.28
Peppers	22.21	95.75	10.22	51.11
Airplane	23.20	97.52	10.24	66.77
Mandrill	24.19	98.14	9.60	55.27
Girl	25.25	97.09	10.56	50.53
Cameraman	24.97	96.75	10.73	49.56
Pirate	21.76	97.29	10.79	75.23
Sailboat	22.79	96.57	10.14	56.04
Splash	23.38	96.27	11.38	61.00
House	25.98	98.45	10.14	60.56
Average	23.76	97.13	10.46	58.53

Table. 1: Comparison of Approximated Rates of Odd-Order and Even-Order Polynomials with Back-Mapping Expansion Polynomial

To obtain the qualities of the back mapping approximating by odd-order and even-order, the PSNR values and approximated rates are used to evaluate. First, the ten test images were distortion corrected by the back mapping expansion polynomial, as shown in (5a), and the obtained result images are stored as golden patterns. After which, we used odd-order and even-order polynomials, as shown in (6a) and (6b), to correct the ten test images and then stored the result images as comparison patterns. Finally, the results can be obtained by computing the PSNR values and approximated rates between the comparison and golden patterns. Table I lists the PSNR values and the

percentages of approximated rates of the ten testing images by odd-order and even-order polynomials with the back mapping expansion polynomial. The results show that the odd-order polynomial achieves 97.13% average approximation to the back mapping expansion polynomial, which is much better than even-order one by 58.53%.

D. Simplified Back-Mapping Procedure

As mentioned above, the odd-order polynomial is high-approximation to the back mapping expansion polynomial. According to previous techniques [13]–[15], the back-mapping expansion polynomial can be approximated to odd-order polynomial as

$$\rho = c_0\rho + c_1\rho^3 + c_2\rho^5 + c_3\rho^7 \dots\dots \quad (2.12)$$

where c_0, c_1, c_2, c_3 ... are back-mapping coefficients of the odd-order back-mapping polynomial. In fact, the ρ is the square root of the sum of $(u - u_c)^2$ and $(v - v_c)^2$ as shown in (2.2). It cannot be calculated by simple arithmetic operations. In literature [12], the CORDIC module is implemented with huge hardware cost to obtain the value of ρ.

There are two steps to reduce the computing resources of back-mapping procedure. The first is eliminating the calculation of θ [13]. According to (2.9), the θ and θ are the same.

Thus, the $\cos\theta$ and $\sin\theta$ can be obtained as

$$\cos\theta = \cos\theta = \frac{u-u_c}{\rho} \quad (2.13)$$

$$\sin\theta = \sin\theta = \frac{v-v_c}{\rho} \quad (2.14)$$

Based on the odd-polynomial technique by (7) and eliminating the calculation of angle θ by (2.13), (2.14), the back-mapping and polar to Cartesian coordinate steps can be simplified as

$$u = u_c + \rho \cdot \frac{u-u_c}{\rho}$$

$$u = u_c + (c_0\rho + c_1\rho^3 + c_2\rho^5 + c_3\rho^7 \dots\dots) \cdot \frac{u-u_c}{\rho}$$

$$u = u_c + (c_0 + c_1\rho^2 + c_2\rho^4 + c_3\rho^6 \dots\dots) \cdot (u - u_c) \quad (2.15)$$

$$v = v_c + \rho \cdot \frac{v-v_c}{\rho}$$

$$v = v_c + (c_0\rho + c_1\rho^3 + c_2\rho^5 + c_3\rho^7 \dots\dots) \cdot \frac{v-v_c}{\rho}$$

$$v = v_c + (c_0 + c_1\rho^2 + c_2\rho^4 + c_3\rho^6 \dots\dots) \cdot (v - v_c) \quad (2.16)$$

After the simplification process, the u and v can be obtained by simple arithmetic operations without the need of square root and the value of ρ^2 can be calculated by

$$\rho^2 = (u - u_c)^2 + (v - v_c)^2 \quad (2.17)$$

E. Hornor's Algorithm

The purpose of Horner's algorithm [16] is to efficiently evaluate the polynomials in monomial form. For example, a typical polynomial equation is represented as

$$f(x) = \sum_{i=0}^{n} c_i \cdot x^i = c_0 + c_1x + c_2x^2 + \dots + c_nx^n \quad (2.18)$$

where c_1, c_2, c_3, ... are real numbers, and the polynomial is necessary to be evaluated at the appointed value of $x = x_0$. In usual method, to evaluate the polynomial $f(x_0)$ is to evaluate

$$f(x) = c_o + x\left(c_1 + x\left(c_2 + x(c_3 + \dots + x(c_{n-1} + c_nx))\right)\right) \quad (2.19)$$

The purpose of Horner's algorithm is to evaluate the polynomial at a specific value of $x = x_0$. Thus, we defined a new sequence of constants set a_n as

$$\begin{aligned} a_n &= c_n \\ a_{n-1} &= c_{n-1} + a_n x_0 \\ a_{n-2} &= c_{n-2} + a_{n-1} x_0 \\ &\vdots \\ &\vdots \\ a_0 &= c_0 + a_1 x_0 \end{aligned}$$

To replace x_0 by x and iteratively substitute constants set a_n into (12). The polynomial can be represented as

$$\begin{aligned} f(x) &= c_o + x_0\left(c_1 + x_0\left(c_2 + x_0(c_3 + \dots + x_0(c_{n-1} + a_nx_0))\right)\right) \\ &= c_o + x_0\left(c_1 + x_0\left(c_2 + x_0(c_3 + \dots + x_0(a_{n-1}))\right)\right) \\ &\vdots \\ &\vdots \\ &= c_0 + x_0(a_1) \\ &= a_0 \end{aligned}$$

As described above, the polynomial can be rewritten by an iteration function. The function f^n which is the nth iteration of f can be represented as $f \cdot f^{n-1}$ and it is defined as

$$f^n = f \cdot f^{n-1} \quad (2.20)$$

To rewrite the polynomial as iteration function, the evaluation of the polynomial at the appointed value of $x = x_0$ can be represented as

$$f(x_0) = (c_{n-1} + c_n \cdot x_0 \cdot f^{n-1}(x_0) \quad (2.21)$$

The purpose of evaluating the polynomial is to compute $f(x_0)$ where x_0 is a constant. The Hornor's algorithm [16] supports an efficient method and it works as follows.

– *Hornor's Algorithm*

Step 1. Set $u \leftarrow n$ (where n is the degree of the polynomial)
Step 2. Set Result $\leftarrow C_n$.
Step 3. If $u = 0$ stop. Answer is Result.
Step 4. Compute Result $\leftarrow$ Result $\times x_0 + C_{u-1}$.
Step 5. $u \leftarrow u - 1$.
Step 6. Go to step 3.

By Hornor's algorithm, the computing complexity of the evaluating polynomial can be efficiently decreased. The complexity of back-mapping and polar to Cartesian coordinate steps can be reduced in the same way. Equations (2.15) and (2.16) can be rewritten as

$$u = u_c + \left(c_{n-1} + c_n\rho^2 \cdot f^{n-1}(\rho^2)\right) \times (u - u_c) \quad (2.22)$$

$$v = v_c + \left(c_{n-1} + c_n\rho^2 \cdot f^{n-1}(\rho^2)\right) \times (v - v_c) \quad (2.23)$$

where c_1, c_2, c_3, ... c_{n-1}, c_n are the combined mapping coefficients of the polynomial. The computing resource of the back-mapping and polar to Cartesian coordinate steps can be efficiently decreased by only computing ρ^2. The most complexity of computing the power values of ρ^2 (ρ^4, ρ^6, ρ^8 ...) can be efficiently reduced by Hornor's algorithm.

	Step 1: Cartesian to Polar Coordinate	Step 2: Back-Mapping	Step 3: Polar to Cartesian Coordinate	Step 4: Linear Interpolation	Computing Resource Per Pixel
[12]	$\rho'=\sqrt{(u'-u'_c)^2+(v'-v'_c)^2}$ $\theta'=\arctan\left(\frac{v'-v'_c}{u'-u'_c}\right)$	$\rho'=\sum_{n=1}^{N} b_n\rho^n$	$u'=u'_c+\rho'\cos\theta'$ $v'=v'_c+\rho'\sin\theta'$	$I(u,v)=I'(x,y)\times(1-dx)\times(1-dy)$ $+I'(x+1,y)\times dx\times(1-dy)$ $+I'(x,y+1)\times(1-dx)\times dy$ $+I'(x+1,y+1)\times dx\times dy$	1 square root 1 arctan 15 multiply 10 add
[13]	$\rho^2=(u-u_c)^2+(v-v_c)^2$ $u'=u'_c+(c_0+c_1\rho^2+c_2\rho^4+c_3\rho^6+c_4\rho^8)\times(u-u_c)$ $v'=v'_c+(c_0+c_1\rho^2+c_2\rho^4+c_3\rho^6+c_4\rho^8)\times(v-v_c)$			$I(u,v)=I'(x,y)\times(1-dx)\times(1-dy)$ $+I'(x+1,y)\times dx\times(1-dy)$ $+I'(x,y+1)\times(1-dx)\times dy$ $+I'(x+1,y+1)\times dx\times dy$	19 multiply 14 add
This paper	$\rho^2=(u-u_c)^2+(v-v_c)^2$ $u'=u'_c+(c_{n-1}+c_n\rho^2\circ f^{n-1}(\rho^2))\times(u-u_c)$ $v'=v'_c+(c_{n-1}+c_n\rho^2\circ f^{n-1}(\rho^2))\times(v-v_c)$			$I(u,v)=dx\times\{[I'(x+1,y+1)-I'(x+1,y)]\times dy$ $+I'(x+1,y)-(I'(x,y+I)-I'(x,y))\times dy$ $+I'(x,y)\}+(I'(x,y+1)-I'(x,y))\times dy$ $+I'(x,y)$	11 multiply 17 add

Table. 2: Comparison of Equations and Computing Resource with Previous Technologies [12], [13]

F. Simplified Linear Interpolation

In previous literatures, [12], [13], the linear interpolation is directly implemented by the linear interpolation equation as

$$I(u,v)=I(x,y)\times(1-dx)\times(1-dy)+I(x+1,y)\times I(1-dy)+I(x,y+1)\times(1-dx)\times dy+I(x+1,y+1)\cdot dx\cdot dy \quad (2.24)$$

where $I\ (x, y), I\ (x+1, y), I\ (x, y+1)$, and $I\ (x+1, y+1)$ are the intensity values of the four neighboring pixels at the location of $(x, y), (x + 1, y), (x, y + 1)$, and $(x + 1, y + 1)$. Hence, the computing resource of linear interpolation costs at least eight multiply and three add operations. It also means that eight multipliers and three adders are necessary when implementing a VLSI circuit.

For this reason, we used the algebraic manipulation to decrease the computing resource as well as hardware cost for the linear interpolation. Equation (17) shows the general arithmetic function of linear interpolation combined by (16). By the algebraic manipulation, the arithmetic function of the linear interpolation can be simplified as (17d). The simplification procedure is illustrated from (17a) through (17b) and (17c) to (17d). Besides, the function of $(I\ (x, y+1)-I\ (x, y))\times dy + I\ (x, y)$ is obtained in (17d) twice, which means that one calculation of this function will be reduced. As mentioned above, the computing resource of linear interpolation can be simplified as three multiplication and six addition operations. Thus, this simplified arithmetic function of the linear interpolation provides a low-cost and low-power element for VLSI circuit.

Table II lists the comparison of the equations and computing resource of previous techniques [12], [13] with this paper. Obviously, the complexity and computing resource of this paper are lower than others.

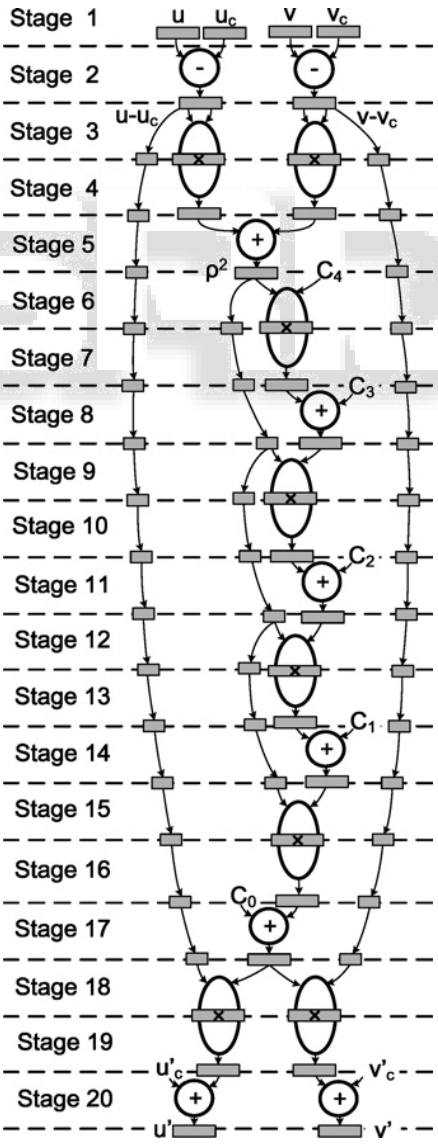

Fig. 2: Architecture of the combined mapping unit.

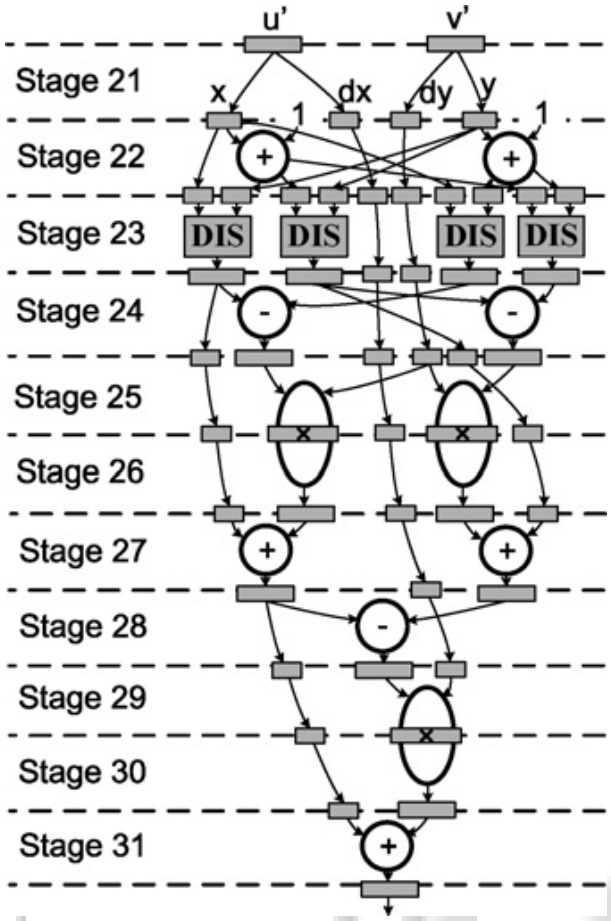

Fig. 3: Architecture of the simplified linear interpolation unit.

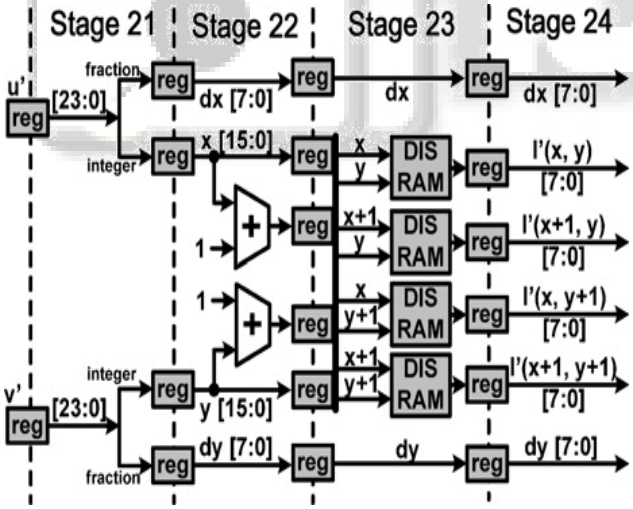

Fig. 4: Architecture of neighboring pixels reading and fractions obtaining circuit.

Fig. 4 shows the architecture of neighboring pixels reading and fractions obtaining circuit. The u and v are obtained from stage 20 of the combined mapping unit. The length of u and v is 24-bit, and the highest 16 of 24 bits are integer parts and the lowest 8 of 24 bits are fractional parts. Thus, the register value of x or y vector can be obtained directly by connecting wires with the highest 16-bit of the u or v register. The register value of dx or dy can be obtained from the lowest 8-bit of the u or v register in the same way. The size of each DIS RAM is the same as the size of the input image and the four intensity values of $I(x, y), I(x + 1, y), I(x, y + 1)$, and $I(x + 1, y + 1)$ can be read simultaneously from the four DIS RAMs.

G. Time Multiplexed Architectures

As mentioned in Section III, the back-mapping step demands the most computing resource of the distortion correcting procedure. Although the computing complexity of the back-mapping step is efficiently decreased by Hornor's algorithm, implementation of the distortion correcting circuit presents a limitation on hardware cost. As shown in (13a), (13b), the evaluation of polynomial can be described as an iteration function, which provides flexibility in implementing the back-mapping circuit with different architecture by time multiplexed technique [17], [18].

Fig. 5 shows three architectures of the back-mapping circuit. Since the back-mapping step can be described as an iteration function as shown in (18a) and (18b), the hardware architecture can be implemented as one-cycle, two-cycle, or four-cycle to complete the back-mapping procedure. Fig. 5(a) shows the one-cycle architecture of back-mapping circuit, the same as the stages from 6 to 17 of the combined mapping unit in Fig. 2, which costs four pipelined multipliers and four adders. It can obtain one back-mapping result during each clock cycle with 12-stage pipelined architecture. However, the hardware cost and memory requirement are quite a few. Fig. 5(b) and (c) shows the two-cycle and four-cycle architectures of back- mapping circuit, respectively. The hardware costs of these

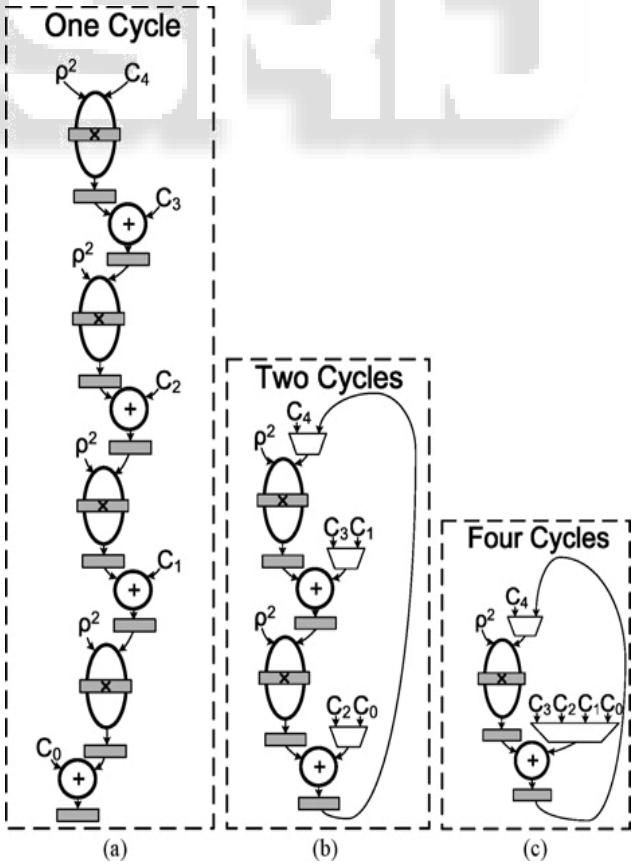

Fig. 5: Architectures of back-mapping circuit with (a) one-cycle, (b) two- cycles, (c) four-cycle to complete a back-mapping procedure.

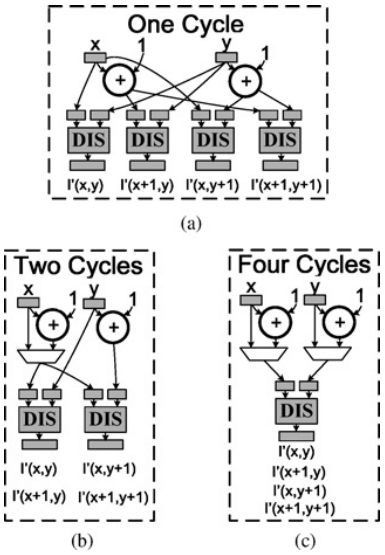

Fig. 6 : Architecture of memory reading circuit with (a) one-cycle, (b) two- cycle, (c) four-cycle to complete reading four neighboring pixel values.

architectures are less than that of one-cycle. Nevertheless, the execution time to get one back-mapping result needs two or four cycles. Fig. 6 shows the three architectures of the memory reading circuit for reading the identity values of four neighboring pixels of location (u , v). Fig. 6(a) shows the one-cycle architecture which can read four identity values during each clock cycle and provide a high-speed circuit for parallel.

	[12]	[13]	This Paper		
			One-	Two-	Four-
Square root	1	0	0	0	0
Arc-tangent	1	0	0	0	0
24 × 24	7	7	4	2	1
16 × 24	0	2	2	2	2
16 × 16	0	2	2	2	2
8 × 8 multiplier	8	8	3	3	3
Adder	10	14	17	15	14

Table. 4: Hardware Resources of various Distortion Correction Architectures

memory read. However, it demands four input image size of DIS RAMs.

Fig. 6(b) and (c) shows the two-cycle and four- cycle architectures of memory reading circuit. Obviously, the memory requirement of two-cycle or four-cycle architectures is only two or one of DIS RAMs.

	Distortion Correction	Distortion Correction	Distortion Correction
Back-mapping circuit cycle	One-cycle	Two-cycles	Four-cycles
Process	0.18 µm CMOS	0.18 µm CMOS	0.18 µm CMOS
Gate counts	28 622	15 895	13 917
Frequency	200 MHz	200 MHz	200 MHz
DIS RAM	4	2	1
Power	45.68 mW	24.97 mW	16.81 mW
Throughput	160 Mpixels/s	80 Mpixels/s	40 Mpixels/s
Quality support	2560 × 2048 30 frame/s	1920 × 1080 30 frame/s	1024 × 1024 30 frame/s
Normalized area	2.06	1.14	1

Table. 5: Comparison of Distortion Correcting Circuit with One-Cycle, Two-Cycle, and Four-Cycle Back-Mapping Designs

Table 4 shows detailed comparisons between the hardware resources of various architectures of distortion correction designs. According to the table, Ngo *et al*.'s [12] does not contain any 16 × 24 and 16 × 16 multipliers, which is less than other architectures. However, the hardware implementation of square root by CORDIC module in [12] and [19] consumes a huge hardware cost. Comparing the hardware resources of [12] and [13], and the three architectures of this paper, we can clearly find the contribution of each technique used in this paper. The results show that the algebraic manipulation of linear interpolation by this paper contributes reducing 62.5% (five of eight) 8 × 8 multipliers from [12] and [13]. Observing the number of 24 × 24 multipliers in Table IV, the one-cycle architecture of this paper consists of four 24×24 multipliers as compared to the 24 × 24 multipliers in [12] and [13] which have seven. As mentioned above, the Hornor's algorithm technique contributes decreasing 37.5% (three of seven) 24 × 24 multipliers from [12] and [13]. Furthermore, the Hornor's algorithm technique not only directly conserves hardware cost by lessening the requirements of multipliers but also provides a flexible architecture for the time multiplexed design. To compare the four-cycle and one-cycle architectures

	[12]	[13]	This Paper	[13]	This Paper
FPGA device/ ASIC Library	Altera EP20K600EBC652-1X	Altera EP20K600EBC652-1X	Altera EP20K600EBC652-1X	0.18 µm CMOS	0.18 µm CMOS
Total logic elements	18 344	7163	1686	N/A	N/A
Flip-flops	15 355	2811	770	N/A	N/A
Gate counts	N/A	N/A	N/A	44 992	13 917
Memory requirements (size of input image)	4 (1024 × 1024 byte)	4 (1024 × 1024 byte)	1 (1024 × 1024 byte)	4 (1024 × 1024 byte)	1 (1024 × 1024 byte)
Frequency	40 MHz	56.98 MHz	59.93 MHz	200 MHz	200 MHz
Throughput	30 Mpixels/s	40 Mpixels/s	14 Mpixels/s	140 Mpixels/s	40 Mpixels/s
Normalized logic elements/gate counts	10.88	4.25	1	3.23	1

Table. 6: Comparisons of This paper With Other Previous Low-Complexity Designs

of this paper in Table IV, there are three 24 × 24 multipliers are saved. Accordingly, the time multiplexed technique contributes reducing 75% (three of four) 24 × 24 multipliers from the architecture without time multiplexed design. The results show that the proposed three architectures consume the less hardware resources than [12] and [13] especially multi- pliers. The proposed four-cycle architecture achieves reducing hardware resources of 46.6% or 57.9% multipliers than [12] or [13].

Table V lists the comparison of the distortion correcting circuit with one-cycle, two-cycle, or four-cycle back-mapping circuit design. Since time multiplexed design, the two-cycle and four-cycle architecture decrease the throughput from 160 to 80 and 40 mega pixels per second (Mpixels/s). Although the throughputs of two-cycle and four-cycle architectures are much less than one-cycle design, other parameters such as hardware cost, memory requirements, and power consumptions have shown superiority as compared to one-cycle architecture. To summarize, the selection of back-mapping architecture should depend on the demand of image quality. For example, if the quality of the distortion image is the resolution of high definition multimedia interface of QSXGA (2560 × 2048) at 30 frames/s, only one-cycle architecture can be selected to meet the demand of throughput. In contrast, if the quality of image demands no more than the size of 1024 × 1024 at 30 frames/s, the four-cycle architecture is applicable due to low-cost and power consumption. As suggested above, we select the four-cycle time multiplexed architecture as our distortion correcting circuit for biomedical endoscopic applications.

III. EXPERIMENTAL RESULTS

The VLSI architecture of this paper was implemented by Verilog HDL. Based on a 0.18 μm CMOS standard cells, this design was synthesized by using SYNOPSYS Design Vision. The layout of this design was generated by the auto placement and routing tool SYNOPSYS Astro. Synthesis results show that the distortion-correction circuit contains 13 917 gates, and its chip area is 139 175 μm^2. The power consumption is 16.81 mW at 200 MHz operation frequency with 1.8 V supply voltage. Furthermore, we also implemented this design for emulation by a FPGA verification board. The Quartus II was used to synthesize this design based on Altera EP20K600EB FPGA and it consumes 1686 logic elements.

Table 6 lists the implemented results of this paper with previous low-complexity designs [12], [13]. The logic elements in FPGA and gates in silicon chip of this paper are both less than the others. It achieves the reductions of 90.8% or 76.4% logic elements in FPGA than [12] or [13], and 69% gates than [13]. The critical path of this design is 5 ns, which results in the operation clock frequency up to 200 MHz Although the time multiplexed design limits the throughput due to obtaining one result per four cycles, the throughput of this paper achieves 40

Mpixels/s. It is more than the value of (1024 × 1024 + 31) × 30, which includes 31 stages of pipeline delays to obtain the first result for each frame. Accordingly, this paper is fast enough to real-time correcting high quality biomedical endoscopic image (1024 × 1024) at 30 frames/s. Therefore, this de- sign satisfies the requirement of biomedical video-endoscopy applications.

Fig. 7 shows the distorted and corrected images resulting in wide-angle camera and endoscopic images. The distorted black-point image taken by a wide-angle camera is shown in Fig. 7(a) and the corrected image result by this paper is shown in Fig. 7(b). A typical gastrointestinal image captured by an endoscope [20] is shown in Fig. 7(c) or (e), and the corresponding corrected image result is shown in Fig. 7(d) or (f). Fig. 7(g) shows a gastrointestinal image captured by an endoscope with grid. The correcting effect can be obtained by comparing the corrected image result, as shown in Fig. 7(h), by this paper with the corresponding gastrointestinal image with grid.

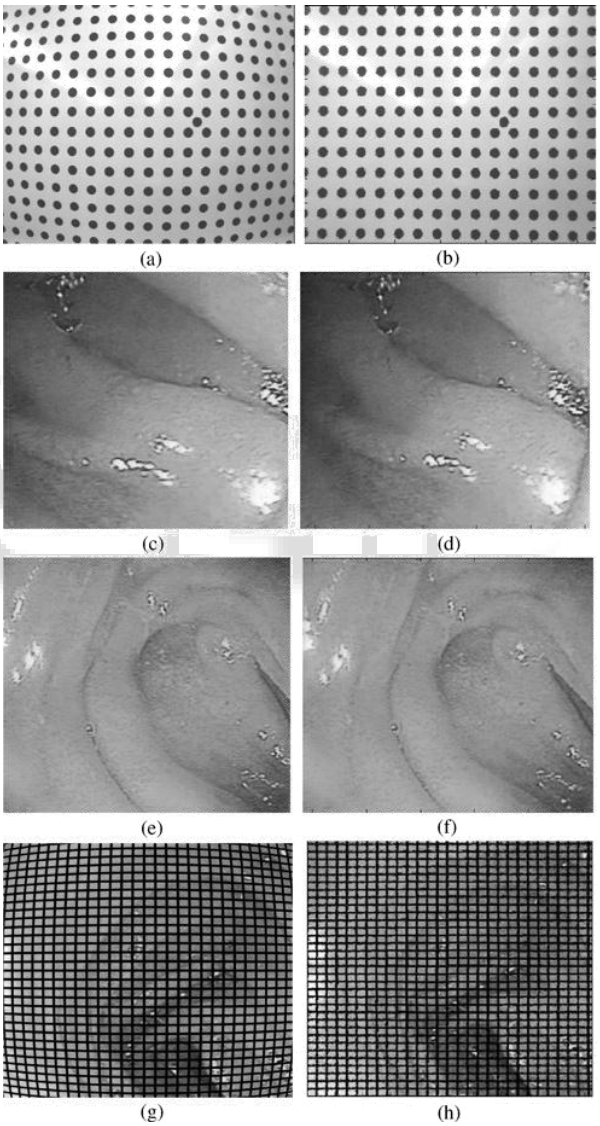

Fig. 7. Hardware emulation image results of correcting barrel distortion in wide-angle camera and endoscopic images. (a) Distorted black-point images. (b) Corrected black-point image by this paper. (c) Gastrointestinal image captured by video-endoscope. (d) Corrected gastrointestinal image by this paper. (e) Gastrointestinal image captured by video-endoscope. (f) Corrected gastrointestinal image by this paper. (g) Gastrointestinal image with grid. (h) Corrected grid gastrointestinal image by this paper.

IV. CONCLUSION

In this paper, a low-cost, low-power, and low memory requirement VLSI architecture of distortion correcting circuit was presented for biomedical endoscope applications. The computing complexity of correcting functions is decreased by Hornor's algorithm and the algebraic manipulation of the linear interpolation. Moreover, the time multiplexed de- sign provides different architectures for satisfying different applications. With reference to other low-complexity architectures, this paper reduces at least 69% hardware cost and 75% memory requirement than other VLSI correcting designs.

ACKNOWLEDGMENT

The authors would like to thank the Center for Micro/Nano Technology Research and the Center of Bio-science and Technology, National Cheng Kung University, Tainan, Taiwan.

REFERENCES

[1] W. E. Smith, N. Vakil, and S. A. Maislin, "Correction of distortion in endoscopic images," *IEEE Trans. Med. Imaging*, vol. 11, no. 1, pp.117–122, Jan. 1992.

[2] H. Hideaki, Y. Yagihashi, and Y. Miyake, "A new method for distortion correction of electronic endoscope images," *IEEE Trans. Med. Imaging*, vol. 14, no. 3, pp. 548–555, Mar. 1995.

[3] K. V. Asari, S. Kumar, and D. Radhakrishnan, "A new approach for nonlinear distortion correction in endoscopic images based on least squares estimation," *IEEE Trans. Med. Imaging*, vol. 18, no. 4, pp. 345–354, Apr. 1999.

[4] J. P. Helferty, C. Zhang, G. McLennan, and W. E. Higgins, "Video endoscopic distortion correction and its application to virtual guidance of endoscopy," *IEEE Trans. Med. Imaging*, vol. 20, no. 7, pp. 605–617, Jul. 2001.

[5] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," IEEE *Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 10, pp. 965–980, Oct. 1992.

[6] Y. Nomura, M. Sagara, H. Naruse, and A. Ide, "Simple calibration algorithm for high-distortion-lens camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 11, pp. 1095–1099, Nov. 1992.

[7] R. Swaminathan and S. K. Nayar, "Non-metric calibration of wide-angle lenses and polycameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1172–1178, Oct. 2000.

[8] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 8, pp. 1335–1340, Aug. 2006.

[9] R. Hartley and S. B. Kang, "Parameter-free radial distortion correction with center of distortion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 8, pp. 1309–1321, Aug. 2007.

[10] J. Park, S. C. Byun, and B. U. Lee, "Lens distortion correction using ideal image coordinates," *IEEE Trans. Consumer Electron.*, vol. 55, no.3, pp. 987–991, Aug. 2009.

[11] K. Asari, "Design of an efficient VLSI architecture for non-linear spatial warping of wide-angle camera image," *J. Syst. Architecture.*, vol. 50, pp. 743–755, Aug. 2004.

[12] H. T. Ngo and V. K. Asari, "A pipelined architecture for real-time correction of barrel distortion in wide-angle camera images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 436–444, Mar.2005.

[13] P. Y. Chen, C. C. Huang, Y. H. Shiau, and Y. T. Chen, "A VLSI implementation of barrel distortion correction for wide-angle camera images," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 56, no. 1, pp.51–55, Jan. 2009.

[14] M. T. El-Melegy and A. A. Farag, "Nonmetric lens distortion calibration: Closed-form solutions, robust estimation and model selection," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1. Oct. 2003, pp. 554–559.

[15] M. Marder, "Comparison of calibration algorithms for a low-resolution, wide angle, 3-D camera," M.S. thesis, Electr. Eng. Dept., KTH, Roy. Inst. Technol., Stockholm Univ., Stockholm, Sweden, Mar. 2005.

[16] W. G. Horner, "A new method of solving numerical equations of all orders, by continuous approximation," *Philos. Trans. Royal Soc. London*, pp. 308–335, Jul. 1819.

[17] P. Tummeltshammer, C. Hoe, and M. Puschel, "Time-multiplexed multiple-constant multiplication," *IEEE Trans. Comput.-Aided Des.s Integrated. Circuits Syst.*, vol. 26, no. 9, pp. 1551–1563, Sep. 2007.

[18] Y. S. Kwon and C. M. Kyung, "Performance-driven event-based synchronization for multi-FPGA simulation accelerator with event time- multiplexing bus," *IEEE Trans. Comput.-Aided Des. Integrated Circuits Syst.*, vol. 24, no. 9, pp. 1444–1456, Sep. 2005.

[19] J. D. Bruguera, N. Guil, T. Lang, J. Villalba, and E. L. Zapata, "CORDIC based parallel/pipelined architecture for the Hough transform," *J. VLSI Signal Process.*, vol. 12, no. 3, pp. 207–221, Jun. 1996.

[20] Gastroenterologist Hospital. *El Salvador Atlas of Gastrointestinal Video Endoscopy* [Online]. Available: http://www.gastrointestinalatlas.com medicine.

