Correspondence

Real-Time System for High-Image Resolution Disparity Estimation

Javier Díaz, Eduardo Ros, Richard Carrillo, and Alberto Prieto, *Senior Member, IEEE*

***Abstract*—We present the hardware implementation of a simple, fast technique for depth estimation based on phase measurement. This technique avoids the problem of phase warping and is much less susceptible to camera noise and distortion than standard block-matching stereo systems. The architecture exploits the parallel computing resources of FPGA devices to achieve a computation speed of 65 megapixels per second. For this purpose, we have designed a fine-grain pipeline structure that can be arranged with a customized frame-grabber module to process 52 frames per second at a resolution of 1280 × 960 pixels. We have measured the system's degradation due to bit quantization errors and compared its performance with other previous approaches. We have also used different Gabor-scale circuits, which can be selected by the user according to the application addressed and typical image structure in the target scenario.**

***Index Terms*—Embedded and real-time systems, pipeline processing, scale space, stereo image processing.**

I. Introduction

The biological capacity for binocular depth perception is useful in many visual domains such as autonomous navigation, 3-D reconstruction, active tracking, or face recognition [1]–[4]. It permits the reconstruction of information about depth encoded within binocular images, a task which is performed in the visual cortex by specialized receptive field structures [5]. Studies have shown that a substantial proportion of neurons in the striate and prestriate cortex of monkeys have stereoscopic properties, i.e., they respond differentially to binocular stimuli, thus providing cues for stereoscopic depth perception [6]–[9].

Engineered processing architectures specifically designed for tasks that biological systems solve with impressive efficiency can benefit considerably by mimicking computing strategies developed by nature over the long process of evolution, but the adaptation of such techniques is not straightforward since the physical principles upon which biological tissues are based differ considerably from those used in electronic technology. Furthermore, biological and electrical "technologies" face different constraints, which can be overcome by resorting to different strategies. Nevertheless, an "opportunistic attitude" which takes the key functional principles that contribute to the outstanding performance of biological systems and also uses technology-motivated processing techniques to adapt those computing primitives must be of considerable interest.

We illustrate here one example of such an approach using FPGAs, which have already shown their high performance capacity for image-processing tasks [10]. The proper use of fine-grain pipeline structures combined with that of FPGA parallelism and the high processing speed of the semiconductor substrate allow us to reproduce bio-inspired processing capabilities. These hardware-based approaches normally rely on correlation-based models [11] because they fit in well with specific hardware architectures. Nevertheless, over the last decade, phase-based computational models have been proposed as an interesting alternative [12] to feature correspondence and correlation techniques, mainly because they are based on local operations and produce dense depth maps with direct subpixel resolution. Several real-time approaches based on this technique have been proposed recently [13] and [14].

Our contribution goes one step beyond these approaches. We describe here a stereo-processing system based on an FPGA device that computes a bio-inspired modified phase-based technique described by Solari *et al.* [15]. This model avoids the explicit computation of the phase difference of Gabor filters, thus making the approach hardware friendly and allowing our design to outperform previous approaches considerably. The main innovation of our contribution is the design of a finely pipelined data-path able to compute one estimation per system clock cycle. The translation of a processing model into an efficient specific-purpose circuit is not easy and requires a structured implementation of the different stages, clearly defining their data exchanges and the potential computing parallelism at each stage. This is described in the next section.

In addition, our approach allows the stereo computation of high-resolution images faster than video rates. This is of crucial importance since the reliability of stereo-depth estimation depends highly upon the resolution of the input images.

Furthermore, the proposed scheme is scalable since there are plenty of computing resources available on the same chip; if further parallelism is needed, two or more processing units can be used to extract more estimations or increase the spatial resolution.

II. Hardware-Friendly Phase-Based Stereo

The adopted computing model has been proposed by Solari and Sabatini [15]. In a first approach, the positions of corresponding points are related by a 1-D horizontal shift, the disparity, along the direction of the epipolar lines. Formally, the left and right observed intensities of the two eyes, $I^L(x)$ and $I^R(x)$, can be expressed in terms of (1)

$$I^R(x) = I^L[x + \sigma(x)] \tag{1}$$

where $\pm\sigma(x)$ is the (horizontal) binocular disparity.

Disparity can be estimated in terms of phase differences in the spectral components of the stereo-image pair [8]. Since the two images are locally related by a shift, within the neighborhood of each image point the local spectral components of $I^L(x)$ and $I^R(x)$ are related by a phase difference equal to $\Delta\phi(k) = \phi^L(k) - \phi^R(k) = k\delta$. Spatially localized phase measurements can be obtained by filtering operations with complex-valued, quadrature-pair, bandpass kernels (e.g., Gabor filters), approximating a local Fourier analysis of the retinal images. Taking a complex Gabor filter (h) with a peak frequency k_0 and corresponding Gaussian variance σ^2

$$h(x;k_0) = \exp\left(\frac{-x^2}{\sigma^2} + jk_0x\right) = h_C(x;k_0) + jh_S(x;k_0) \tag{2}$$

the resulting convolutions with the left and right binocular signals can be expressed as in (3)

$$Q(x) = \int I(\xi)h(x-\xi;k_0)d\xi = C(x) + jS(x) = \rho(x)e^{j\phi(x)} \tag{3}$$

Manuscript received December 5, 2005; revised June 5, 2006. This work was supported in part by the Spanish Government under Grant DPI2004-07032 and in part by the EU under Grant DRIVSCO: 016276-2. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Srdjan Stankovic.

The authors are with the Computer Architecture and Technology Department, University of Granada, E-18071 Granada, Spain (e-mail: jdiaz@atc.ugr.es; eros@atc.ugr.es; rcarrillo@atc.ugr.es; aprieto@ugr.es).

Digital Object Identifier 10.1109/TIP.2006.884931

where $\rho(x)$ and $\phi(x)$ denote their amplitude and phase components, while $C(x)$ and $S(x)$ are the responses of the quadrature filter pair (C and S stand for cosine and sine, respectively). Local phase measurements are stable, with a quasi-linear behaviour over relatively large spatial extents, except around singular points, where the amplitude of $Q(x)$ vanishes and the phase becomes unreliable. This property of the phase signal yields good predictions of binocular disparity by

$$\sigma(x) = \frac{\phi^L(x) - \phi^R(x)}{k(x)} = \frac{\lfloor \phi(x) \rfloor_{2\phi}}{k(x)} \tag{4}$$

where we note $\lfloor\ \rfloor_{2\pi}$ as the principal part of the argument (i.e., ϕ belongs to $[-\pi, \pi]$) and $k(x)$ is the average instantaneous frequency of the bandpass signal, measured using the phase derivative from the left and right filter outputs (x subscripts indicate differentiation along the x axis)

$$\frac{k(x) = \phi_x^L(x) + \phi_x^R(x)}{2.} \tag{5}$$

As a consequence of the linear-phase model, the instantaneous frequency is generally constant and close to the tuning frequency of the filter $(k(x) \approx k_0)$, except near singularities, where abrupt frequency changes occur as a function of spatial position. Therefore, a disparity estimation at a point x is accepted only if $|(\phi_x - k_0)| k_o \tau$, where τ is a proper reliability threshold.

It should be noted that (4) does not require the explicit calculation of the left and right phases, and, thus, we can compute the phase difference in the complex plane directly using the following identities:

$$\begin{aligned} \lfloor \phi(x) \rfloor_{2x} &= \lfloor \arg(Q^L Q^{*R}) \rfloor_{2\phi} \\ &= \arctan 2(\mathrm{Im}(\arg(Q^L Q^{*R}), \mathrm{Re}(\arg(Q^L Q^{*R}))) \\ &= \arctan 2(C^R S^L - C^L S^R, C^L C^R + S^L S^R). \end{aligned} \tag{6}$$

This approach has several advantages that make the system hardware-friendly. Although (6) increases the number of multiplications compared to circuits with direct phase subtraction, current FPGA devices include embedded multipliers for DSP operations, which make this technology particularly interesting for vision tasks. In fact, the main advantage provided by this approach is that of avoiding the explicit logic required for the wrap-around mechanism. This implies reducing comparison logic considerably. Furthermore, the number of division operations is reduced by 50%. This reduction is important because the division using fixed-point arithmetic requires high precision. In fact, quantization errors make the former approach noisier and, thus, demand more hardware resources to achieve similar accuracy.

To address the hardware implementation of this approach the basic steps can be summarized as follows:

1) DC component image removal using the local image contrast I-I_{mean} operator for the even Gabor filter;
2) even (C) and odd (S) 1-D Gabor filtering of left and right images;
3) direct phase-difference calculation from (6);
4) disparity computation using (4), assuming $k(x) \approx k_0$.

III. Hardware Implementation

Most of the previous real-time contributions are based on correlation techniques [11] because this approach fits in quite well with customized hardware but the choice of a phase-based stereo approach is also justified because of its capacity to reduce illumination problems. As mentioned in [16], a contrast test shows that this approach is not very susceptible to differences in local contrast. It also seems to be capable of dealing with imbalanced images too, which are usual in real cameras since they have slightly different luminance gains.

The phase-based approach can also be of interest in biological studies to extract real-time working models. The problem is that this high-level system description is hard to implement onto hardware structures. This has prompted us to use the Handel-C HDL [17], which allows the system's data-path to be described in a very algorithmic-like manner.

A. Camera Calibration

Setting up the system requires image rectification and camera calibration (which is a critical stage). After a manual calibration to arrange the cameras in parallel, the current implementation only involves a simple preprocessing method based on image displacement, which runs in a setup system configuration stage as follows:

We define a plane of null disparity and we allocate a flat object or picture on it with some texture (for instance a chessboard pattern). A frame-grabber shift of up to 32 pixels is explored iteratively along the horizontal and vertical co-ordinates to obtain the best overall matching value within that range (integrating four times during each iteration to reduce the error due to camera noise and image flickering). This plane defines the zero disparity distance. Closer and farther objects will lead to positive and negative disparities respectively. This reference plane is defined depending on the camera's configuration, system scale and target application to properly tune the filter disparity range to the target scenario (close to the reference plane). With this method, we reduce the range of disparities presented at the image close to this reference plane, which allows us to recover disparities with only a small Gabor kernel. When the calibration process finishes, the system is auto-reprogrammed from external Flash memory with the new configuration file and the stereo computation starts.

This simple calibration process takes about 32 s using a 40-MHz FPGA clock, but this time is not critical since the calibration is computed only once at the initial configuration stage. In fact, up to a 70-MHz clock frequency is supported by this circuit, but we only use a 40-MHz clock to facilitate the on-line generation a VGA output of the imaged matching process. In this way, we are able to visually monitor the calibration process (iterative matching). This allows us to discard wrong initial camera settings that lead to poor matching results. Future work will address an improved calibration preprocessing including image rectification techniques.

B. System Architecture

Handel-C [17] allows us to define very straightforwardly the level of parallelism and pipelined structure, which can be easily grouped on the basis of functionality and finely subdivided to get well balanced pipelined structures of high data throughput.

According to this strategy, the system is configured in seven functional stages (coarse-grain pipelined structure) which are divided into fine-grain pipelined substage data paths. This leads to a total latency of 115 clock cycles (equal to the number of fine pipeline stages) and a data throughput of one estimation per clock cycle.

The stereo architecture according to this strategy is shown in Fig. 1. There are two parallel pathways which process each camera image to compute the Gabor-filtered values (implemented as optimized convolution circuits). The level of parallelism at each stage has been expanded to achieve a data throughput of one estimation per clock cycle. The direct calculation of phase difference expressed in (6) is based on two different paths, (A) and (B), in the circuit. Unit (B) computes the disparity value whilst unit (A) measures the confidence estimation (module of the Gabor filter outputs). We use this confidence measurement since phase is not clearly defined near module singularities, and, therefore, no reliable information is present at these points [16]. The *TH Buffer* is a memory buffer used to balance the two processing paths (A and B).

The system has been fully implemented in a stand-alone board as a prototype for embedded applications (the RC300 board [18]). This complete system setup is shown in Fig. 2. All the processing operations

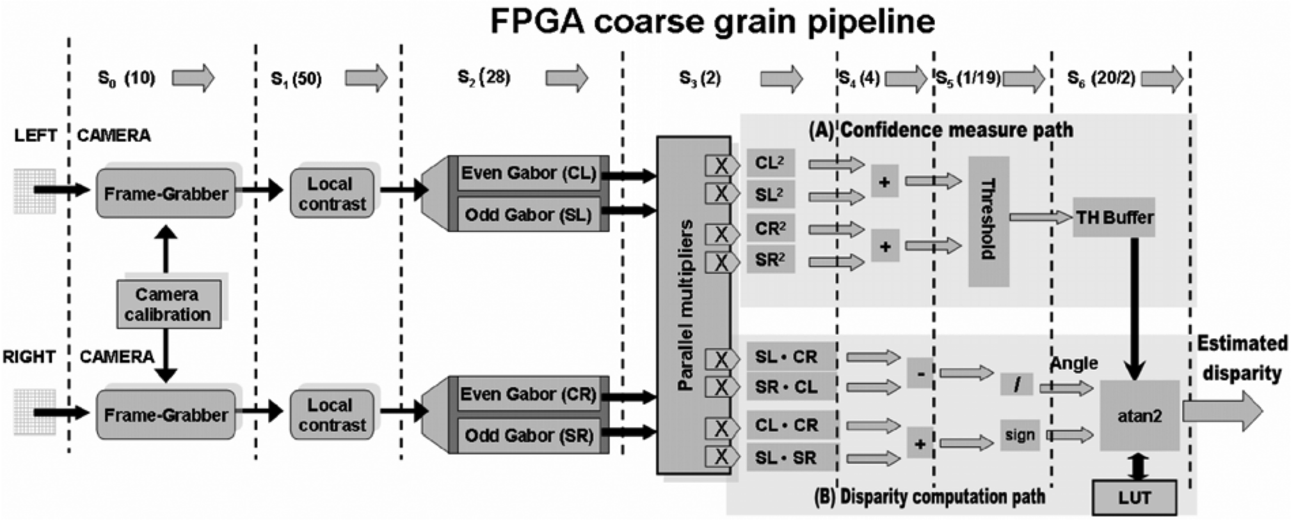

Fig. 1. Stereo-system hardware architecture. System calibration parameters are stored and used as input shifts (horizontal and vertical) for the camera frame-grabbers of stage S_0. At stage S_1, the local contrast is removed to eliminate even Gabor filter DC response. At S_2, we compute the even and odd Gabor outputs (where “S” stands for the sine or odd Gabor filter and “C” for the cosine or even Gabor filter). Note that left and right images have parallel pathways during these stages (high processing performance is enhanced by replicating scalar units). “L” and “R” denote the responses coming from the left and right images. Stages S_3 to S_6 implement the direct phase-difference computation as described in (6). Left and right image responses are combined during these stages into two different datapaths. The upper pathway (A) computes the Gabor output energy, which is used as a confidence measure. The bottom pathway (B) computes the phase difference. Note that the efficient use of the intrinsic parallelism available in the FPGAs is achieved by a customized pipeline processing architecture based on well balanced parallel computing blocks at different stages. It allows the computation of one estimation per clock cycle. For this purpose, we have designed a micropipelined architecture. In the upper part of the figure, we indicate in brackets the number of micropipelined steps at each functional stage; in stages S_5 and S_6, “/” is used to indicate the different number of micropipelined steps of each of their parallel datapaths (A/B).

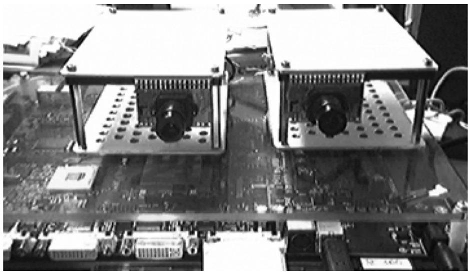

Fig. 2. Stereo processing system setup.

are computed in the FPGA device as a *system-on-a-chip* (SoC), which also contains the camera’s frame-grabbers, memory management units, VGA controller and user interface.

IV. Analysis of System Requirements and Performance

The full system has been successfully implemented on a Xilinx Virtex-II FPGA [19]. The system frequency is 65 MHz and due to the regular data path of the proposed model, we achieve one pixel per clock cycle. This means that we can compute up to 65 megapixels per second (arranged as 52 fps of 1280 × 960 pixels per image, for instance). The consumption of system resources has also been evaluated. The factors to take into account for implementation are those of model degradation due to limited fixed-point bit-width and to the Gabor-filter wavelength (large values improve the disparity range but consume more resources.).

We have arrived at several conclusions concerning the data representation and bit width at each pipeline stage. The bit width of the convolved images with the Gabor filters is critical because its precision affects the following stages in two ways. First, the bit width of the next computation grows concomitantly with the square of the number of bits of this stage, and, second, any limitations in precision are transferred to the following stages, thus reducing the overall accuracy of the system.

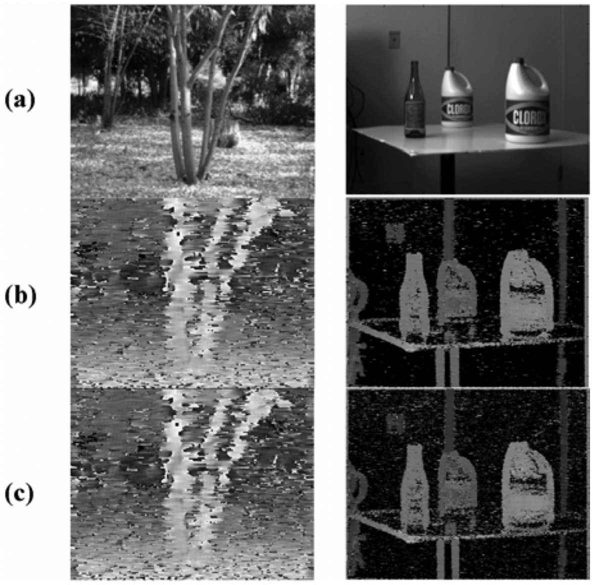

Fig. 3. Software versus hardware implementation: (a) original images, (b) software stereo processing, and (c) hardware stereo processing. The image on the left was processed using a small scale (Gabor filters with a length of 15 pixels) and that on the right with a medium scale (Gabor filters with a length of 31 pixels). Note that only small differences are visible as an increase in salt-and-pepper noise (more visible in the right-hand image) in the hardware results due to the restricted precision available in the hardware implementation.

On the basis of a previous study concerning the bit width of the system [20], our stages are developed as follows.

- At the convolution stages, the processing is done with fixed-point data representation of 9 bits.
- Intermediate data precision is 19 bits using fixed-point arithmetic, avoiding bit wrapping or saturation operations.

TABLE I
SYSTEM RESOURCES REQUIRED ON A VIRTEX II XC2V6000-4. FIRST ROW: SIMPLE CAMERA CALIBRATION SYSTEM. FOLLOWING ROWS: PHASE-BASED STEREO DEVICE USING SEVERAL GABOR SCALES ("Mpps": MEGA-PIXELS PER SECOND AT ITS MAXIMUM SYSTEM PROCESSING CLOCK FREQUENCY). "EMBs" STANDS FOR EMBEDDED MEMORY BLOCKS

Slices / (%)	**EMBs /(%)**	**Embedded multipliers / (%)**	**Mpps**	**Gabor spatial scale (filter length)**	**Image Resolution**	**Fps**
2864 / (8%)	1 / (1%)	0 / 0	70	-	640x480	56
6411 / (18%)	15 / (10%)	21 / (14 %)	65	15	640x480 1280x960	211 52
9197 / (27 %)	39 / (27 %)	31 / (21 %)		31		
13048 / (38%)	71 / (49%)	59 / (49 %)		55		

- The division operation is implemented using a Xilinx pipelined division core [19] with 24 bits (19 bits from the above data plus a fractional part of 5 bits for the $\arctan$ function).
- The $\arctan$ function is implemented using a look-up table of 1024 addresses of 10 bits with 5 fractional bits. Only the $[0, \pi/2]$ interval is sampled. A decision logic based on the input data sign allows us to recover the quadrant of the angle within the full range $[-\pi, \pi]$. This simple scheme allows a maximum estimation error of 0.03 rad for the $\arctan$ function with a very simple logic and, therefore, complex circuits, such as CORDIC [21], for example, are not required.

Fig. 3 shows the disparity estimation for a couple of real binocular image pairs. Gray levels encode depth information (lighter levels indicate closer objects). Software (with double floating-point representation data) and hardware (with limited fixed-point data representation) approaches are compared. Qualitatively, the degradation is very low. For a quantitative study, we measure the quantization error of the images in Fig. 3, which have significantly reduced bit width. We measure the mean disparity error for the left and right images, obtaining values within the interval of 0.05 to 0.06 pixels. This represents a negligible noise contribution compared to the error coming from the stereo estimation model itself, which hardly achieves a precision higher than 0.1 pixels for arbitrary scenes [11]. We also measure the *signal-to-quantization-to-noise ratio* (SQNR), obtaining values within the interval of 17.1 to 23.52 dB, which indicates signal energy at least 50 times larger than the quantization noise. We conclude from these measurements that the system quantization degradation *versus* hardware resources consumption (Table I) tradeoff is appropriate. For more details on the bit-cutting procedure, see [20].

Given that stereo techniques work better for small disparities, we have designed three different scales, using Gabor filters with a length of 15, 31, and 55 pixels. The Gabor-filter wavelength determines system quality according to image resolution and disparity range.

The disparity range for each circuit is ± 4, ± 8, and ± 14 pixels, respectively [16]. It is important to note that larger filters work as low-pass filters and high-frequency image structures are lost; therefore, although the first stage of camera calibration reduces overall image displacement, the Gabor filters must be tuned to the desired application to get the best results.

According to this strategy, the implementations presented here, which only compute one scale, must be scale tuned according to the application addressed and image structure required. This is done by the user (or agent), who can reconfigure the circuit (i.e., the FPGA device can be reprogrammed in less than 400 ms) from the system interface or PC command line to change the disparity scale. Future studies will try to combine different scales dynamically based on the image structure without the need for intervention by the user.

The final consumption of system resources for the whole system with the choice of bit widths described above is shown in Table I. We consider these bit widths to be good tradeoffs between system accuracy and hardware resource requirements. The first row indicates the calibration-system circuit resources, and the following rows show the consumption for each Gabor scale, which grows for longer scales.

Any evaluation of system performance should take into account image resolution, frames per second and the number of cameras. It is also important to consider the searching area where the two images are compared (small searching areas or filter lengths require less computing performance than larger areas). According to these criteria, we used a common comparison metric of stereo-vision systems [14] in order to rank the system, the performance being given by measuring the number of disparities computed per second. This is the point-time disparity per second (PDS), measured as: $\mathrm{PDS} = N \cdot D \cdot (C - 1)$, where N is the number of pixels processed per second and D the number of disparity values estimated (the disparity range). We also include C, which is the number of cameras to extend the metric to multi-baseline stereo approaches. It should be borne in mind that this metric only measures the system performance and not the architecture complexity, which is based on different factors such as the disparity estimation model and the computing platform.

Using this metric, our binocular system achieves different performances according to the Gabor scale and consequently shows different system-resource consumptions. Equation (7) calculates the performance of the system in terms of the PDS of the different configurations based on scales of 15, 31, and 55 Gabor-filter lengths. The disparity range for each circuit is ± 4, ± 8, and ± 14 pixels, which gives us equivalent D values of 9, 15, and 29, respectively. Thus, the corresponding PDS values are as follows:

$$\mathrm{PDS} = N \cdot D \cdot (C - 1) = N \cdot \mathrm{f}_{\mathrm{clk}} \cdot 1 = \begin{cases} 9 \cdot 65 = 585\ \mathrm{M} \\ 15 \cdot 65 = 975\ \mathrm{M} \\ 29 \cdot 65 = 1885\ \mathrm{M}. \end{cases} \quad (7)$$

A comparative performance study is shown in Table II. The large differences between architectures (standard processors, custom hardware, Graphical cards, and FPGAs) makes a direct comparison difficult but it is still worthwhile illustrating how well each approach fits the stereo computation task. We use the system raw performance in PDS as comparative metric.

There are recent software-based approaches, but most of the processing platforms are based on FPGAs [10] and they use different FPGA families with different performances, which hinder comparison due to technological advances. Because of that we are not taking, resources consumption into account. In FPGA-based approaches, we try to reduce any comparative bias due to technological advances by normalizing the PDS performance by the clock frequency as shown in Table II. The performance obtained by our system is faster than the earlier block-matching-based binocular implementations commented upon in [11] (the fastest version of which is included in the sixth row of Table II).

TABLE II
Performance Comparison of Selected Real-Time Binocular Systems. We Have Only Included Systems With Performance Information Available in the Literature. In the Systems Based on FPGAs, we Also Include the Normalized PDS/f_{clk} Performance Measurements in Order to Facilitate the Evaluation of the Efficient Use of the Parallel Resources Available Within These Devices

Real-time system	**Image resolution**	**fps**	**Disparity range**	**PDS x10^6 / [PDSx10^6/f_{clk}]**	**Method**	**Processor type**
Proposed here	*1280x960*	*52*	*9* *15* *29*	*585 / 9* *975 / 15* *1885 /29*	*Phase based*	*Custom FPGA, Xilinx Virtex-II (65 MHz)*
Gong and Yang [22] (2005)	512x384	14.7	40	117 / --	Correlation-based with image-gradient-guided cost aggregation	Pentium 4 3GHz equipped with an ATI 9800 XT (412 MHz)
Forstmann et al. [23] (2004)	256x256 640x480 1024x1024	30.4 7.23 2.2	100	200 / -- 222 / -- 230 / --	Dynamic programming	AMD AthlonXP 2800+ and MMX optimization
Niitsuma and Maruyama [24] (2004)	640x480	30	27	248.8 / 3,66	Correlation. SAD	Custom FPGA, Xilinx Virtex-II (68 MHz)
Darabiha et al. [14] (2003)	360x256	30	20	55.3 / 1,1	Correlation phase-based	Custom FPGA, Xilinx Virtex, (50MHz)
Woodfill, and Herzen [25] (1997)	320x240	42	24	77.4 / 2,35	Census matching	Custom FPGA Xilinx XC4000 (33MHz)
T. Kanade et. al. [26] (1994)	256x240	24,4	20	30 / --	Multi-baseline Correlation. SSAD	Custom HW & C40 DSP (2-6 cameras)

Table II includes approaches with software implementations using graphic cards and MMX extensions [22], [23]. Despite the computing performance of such systems being quite high, they consume all the resources of the computer, rendering it impossible to compute higher level algorithms based on stereo in real time. Furthermore, it is difficult to use these approaches on mobile platforms for embedded applications such as robotics or smart sensors.

In terms of PDS, our system outperforms the fastest implementation in Table II (more specifically, the approach of Niitsuma and Maruyama [24], by a factor of between 2.3 and 7.5, depending upon the chosen Gabor scale) and larger factors for other FPGA-based systems [14], [25]. Similar outperformances are achieved when using the normalized PDS. Note that this system [24] uses the same FPGA technology and fast clock frequency but our outstanding performance is based most significantly on the optimized computing architecture described in this work. This becomes clearer when we compare the normalized PDS to evaluate the computing parallelism of the different approaches. Our system achieves far higher normalized PDS through the intensive use of the parallel processing resources available within the FPGA device (mainly due to the fine pipeline architecture).

Finally, the last row in Table II shows a very interesting approach [26]. In this case, a comparison with our system is not wholly just because they have built a full custom system, which can not be easily updated, while our approach easily takes advantage of continuous technological advances). Furthermore, this system has very high power consumption while in ours everything is built on the same chip, which significantly reduces the power required.

There are also commercial products such as Bumblebee and Digiclops from Point Grey Research (http://www.ptgrey.com/). These devices consist of calibrated stereo cameras plus software libraries to compute the stereo. We have not included these devices in the comparison in Table II because the information about processing performance on standard computing platforms is obsolete.

V. Conclusion

We present a bio-inspired model implemented onto programmable hardware that runs on a stand-alone chip for embedded applications. The pipeline processing structure, including some well-balanced parallel processing modules, efficiently computes phase-based disparity estimations. The most important contribution of this work is the efficient implementation of a vision model on specific circuits adopting a well structured design strategy, as described in Section III. The regular data path is able to compute one pixel per system clock cycle. This efficient use of the parallel computing resources available on FPGAs plus a fine-grain pipeline design lead to an outstanding processing speed (65 megapixels per second, which can be arranged as 52 fps of 1280 × 960 pixels per image).

The system includes an automatic precalibration stage to improve the system disparity range, as well as the possibility of switching between Gabor scales according to the application addressed and image structure in the target scenario. We have measured the system degradation due to bit-width restrictions and decided upon a good tradeoff between degradation and resource consumption.

In the future, we plan to study the implementation of a multiple-scale stereo system that takes advantage of the designed architecture and combines the different scales according to the image structure presented in the neighborhood of each image pixel.

Acknowledgment

The authors would like to thank S. Sabatini and F. Solari for their support in the stereo model and A. Tate for her revision of our English text.

References

[1] M. Bertozzi and A. Broggi, “GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection,” *IEEE Trans. Image Process.*, vol. 7, no. 1, pp. 62–81, Jan. 1998.

[2] L. Oisel, E. Memin, L. Morin, and L. F. Galpin, "One-dimensional dense disparity estimation for three-dimensional reconstruction," *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1107–1119, Sep. 2003.

[3] L. Ziyi and B. E. Shi, "Subpixel resolution binocular visual tracking using analog VLSI vision sensors," *IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process.*, vol. 47, no. 12, pp. 1468–1475, Dec. 2000.

[4] F. Tsalakanidou, S. Malassiotis, and M. G. Strintzis, "Face localization and authentication using color and depth images," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 152–168, Feb. 2005.

[5] G. C. DeAngelis, I. Ohzawa, and R. D. Freeman, "Depth is encoded in the visual cortex by a specialized receptive field structure," *Nature*, vol. 11,352, no. 6331, pp. 156–159, 1991.

[6] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, pp. 106–154, 1962.

[7] H. B. Barlow, C. Blakemore, and J. D. Pettigrew, "The neural mechanism of binocular depth discrimination," *J. Physiol.*, vol. 193, pp. 327–342, 1967.

[8] G. C. DeAngelis, B. G. Cumming, and W. T. Newsome, "Cortical area MT and the perception of stereoscopic depth," *Nature*, vol. 394, pp. 677–680, 1998.

[9] G. F. Poggio and T. Poggio, "The analysis of stereopsis," *Annu. Rev. Neurosci.*, vol. 7, pp. 379–412, 1984.

[10] B. A. Draper, J. R. Beveridge, A. P. W. Bohm, C. Ross, and M. Chawathe, "Accelerated image processing on FPGAS," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1543–1551, Dec. 2003.

[11] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 993–1008, Aug. 2003.

[12] D. J. Fleet and A. D. Jepson, "Stability of phase information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 12, pp. 1253–1268, Dec. 1993.

[13] B. Porr, B. Nürenberg, and F. A. Wörgöter, "VLSI-Compatible computer vision algorithm for stereoscopic depth analysis in real-time," *Int. J. Comput. Vis.*, vol. 49, no. 1, pp. 39–55, 2002.

[14] A. Darabiha, J. Rose, and W. J. MacLean, "Video-Rate stereo depth measurement on programmable hardware," presented at the Int. Conf. Computer Vision and Pattern Recognition, Madison, WI, Jun. 2003.

[15] F. Solari, S. P. Sabatini, and G. M. Bisio, "Fast technique for phase-based disparity estimation with no explicit calculation of phase," *Elect. Lett.*, vol. 37, no. 23, pp. 1382–1383, 2001.

[16] A. Cozzi, B. Crespi, F. Valentinotti, and F. Wörgötter, "Performance of phase-based algorithms for disparity estimation," *Mach. Vis. Appl.*, vol. 9, pp. 334–340, 1997.

[17] Handel-C Language Reference Manual, Celoxica Company, 2003.

[18] Celoxica Company. [Online]. Available: http://www.celoxica.com

[19] Xilinx Company. [Online]. Available: http://www.xilinx.com

[20] J. Díaz, E. Ros, S. Mota, E. M. Ortigosa, and B. del Pino, "High performance stereo computation architecture," in *Proc. Int. Conf. Field Prog. Logic and Applications*, Tampere, Finland, Aug. 24–26, 2005, pp. 463–468.

[21] J. Valls, M. Kuhlmann, and K. K. Parhi, "Evaluation of CORDIC algorithms for FPGA design," *J. VLSI Signal Process.*, vol. 3, pp. 207–222, Nov. 2002.

[22] M. Gong and R. Yang, "Image-gradient-guided real-time stereo on graphics hardware," in *Proc. 5th Int. Conf. 3-D Digital Imaging and Modeling*, Ottawa, ON, Canada, Jun. 13–16, 2005, pp. 548–555.

[23] S. Forstmann, S. Thüring, Y. Kanou, J. Ohya, and A. Schmitt, "Real time stereo by using dynamic programming," in *Proc. Conf. Computer Vision and Pattern Recognition*, Washington, DC, Jul. 02, 2004, vol. 3, pp. 29–29.

[24] H. Niitsuma and T. Maruyama, "Real-time detection of moving objects," in *Lecture Notes in Computer Science*. New York: Springer-Verlag, 2004, vol. 3203.

[25] J. Woodfill and B. V. Herzen, "Real-time stereo vision on the parts reconfigurable computer," in *Proc. Conf. IEEE Symp. Field-Programmable Custom Computing Machines*, Napa, CA, Apr. 1997, pp. 201–201.

[26] T. Kanade, "Development of a video-rate stereo machine," in *Proc. ARPA Image Understanding Workshop*, Nov. 1994, pp. 549–558.

Inpainting of Binary Images Using the Cahn–Hilliard Equation

Andrea L. Bertozzi, Selim Esedoḡlu, and Alan Gillette

***Abstract*—Image inpainting is the filling in of missing or damaged regions of images using information from surrounding areas. We outline here the use of a model for binary inpainting based on the Cahn–Hilliard equation, which allows for fast, efficient inpainting of degraded text, as well as super-resolution of high contrast images.**

***Index Terms*—Binary images, Cahn–Hilliard equation, image inpainting, super-resolution.**

I. Introduction

Image inpainting is the filling in of damaged or missing regions of an image with the use of information from surrounding areas. In its essence, it is a type of interpolation. Its applications include restoration of old paintings by museum artists, and removing scratches from photographs.

The pioneering work of Bertalmio *et al.* [1] introduced image inpainting for digital image processing. Their model is based on nonlinear partial differential equations, and imitates the techniques of museum artists who specialize in restoration. They focused on the principle that good inpainting algorithms should propagate sharp edges into the damaged parts that need to be filled in. This can be done, for instance, by connecting contours of constant grayscale image intensity (called isophotes) to each other across the inpainting region (see also Masnou and Morel [2]), so that gray levels at the edge of the the damaged region extend continuously into the interior. They also impose the direction of the isophotes as a boundary condition at the edge of the inpainting domain. In subsequent work with Bertozzi [3], they realized that the method in [1] has intimate connections with 2-D fluid dynamics through the Navier–Stokes equation. Indeed, the steady-state equation proposed in [1] is equivalent to the inviscid Euler equations from incompressible flow, in which the image intensity function plays the role of the stream function in the fluid problem. The natural boundary conditions for inpainting are to match the image intensity on the boundary of the inpainting region, and also the direction of the isophote lines $(\nabla^{\perp} I)$. For the fluid problem, this is effectively a generalized "no-slip" boundary condition that requires a NavierStokes formulation, introducing a diffusion term. This analogy also shows why diffusion is required in the original inpainting problem. In practice, nonlinear diffusion [4], [5] works very well to avoid blurring of edges in the inpainting.

A different approach to inpainting was proposed by Chan and Shen [6]. They introduced the idea that well-known variational image denoising and segmentation models can be adapted to the inpainting task by a simple modification. These models often include a fidelity term that keeps the solutions close to the given image. By restricting the effects of the fidelity term to the complement of the inpainting region,

Manuscript received December 9, 2005 ; revised June 30, 2006. This work was supported in part by the National Science Foundation (NSF) and the intelligence community through the joint "Approaches to Combat Terrorism" program under Grant AST-0442037, in part by the Office of Naval Research under Grant N000140410078, and in part by NSF Grants ACI-0321917 and DMS-0410085. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Thierry Blu.

A. L. Bertozzi and A. Gillette are with the Mathematics Department, University of California, Los Angeles, CA 90095-1555 USA (e-mail: bertozzi@math.ucla.edu; agillett@math.ucla.edu).

S. Esedoḡlu is with the Department of Mathematics, University of Michigan, Ann Arbor, MI 48109-1043 USA (e-mail: esedoglu@umich.edu).

Digital Object Identifier 10.1109/TIP.2006.887728