A Parallel DSP System for Real–time Disparity and Optical Flow Using Phase Difference

F. Valentinotti, G. Di Caro, B. Crespi*
IRST, I-38050 Povo, Trento, ITALY
e-mail: `crespi@irst.it`

1 Introduction

The evaluation of disparity maps and optical flows are computationally intensive operations because they involve the search of corresponding points in pairs of images taken from different points of view. Since the basic computation is the same, the same computational technique can be used to treat both problems. In this way, software and hardware requirements for the implementation of the procedures can be better optimized and partial results can be shared by the different processing modules.

In this work an algorithm based on the phase-difference technique introduced in [1, 2] is developed and implemented on a TI-C40 [3] based parallel DSP system. The module performs real-time disparity estimation on images of 128×128 pixels and real-time computation of the optical flow on images of dimension 64×64.

2 Phase-difference-based Stereopsis

In the simplest stereo configuration, in which the optical axes of the cameras are parallel, the positions of corresponding points on the left and on the right images are related by a one-dimensional shift $\delta(x)$ along the x–axis, i.e. *position disparity*, depending on the distance of the point from the cameras.

In the phase-difference method, disparity is computed from the phase difference between the convolutions of the two stereo images, I_1 and I_2, with Gabor filters. A Gabor filter is a Fourier kernel multiplied with a Gaussian envelope,

$$G(x - x_0\,; \sigma, k_0) = e^{i\,k_0\,(x - x_0)}\, e^{-\frac{(x-x_0)^2}{2\sigma^2}}. \quad (1)$$

The Gabor filter is localized at $x = x_0$ in space and at $k = k_0$ in frequency space. A relation between the two free parameters σ and k_0 is imposed in a way that Gaussian envelope contains a complete wavelength λ in the interval $[x_0 - \sigma,\, x_0 + \sigma]$, $\lambda = \frac{2\pi}{3}\sigma \approx 2.1\sigma$.

The results of the one dimensional convolution along the lines of the images,

$$\begin{aligned} C_i(x; k_0) &= \int G(x - z\,; k_0) I_i(z)\, dz \\ &= \rho_i(x) e^{i\psi_i(x)} \end{aligned}$$

are complex functions characterized by amplitudes $\rho_i(x)$ and phases $\psi_i(x)$, index i (dropped in the following) indicates the left and right components.

As a function of the spatial position, it can be showed that the phase of the filter response, $\psi(x)$, has a quasi *linear* behavior dictated by the peak frequency k_0,

$$\psi(x) \approx \psi'(x_0)\,(x - x_0) \approx k_0\,(x - x_0).$$

Disparity is obtained expanding the phase difference, $\Delta\psi(x) = \psi_2(x) - \psi_1(x)$ to the second order in δ,

$$\delta(x) = \frac{[\Delta\psi(x)]_{2\,\pi}}{\psi'(x)}. \quad (2)$$

*Real World Computing Partnership

The phase is not defined when the amplitude vanishes, i.e. when $\rho(x) = 0$. Around these singular points, the phase is very sensitive to spatial or scale variations, and the calculation of disparity is unreliable.

However, the neighborhoods of singular points can be detected [1]. Distance from singularities can be expressed in terms of the spatial variation of the convolution functions $C_1(x)$ and $C_2(x)$ by means of the quantity

$$S(x) = \left| \frac{d}{dx} \log\left(C(x)\right) \right| \qquad (3)$$

Near singularities $S(x)$ go to infinity. Disparity calculation at point x is accepted only if this quantity is not greater than a confidence threshold.

Singuraties are isolated points in the scale space (x, λ). In order to decrease the area affected by singularities, filters of different scale may be used and a unique result can be obtained by means of a weighed sum of the individual responses (section 4.2).

3 The 2D Model: Optical Flow Estimation

Two dimensional shifts occur in the computation of the optical flow, in which 2D displacements between corresponding points of images taken at successive times, $I_1 = I(t_1)$ and $I_2 = I(t_2)$, have to be computed.

The two images are convoluted with a set of N_F two-dimensional Gabor filters characterized by different directions. Following Ref. [2], the entire set of directional Gabor filters is generated by rotating one of them around the origin, for example the one centered on the K_x axis. The filters, $\hat{G}_j(\vec{k}; \vec{k}_0^j, \vec{x})$, can be expressed by the formula

$$e^{i\,(\vec{k}-\vec{k}_0^j)\,\vec{x}} \cdot \exp\left[-\frac{1}{2}(\vec{k}-\vec{k}_0^j)^T \mathbf{A}_j (\vec{k}-\vec{k}_0^j)\right]$$

where $\vec{k}_0^j = k_0\,(cos\,\theta_j,\, sin\,\theta_j)$, and $\theta_j = \pi(j-1)/N_F$, with $j = 1, ..., N_F$. Matrix $\mathbf{A}_j = \mathbf{R}_j \mathbf{D} \mathbf{R}_j^T$ is the 2×2 matrix obtained by rotating the diagonal matrix $\mathbf{D}$, whose elements are the inverse of the square of the standard deviations along the radial and angular directions, by means of rotation matrices $\mathbf{R}_j$. Each filter $\hat{G}_j(\vec{x})$ has a preferential direction $(cos\theta_j,\, sin\theta_j)$ and a peak frequency $\vec{k}_0^j$.

The angular bandwidth depends on the number of filters. It should be chosen in such a way that all the directions are covered by the set of filters, without excessive overlapping. Like as one–dimesional case, a relation between the width along the filter's direction and the peak frequency, k_0, is imposed.

The convolution results are 2D complex functions, $C^j_{1,2}(x,y) = G_j * I_{1,2} = \rho^j_{1,2}\, e^{i\psi^j_{1,2}}$.

Expanding to the second order, the relation between the x and y displacements, δ_x and δ_y, and the phase difference, for each filter G_j, is given by $\Delta\psi^j(x,y) \approx \psi^j_x(x,y)\,\delta_x + \psi^j_y(x,y)\,\delta_y$.

The above set of equations can be solved at point (x, y) if the response of more than one filter is not singular. The definition of singular neighborhoods is the 2D generalization of (3). In general, if the results of several filters are available, the solution is chosen that minimizes the square error,

$$\sum_{j=1}^{N_F} w^j(\vec{x}) \left[\Delta\psi^j(\vec{x}) - \nabla\psi^j(\vec{x}) \cdot \vec{\delta}(\vec{x})\right]^2 .$$

The weight factors $w^j(\vec{x})$ measure the confidence in the response of the filter. A weight is set to zero if the corresponding filter has singular response.

4 Implementation on a Parallel DSP System

The vision processing system consists of five NEL TIM-40 modules and a NEL TIM-40 CFG frame grabber [4]. Each module has a TI-C40 DSP (50 MHz) [3] and two or five MB of memory. The frame grabber is a true color grabber with a 512 × 512 × 32 dual-ported framestore with an on board C40 (40 MHz). The TIMs are plugged-in to two NEL PC-2M motherboards for PC-AT bus.

One of the most important features of the C40 are his six communication ports at 20 MB/sec. of maximum transfer rate, each one with an independent DMA device that permits to overlap comunications and computation. Therefore the C40 can be utilized as basic building block for a powerful MIMD parallel machine showing complex topology and tailored for embedded systems.

An ANSI C compiler supplied with parallel libraries [5] and an optimized vectorial library [6] are used to develops programs.

The parallel configuration, sketched in Fig. 1, is a three stage pipeline: image acquisition, distributed elaboration, and combination of partial results and transfer to the host system. The visual signals from the two cameras are captured simultaneously by the frame grabber and the corresponding two image matrices that are sent concurrently to the four processors of level 2.

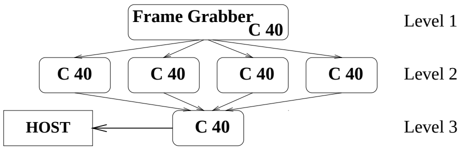

Figure 1: *The DSP system is a three-level pipeline: 1) image acquisition, 2)distributed processing, and 3) combination of partial results.*

The core of the computational procedure is subdivided among the four processors of level 2 to exploit geometrical parallelism. These processors perform independently and therefore they are not interconnected. Individual results are sent at approximately the same time to the processor of level 3, where they are recombined and made available for further processing and/or visualization.

All processes are organized in threads and the acquisition, the communication and the computation tasks are assigned to different, overlapping, threads. To minimize the *reaction time* of the system, i.e. the time the system takes to respond to a stimulus, the acquisition of a new image is enabled only when it can be processed with not delay across the pipeline stages (see Figure 2).

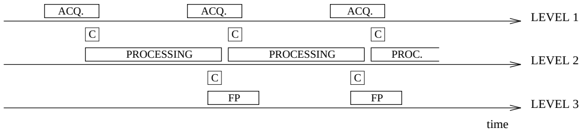

Figure 2: *Schematization of the computational procedure. Processes are indicated as: ACQ = image acquisition, PROCESSING = image processing, and C = communication, FP = further processing.*

4.1 Disparity Estimation

The signals from the two cameras are synchronized and images are captured simultaneously. The image pair, of size 256×256, is reduced to size 128×128 by subsampling, or by averaging four pixels into one. Averaging improves results' quality but increases the reaction time of the system.

The simplest and most flexible way to parallelize the algorithm is to exploit the data parallel structure. Each image is divided into four subimages 32×128 that are sent to the four processors in level 2. Each processor receives two stripes, one from the left image, the other from the right image. The computation performed by each processor is schematized as follows.

1. Each processor performs a one-dimensional FFT of the rows.
2. The Fourier transforms of the two subimages are multiplied by a selected Gabor filter and anti-transformed.
3. The derivatives of phase and amplitude along the x direction, are used to calculate $S(x)$. A signal-to-noise constraint on the filter energy is also added.
4. Position disparity is computed using formula (2), with the averaged phase derivative, $(\psi'_1(x) + \psi'_2(x))/2$, at the denominator.
5. The partial results from the four processors are patched together by the processor in level 3.

If several filters are used, the *processing cycle* for one filter (points 2, 3, 4 and 5) have to be repeated for each filter.

Processing times are displayed in Table 1

	T_A	T_P	T_T	T_F	T_C	$T_{Tot.}$
aver.	55	22	11	$N_F \cdot 60$	10	158
subs.	35	3	11	$N_F \cdot 60$	10	119

Table 1: *Processing times for computation of the disparity map of images of size* 128×128 *using 4 processors. Times are expressed in msec.* T_A *is the acquisition time for a pair of stereo images,* T_P *is the time required to prepare the images for processing,* T_T *is the time to perform direct Fourier transform on the stereo pair,* T_F *is the processing cycle time for one filter, and* T_C *is the time delay due to communications between processors (*T_C *is usually larger than the physical communication time because of synchronization delays).*

Acquisition, communications and computations are partially overlapping processes because they are performed in parallel. T_A is an average value because it depends on the synchronization between camera signals and processing cycle. T_F may vary, usually not more than 5%, depending on the type of image and illumination. Variations of the other times are not more than 2%.

The total processing time (reaction time) for one image, is given by $T_{TOT} = T_A + T_P + T_T + N_F T_F + T_C$. The time difference between two successive images is approximatively given by the processing time of level 2, $\Delta t \approx (T_P + T_T + N_F T_F)$.

Figure 3 show a disparity map obtained by applying one filter of wavelength $\lambda = 1.28$ to images of size 128×128. Disparity maps are represented as gray level images; singular regions are black.

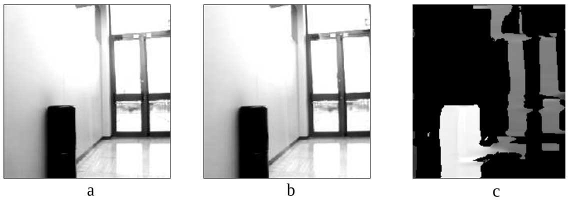

Figure 3: *a), b) Stereo pair. c) Disparity map.*

Algorithm performance depends on the choice of the filter width. Since disparity estimation relies on the matching between local regions, a filter cannot be expected to detect disparities larger than its width σ.

For real-time application a qualitative analysis and a fast reaction of the system is preferable to an accurate analysis and a slow response. Therefore, in our applications, we use one filter, whose width is adjusted dynamically, depending on the disparity values detected in the previous image.

4.2 Optical Flow

The estimation of the optical flow involves the computation of local two dimensional shifts between two images, $I_1 = I(t_1)$ and $I_2 = I(t_2 = t_1 + \Delta t)$, taken at successive times.

Since the problem is bidimensional, a set of directional filters must be used. With respect to the 1D algorithm, the processing time for a single filter is increased because 2D FFTs and 2D derivatives of amplitudes and phases need to be computed. Furthermore, for a fixed peak frequency k_0 a set of filters, $N_F \geq 2$, has to cover the direction spectrum. At the end, a linear system that integrates the responses of the individual filters has to be solved.

To achieve real-time processing, captured images, of size 128×128 , are reduced to size 64×64 by averaging four pixels into one. Each image is divided into 4 patches so that each processor of level 2 works on subimages of size 32×32 .

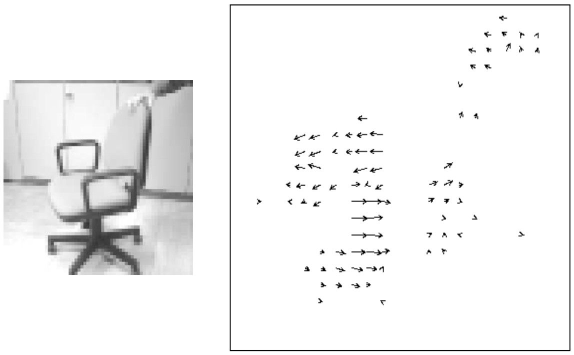

Figure 4: *Image of a spinning chair (*64×64*). On the right, the velocity field obtained with* $N_F = 5$.

Each processor performs the entire computation cycle for N_F directional filters. For each direction, the processing results are combined with those obtained for the previous image and memorized for the successive step. The last processor patches together the subimages from the four processors into a 64×64 velocity matrix, in which at each non-singular point (x, y) the values of the x and y displacements are given. The velocity at each point is then obtained dividing the displacement by the time interval Δt. An example of optical flow estimation is shown in Fig. 4.

Processing times and total processing times for one image ($T_{TOT} = T_A + T_P + T_T + N_F\, T_F + T_C$) are shown in Table 2 and Table 3. The time difference between two successive images is approximatively given by the cycle time of level 2, $\Delta t \approx (T_P + T_T + N_F\, T_F)$ (see Figure 2).

T_A	T_P	T_T	T_F	T_C	Size
47	2	3	N_F x 21	4	64 x 64

Table 2: *Processing times (msec) for estimation of the optical flow. T_A is the acquisition time for one image of size 128×128. (average value). Time T_P is the time to reduce the image size to 64×64. T_T is the time to perform a 32×32 FFT. T_F is the complete processing cycle time for each filter and T_C is the delay time due to communications among processors.*

N_F	$N_F = 2$	$N_F = 3$	$N_F = 4$	$N_F = 5$
T_{TOT}	96	117	138	158

Table 3: *Total processing times (msec).*

5 Discussion

The choice of phase-based techniques for image matching is dictated by several occurrences. The phase-based differential technique is more robust than amplitude-based techniques for small deformations and contrast changes between pairs of images Furthermore, the existence of an intrinsic confidence measure that estimates the reliability of computed results is an important property of the method.

One of the principal drawbacks of the phase-based technique is the high computational load. The processing time increases almost linearly with the number of filters and with the size of the image matrix. However, since computations are local, the algorithm can be easily parallelized.

In this work, the local nature of phase-based computations was exploited to realize a parallel vision processing system for depth and movement perception. Using a TI-C40 based DSP board with six computing processors, computation of disparity maps with one filter on images of size 128×128 can be performed at a rate of 7-8 images per second. Computation of velocity field on images of size 64×64 is performed using three (four) directional filters at a rate of 8 (7) images per second.

Acknowledgements

This work was supported by the Japanese Ministry of International Trade and Industry (RWC Program).

References

[1] D. J. Fleet, A. D. Jepson, and M. Jenkin. Phase–based Disparity Measurement. *CVGIP: Image Understanding*, 53(2):198–210, March 1991.

[2] D.J. Fleet and A.D. Jepson. Computation of Component Image Velocity from Local Phase Information. *International Journal of Computer Vision*, 5(1):77–104, 1990.

[3] Texas Instruments Inc. *TMS320C4x*, March 1993. User's Guide.

[4] National Engineering Laboratory. *TIM-40CFG-RGB V2.0*, August 1993. User Manual.

[5] 3L Ltd. *C40 Parallel C V1.1.1*, March 1994.

[6] Sinectoanalysis Inc. *Optimized DSP Vector Library For TMS320C40*, 1993. Reference Manual.