Phase-Correlation Guided Search for Realtime Stereo Vision

Alfonso Alba[*] and Edgar Arce-Santana[**]

Facultad de Ciencias, Universidad Autónoma de San Luis Potosí
fac@fc.uaslp.mx, arce@fciencias.uaslp.mx

Abstract. In this paper, we propose a new theoretical framework, which is based on phase-correlation, for efficiently solving the correspondence problem. The proposed method allows area matching algorithms to perform at high frame rates, and can be applied to various problems in computer vision. In particular, we demonstrate the advantages of this method in the estimation of dense disparity maps in real time. A fairly optimized version of the proposed algorithm, implemented on a dual-core PC architecture, is capable of running at 100 frames per second with an image size of 256 × 256.

Keywords: stereo vision, real time, phase correlation, area matching.

1 Introduction

Disparity estimation is a fundamental task in 3D stereo vision, whose goal is to determine the distance from the camera array to each of the objects in the scene. Disparity is defined as the difference in position between a given pixel in the reference image, and the corresponding pixel in the matching image. This means that, in order to obtain the disparity for each pixel, one must find a pixel-to-pixel correspondence between both images. This is a seriously ill-posed problem due to various reasons, such as: (1) occlusions, which are regions visible only to one of the cameras and occluded by other objects for the other camera; (2) homogeneous regions and periodic textures, where a pixel may be matched with many other pixels in the region; and (3) noise, which generates artificial differences between the images. State-of-the-art stereo vision techniques are designed to be robust to noise and provide disparity values for regions where the matching process fails, typically by introducing regularization constraints which enforce (piecewise) smoothness of the disparity map. Also, one can usually make certain assumptions which may drastically reduce the search space, for example, when the reference and matching images are adequately aligned, the epipolar lines are perfectly horizontal, thus corresponding points are always on the same row.

[*] A. Alba was supported by Grant PROMEP/103.5/09/573.

[**] E. Arce-Santana was supported by Grant PROMEP/103.5/04/1387.

P. Wiederhold and R.P. Barneva (Eds.): IWCIA 2009, LNCS 5852, pp. 212–223, 2009.

Many algorithms have been proposed to solve the correspondence problem, and most of them can be classified as global or local optimization methods. Global methods typically attempt to find the disparity field that minimizes a global energy function involving a term that measures the number and degree of mismatches, and a term which enforces spatial coherence by penalizing abrupt disparity changes. Several methods have been used to estimate the optimal disparity field, including: Markov random fields [18], hidden Markov measure fields (HMMF) [1], and belief propagation [24]. In contrast, local methods, which are typically less accurate than global methods, take a pixel or region in the reference image and perform a search along the epipolar line in the matching image to find the pixel or region which minimizes a local error measure, or maximizes a correlation measure between both areas. Among these methods, the most common approach relies on the estimation of a similarity measure between a window centered at pixel (x, y) in one of the images and a window centered at pixel $(x + d, y)$ in the other image, for all d in a given range. In a winner-takes-all (WTA) approach, the disparity assigned to pixel (x, y) is the value of d which maximizes the similarity measure. Variuos similarity measures have been proposed, including the sum of absolute differences (SAD), sum of square differences (SSD), statistical correlation, and rank measures. It is also possible to use multiple windows [11], or adaptive windows [12] to improve the estimations within homogeneous regions. An extensive review of various of these methods is provided by Scharstein et al. [23].

In recent years, state-of-the-art algorithms for stereo depth estimation have advanced significatively in terms of quality; however, some of these algorithms may take a significant amount of time (from a few seconds up to a few minutes) to process a single pair of stereo images, which seriously limits their applications in problems which require real-time processing, such as: robot and vehicle navigation, surveillance, human-computer interfaces, and augmented reality. This has led researchers to develop faster methods by taking advantage of parallel and/or reconfigurable architectures, and by implementing clever techniques to reduce artifacts at low computational expense. Various recently proposed algorithms and architectures are already capable of computing dense disparity estimations at video-rate. Most of these algorithms have some degree of parallelization; for example, on a modern PC computer it is possible to use single-instruction multiple-data (SIMD) extensions such as MMX [11] or SSE2 [4] [5], multi-core processing [26], or GPU implementations [6] [2]. Algorithms designed for reconfigurable architectures such as FPGA's are also relatively common [3] [10]. Other proposals assume that the location of the cameras is fixed with respect to the scene, so that they can perform a background/foreground segmentation, and compute the disparity only for those pixels that correspond to new or moving objects [15]. These methods achieve framerates which vary between 5 and 60 frames per second for relatively small images (e.g., 320 × 240 pixels), and between 3 and 30 fps for larger images (see [25] for a good comparison).

One technique that has not been fully exploited is the phase-correlation method, which relies on the fact that a displacement in the spatial domain is observed as a phase change in the frequency domain. Specifically, if $F(\omega)$ is the Fourier transform of $f(x)$, and $g(x) = f(x + d)$, then the Fourier transform of $g(x)$ is given by $G(\omega) = e^{j\omega d}F(\omega)$. One can then compute the normalized cross-spectrum $R(\omega)$ between F and G, which in this case is given by

$$R(\omega) = \frac{F(\omega)G^*(\omega)}{|F(\omega)G^*(\omega)|} = e^{-j\omega d}. \quad (1)$$

The inverse Fourier transform $r(x)$ of $R(\omega)$ is called the phase-only correlation (POC) function. In this case, one obtains $r(x) = \delta(x - d)$ (i.e., an impulse centered at $x = d$); therefore, one can easily recover the displacement by finding the position of the peak of $r(x)$. This technique is commonly used for image registration [21] [7] [13] [14], but it has also been applied in stereo vision [20].

In this paper we introduce a technique to reduce the search space in local methods, which is based on the estimation of multiple phase-correlation peaks. This results in a very efficient disparity estimation algorithm that is between 2 and 3 times faster than methods that use a full-search approach. The paper is divided as follows: Section 2 presents the basic algorithm and some quality refinements, in Section 3 we present some test results with both static and moving images, and finally, our conclusions are presented in Section 4.

2 Methodology

Consider two images $f(x, y)$ and $g(x, y)$ that represent the same scene in slightly different situations (e.g., taken at different times or from two distinct points of view). In general, the same objects, although possibly in different positions, will appear in both images. If the displacement is the same for all objects (including the background), the phase correlation function would approximate a delta function located at the displacement. If, however, different displacements are observed for different objects, the POC function will show various peaks, contaminated and distorted by noise due to borders, occlusions, and homogeneous regions. Intuitively, each of the peaks observed in the POC function may be related to the displacement of a specific object. The goal of this work is to experimentally show that the POC maxima provide useful information that can be used to implement more efficient area matching algorithms.

We are particularly interested in estimating the dense disparity map from a pair of stereo images with horizontal epipolar lines. In this case, corresponding pixels always lie in the same row, and thus the problem can be solved with a row-by-row scheme. In the rest of this paper we will assume that $f(x, y)$ and $g(x, y)$ are, respectively, the left and right images obtained from an horizontal stereo camera arrangement; also, the observation model is $g(x, y) = f(x + d(x, y), y)$, where $d(x, y)$ is the piece-wise smooth disparity field. Note that, under this model, the reference image is the right image $g(x, y)$; however, our methodology can be easily adapted to the case where $f(x, y)$ is the reference image. A typical winner-takes-all (WTA) full-search approach finds, for each pixel (x, y), the disparity value $d \in \{0, \ldots, D\}$ which minimizes a local error function $S(x, y, d)$ that measures the difference between a neighborhood of $g(x, y)$ and a similar-shaped neighborhood of $f(x + d, y)$. Therefore, the disparity $d(x, y)$ for each pixel is given as

$$d(x, y) = \arg \min_{d' \in \{0,\ldots,D\}} \{S(x, y, d')\} , \tag{2}$$

where D is a given disparity range (typically between 32 and 64). Two common error measures are the SAD and SSD, respectively defined as

$$S_{\mathrm{SAD}}(x, y, d') = \sum_{j=-w}^{w} \sum_{i=-w}^{w} \|f(x + d' + i, y + j) - g(x + i, y + j)\| , \tag{3}$$

and

$$S_{\mathrm{SSD}}(x, y, d') = \sum_{j=-w}^{w} \sum_{i=-w}^{w} \|f(x + d' + i, y + j) - g(x + i, y + j)\|^2 , \tag{4}$$

where w determines the size of the correlation window, and it directly affects the granularity of the results: small windows will produce noisy disparity fields, mainly due to ambiguities in homogeneous regions, whereas large windows may blur the borders and are more sensitive to projective distortions between the images. Larger windows may also significatively increase the computational cost, however, for fixed-size rectangular windows, it is possible to use cost-aggregation techniques to efficiently compute the error measure for a large number of pixels.

2.1 Phase-Correlation Guided Stereo Matching

Computation of the scores $S(x, y, d')$ is the most expensive step in a WTA-based stereo matching algorithm. The full-search approach, described above, requires the computation of D score values per pixel. To reduce the search space, we compute the phase-correlation function for each row, and estimate the most likely disparity values (for the pixels in that row) by finding the highest peaks of the POC function. Only those peaks whose value exceeds a given threshold (which, in our case, is zero) are considered.

The basic approach is summarized as follows: given a stereo pair of images $f(x, y)$ and $g(x, y)$, representing the left and right images, respectively, we estimate the disparity field $d(x, y)$ by performing these steps:

1. For each row y, do the following:
   (a) Compute the 1D FFT's $F_y(\omega)$ and $G_y(\omega)$ of the y-th row of $f(x, y)$ and $g(x, y)$, respectively.
   (b) Compute the normalized cross-spectrum $R_y(\omega)$, given by

$$R_y(\omega) = \frac{F_y(\omega)G_y^*(\omega)}{|F_y(\omega)G_y^*(\omega)|}. \tag{5}$$

   (c) Compute the POC function $r_y(x)$ as the 1D IFFT of $R_y(\omega)$.

(d) Find the M most significative positive maxima of $r_y(x)$ in the interval $x \in [0, D]$ where D is the disparity range. Let $d_1, d_2, \ldots, d_M$ be the positions of these maxima; that is, the most likely disparity values for row y. In particular, we have obtained good results with M between $D/4$ and $D/3$.

(e) For each pixel x in the row, find the disparity $d(x, y)$ as

$$d(x, y) = \arg \min_{k \in \{1, \ldots, M\}} S(x, y, d_k). \tag{6}$$

Note that, in step (e) of the algorithm, the search is performed only over the M candidates, and not across the full disparity range as in other methods.

2.2 Regularization and Collision Detection

Reduction of the displacement search space induces a certain quantization of the disparity values. Under certain smoothness assumptions, this quantization enforces regularization constraints in the disparity field, but only along the horizontal direction. Because of this, the basic algorithm may suppress discontinuities at the borders of the objects, causing some artifacts in the form of streaks of incorrect disparity values along the scanline. One way to reduce these artifacts is to enforce inter-line consistence in the set of likely disparities. To achieve this, we first compute the POC function $r_y(x)$ for all rows and reorganize it as an image $r(x, y)$, and then we apply a 1D low-pass filter to $r(x, y)$ along the Y-axis. One can then find the maxima corresponding to each row of the filtered POC image. Note that this regularization filter is applied not to the final disparity map, but to the phase correlation function that determines the disparity candidates for each scanline, in order to induce inter-line consistence. We have obtained good results using a Gaussian low-pass filter with standard deviation between 3 and 16. On the other hand, applying a median filter to the estimated disparity map may improve the quality of the results with little computational cost.

We have also implemented a simple detection stage for invalid disparities due to collision, based on the same idea used by Di Stefano et al. [4]. This idea relies on the uniqueness constraint, which states that each point in the 3D scene will be projected to at most one point in each image of the stereo pair. Therefore, if pixel $g(x, y)$ is matched with pixel $f(x+d, y)$, and later, pixel $g(x', y)$ is matched with $f(x' + d', y)$, so that $x + d = x' + d'$, then at least one of those matchings is wrong. We simply mark both matchings as invalid; however, in the future we expect to implement a mechanism to decide which is the correct matching, and possible recompute invalid matchings. In any case, due to the quantization induced in the disparity field by the search space reduction, the number of invalid matchings produced by our algorithm is relatively low.

2.3 Implementation Details

We have implemented the proposed method in a PC platform with several optimizations to achieve high frame rates. The most relevant algorithmic optimizations are:

- Since the images f and g are real, half of the coefficients of their Fourier transforms and the normalized cross spectrum are redundant, thus one can only compute the coefficients corresponding to non-negative frequencies. This reduces the POC computation time roughly by half.
- Computation of error scores (SAD and SSD measures) is performed by cost accumulation. Specifically, for a given row y, we first compute the partial errors $C(x, y, d)$ corresponding to the columns of the correlation window; for example, when using the SAD measure, the column errors are given by

$$C(x, y, d) = \sum_{j=-w}^{w} |f(x + d, y + j) - g(x, y + j)|. \tag{7}$$

  Error scores $S(x, y, d)$ for all x are then simply computed as

$$S(x, y, d) = S(x - 1, y, d) - C(x - 1 - w, y, d) + C(x + w, y, d). \tag{8}$$

- Median filter computation is performed efficiently using the algorithm described in [17].

Our implementation also takes advantage of parallelization provided by the Intel Core 2 Duo platform. Specifically, the most notable platform-dependant optimizations are:

- FFT's are performed using the FFTW library [8], which includes SSE2 optimizations and efficient algorithms for real-valued input data.
- Computation of error scores and matching is performed in parallel under a multi-core processor, where each core processes a given range of rows.
- SSE2 extensions used for column error estimation and POC image filtering, allowing the algorithm to process up to 8 contiguous pixels in parallel.

A faster algorithm was also implemented for input images whose width is a power of 2. In this case, the FFTW algorithms are particularly efficient, and any multiplication by the image width can be replaced by a binary shift.

It is worth mentioning that our implementation could be further optimized by using SSE2 and multi-core processing during the estimation of the normalized cross-spectrum, and the median filter stages, both of which are implemented using plain C++ code.

3 Results

To determine the quality and computational efficiency of our method under different circumstances, we have performed a series of tests using both static and moving images. In the case of static images, the ground truth is also known, which permits us to obtain quantitative quality measurements. All tests were performed under an Intel Pentium Core 2 Duo workstation running at 2.4 GHz with 2 Gb of RAM.

Table 1. Percentages of disparity values with error above 1 pixel, measured over different image regions (non-occluded, all regions, and near discontinuities). These scores were computed by the Middlebury evaluation system.

	Tsukuba	Venus	Teddy	Cones
Non-occluded	7.86	6.06	37.0	22.5
All regions	9.78	7.65	43.3	31.0
Discontinuities	29.1	45.2	50.1	42.3

Table 2. Percentages of correct, incorrect, and invalid matchings for various real-time algorithms

Tsukuba scene

Method	% Correct	% Incorrect	% Invalid
SAD5 [11]	85.12	4.56	10.32
SMP [4]	60.06	30.62	9.32
Hierarchical [16]	95.93	4.07	—
Hardware LWPC [3]	75.10	24.90	—
Our method	86.08	8.79	5.13

Venus scene

Method	% Correct	% Incorrect	% Invalid
SMP [4]	93.79	4.19	2.02
Hardware LWPC [3]	89.49	10.51	—
Our method	87.40	8.67	3.93

3.1 Results with Static Images

We have applied our algorithm to four stereo pairs (Tsukuba, Venus, Teddy, and Cones) obtained from the Middlebury database [23] [22], and submitted the resulting maps to the Middlebury evaluation system. In all cases, the search space is reduced to only $M = 16$ disparity values per row. Results from this evaluation are shown in Table 1. Note that our scores are comparable to some of the worst-performing methods reported in the Middlebury database; however, most of those methods are designed for quality instead of speed, and only a few are able to perform in real-time. In any case, this means that, in terms of quality, our method is roughly comparable to other recently published algorithms. To perform a more fair assessment, we have compared our algorithm against other real-time oriented methods using the Tsukuba and Venus scenes. In this case, we compute the percentage of correct disparity estimations, incorrect disparity estimations (error above 1), and invalid disparities (due to collisions), and compare them with the results reported for the other algorithms. These scores are shown in Table 2. Note that, in the case of the Tsukuba scene, only the Hierarchical algorithm by Kim et al. performs significantly better than our method; however, this algorithm is based on a background/foreground segmentation, which requires the camera to be in a fixed position, and re-computation of the background disparities takes up to a few seconds. On the other hand, our method clearly does not perform so well

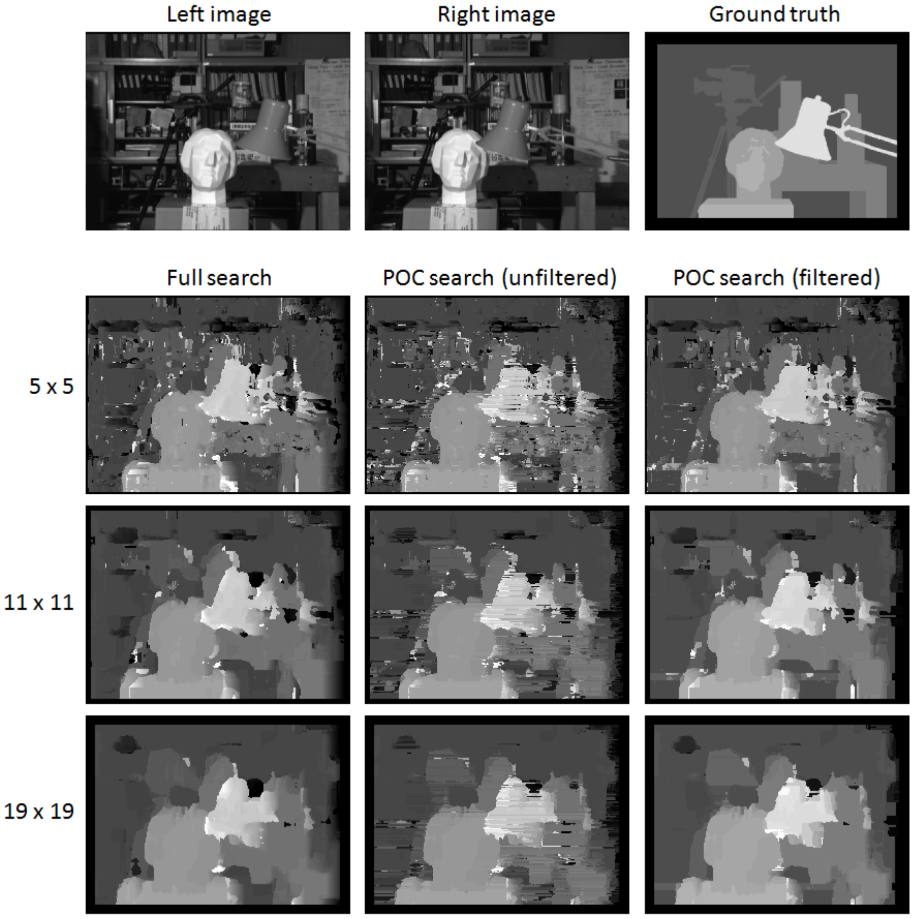


Fig. 1. Disparity estimation for the Tsukuba scene. The top row shows the left and right images, and the true disparity map. Rows 2-4 show the estimated disparity maps using an area-based WTA approach with the SAD measure computed for different window sizes (5×5, 11×11, and 19×19). The left-most column shows the result using a full-search approach over a disparity range of 48 pixels. The middle column shows the results using the basic POC-driven search with $M = 12$ maxima. The right-most column shows the results when a Gaussian filter with a standard deviation of 10 pixels is applied to the POC along the Y-axis.

Table 3. Average computation times for the estimated disparity maps shown in Figure 1. The image size is 256 × 192.

	Full search	**POC search**	**POC Filter**	**POC + median filters**
5 × 5	14 ms	5 ms	6 ms	8 ms
11 × 11	17 ms	6.5 ms	7 ms	9 ms
19 × 19	22 ms	7.5 ms	8 ms	10 ms

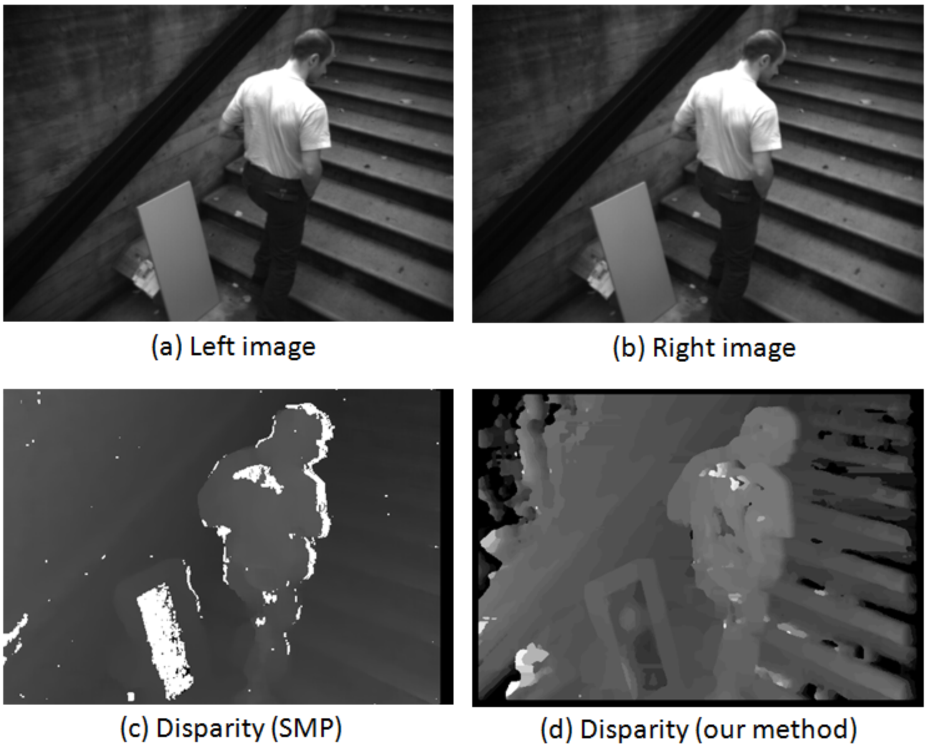
with the Venus scene; this is because the objects in the Venus scene form oblique planes whose disparity varies smoothly and cannot be correctly estimated using only 16 values.

We have also measured the computation time for the Tsukuba scene varying the size of the correlation window and the type of regularization used. In this case we used a 256 × 192 sub-sampled version of the Tsukuba images, which allows us to use our more efficient algorithm. Figure 1 shows the results obtained for the 256 × 192 Tsukuba scene with varying correlation window sizes, using both full and POC-guided search (with $M = 12$). Performance results for these tests are shown in Table 3.

3.2 Results with Video Sequences

We have also tested our method with some of the real video sequences used by Di Stefano et al. in their 2004 paper [4] [19]; in particular, we tested the rectified 640×480 “Indoor” and “Outdoor” sequences. Figure 2 shows the results obtained for frame 0050 of the Outdoor sequence using both Di Stefano’s SMP method (where white areas represent invalid matchings), and our algorithm. The true disparities for these sequences are unknown, thus quality evaluation can only be performed from a qualitative perspective. In our opinion, both algorithms perform similarly: the main objects in the scene and their relative distance to the cameras can be distinguished, but there are some clear mismatches in homogeneous such as the wood board, and the person’s shoulders, and in occluded regions such as object borders.

Finally, we present in Table 4, a comparison of the computational performance of our method against various state-of-the-art real-time algorithms. For this comparison to be meaningful, we have included a column which describes the platform where each algorithm was implemented, as well as the size and disparity range of the test images. Still, this comparison is not completely fair, since some of the algorithms include extra steps such as image rectification, and sub-pixel interpolation; however, the computational cost of these refinements is relatively small. For example, the S^3E algorithm proposed by Zinner et al. [26], which also runs in an Intel Core 2 Duo platform and takes advantage of dual-core and SSE2 optimizations, includes various quality refinement steps and a 3D reconstruction stage; however, according to their results, these extra steps account only for 11% of the total computation time (for an image size of 450 × 375). Therefore, even if one removed the extra steps from the S^3E system, the POC-driven search method would still be roughly twice as fast.


(a) Left image

(b) Right image

(c) Disparity (SMP)

(d) Disparity (our method)

Fig. 2. Results on frame 0050 of the "Outdoor" sequence: (a) rectified left image, (b) rectified right image, (c) disparity map obtained with the SMP method [4], (d) disparity map obtained with our method. In both cases, the disparity range is 48 pixels and the size of the correlation window is 15 × 15.

Table 4. Performance comparison between our POC-based implementation and other recent methods. Note that some of these methods may include additional processes such as frame grabbing from the video cameras, and image rectification.

Method	**Platform**	**Image size**	**Range**	**Rate**
SMP [4]	Intel Pentium III, 800 MHz	320 × 240	32	31.25 fps
SMP [4]	Intel Pentium III, 800 MHz	640 × 480	48	5.77 fps
LWPC [3]	Xilinx Virtex2000E FPGAs	256 × 360	20	30 fps
[9]	ATI Radeon X800 GPU	288 × 216	40	17.8 fps
[6]	GeForce 8800 Ultra GPU	320 × 240	64	13 fps
[6]	GeForce 8800 Ultra GPU	640 × 480	128	4.2 fps
S^3E [26]	Intel Core 2 Duo, 2.0 Ghz	320 × 240	30	42 fps
S^3E [26]	Intel Core 2 Duo, 2.0 Ghz	480 × 360	50	13 fps
Our method	Intel Core 2 Duo, 2.4 Ghz	256 × 256	48	110 fps
Our method	Intel Core 2 Duo, 2.4 Ghz	512 × 360	48	38 fps

4 Conclusions

The phase-correlation technique is commonly used to determine the geometrical translation that may exist between two images. In general, however, each object in the scene may suffer a different displacement in one image with respect to the other. In this case, the phase-correlation function does not approximate a delta function, but shows instead several peaks, some of which correspond to the displacements corresponding to the different objects. This information can be used in a stereo vision algorithm to reduce the search space in a local area matching approach, simply by finding a set of likely disparity candidates for each row of the images. This induces a quantization of the disparity values in the image, which leads to undesirable artifacts; however, these artifacts can be significatively reduced by smoothing the phase-correlation function along the vertical axis.

Preliminary results show that a well-optimized implementation of this technique may yield disparity estimations which are at least as accurate as some of the recently published methods, but in only a fraction of the time. On an Intel Core 2 Duo workstation running at 2.4 GHz, our implementation is able to process 100 frames of size 256×256, or 35 frames of size 512×512 per second. This leaves enough processing power for additional steps which may be necessary in a real-time computer vision application, such as: image rectification, sub-pixel refinements, object segmentation, and motion estimation.

Future research will focus on: (1) improving the quality of the disparity estimations, (2) using these techniques in an augmented reality application, and (3) applying the POC-search matching approach to other problems in computer vision, such as motion estimation. In particular, we are currently investigating various ways to refine the disparity measurements to obtain sub-pixel accuracy and recover bad matchings due to collisions. We are also developing a fast multi-channel segmentation method for real-time object detection in stereo video sequences.

References

1. Arce, E., Marroquin, J.L.: High-precision stereo disparity estimation using HMMF models. Image and Vision Computing 25, 623–636 (2007)
2. Bartczak, B., Jung, D., Koch, R.: Real-time neighborhood based disparity estimation incorporating temporal evidence. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 153–162. Springer, Heidelberg (2008)
3. Darabiha, A., MacLean, W.J., Rose, J.: Reconfigurable hardware implementation of a phase-correlation stereo algorithm. Machine Vision and Applications 17(2), 116–132 (2006)
4. Di Stefano, L., Marchionni, M., Mattoccia, S.: A fast area-based stereo matching algorithm. Image and Vision Computing 22, 983–1005 (2004)
5. Di Stefano, L., Marchionni, M., Mattoccia, S.: A pc-based real-time stereo vision system. Machine Graphics & Vision 13(3), 197–220 (2004)
6. Ernst, I., Heiko, H.: Mutual information based semi-global stereo matching on the GPU. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.-M., Monroe, L. (eds.) ISVC 2008, Part I. LNCS, vol. 5358, pp. 228–239. Springer, Heidelberg (2008)

7. Foroosh, H., Zerubia, J.B., Berthod, M.: Extension of phase correlation to subpixel registration. IEEE Trans. PAMI 11(3), 188–200 (2002)
8. Frigo, M., Johnson, S.G.: FFTW Web Page (2009), `http://www.fftw.org`
9. Gong, M.: Enforcing temporal consistency in real-time stereo estimation. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 564–577. Springer, Heidelberg (2006)
10. Han, S.K., Jeong, M.H., Woo, S., You, B.J.: Architecture and implementation of real-time stereo vision with bilateral background subtraction. In: Huang, D.-S., Heutte, L., Loog, M. (eds.) ICIC 2007. LNCS, vol. 4681, pp. 906–912. Springer, Heidelberg (2007)
11. Hirschmüller, H., Innocent, P.R., Garibaldi, J.: Real-time correlation-based stereo vision with reduced border errors. Int. J. Comput. Vision 47(1-3), 229–246 (2002)
12. Kanade, T., Okutomi, M.: A stereo matching algorithm with an adaptive window: Theory and experiment. IEEE Trans. PAMI 16(9), 920–932 (1994)
13. Keller, Y., Averbuch, A., Moshe, I.: Pseudopolar-based estimation of large translations, rotations, and scalings in images. IEEE Trans. on Image Processing 14(1), 12–22 (2005)
14. Keller, Y., Shkolnisky, Y., Averbuch, A.: The angular difference function and its application to image registration. IEEE Trans. PAMI 27(6), 969–976 (2005)
15. Kim, H., Min, D.B., Choi, S., Sohn, K.: Real-time disparity estimation using foreground segmentation for stereo sequences. Optical Engineering 45(3), 037402–1–037402–102 (2006)
16. Kim, H., Min, D.B., Sohn, K.: Real-Time Stereo Using Foreground Segmentation and Hierarchical Disparity Estimation. In: Ho, Y.-S., Kim, H.-J. (eds.) PCM 2005. LNCS, vol. 3767, pp. 384–395. Springer, Heidelberg (2005)
17. Kopp, M., Purgathofer, W.: Efficient 3x3 median filter computations. Technical University, Vienna (1994)
18. Lorenzo, J.T., Portillo, J., Alberola-López, C.: A novel markovian formulation of the correspondence problem in stereo vision. IEEE Trans. on Systems, Man and Cybernetics - Part A: Systems and Humans 34(3), 428–436 (2004)
19. Mattoccia, S.: Real-time stereo vision based on the uniqueness constraint: experimental results and applications (2004), `http://www.vision.deis.unibo.it/smatt/stereo.htm`
20. Muquit, M.A., Shibahara, T., Aoki, T.: A High-Accuracy Passive 3D Measurement System Using Phase-Based Image Matching. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. E89-A(3), 686–697 (2006)
21. Reddy, B.S., Chatterji, B.N.: An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration. IEEE Trans. on Image Processing 5(8), 1266–1271 (1996)
22. Scharstein, D., Szeliski, R.: Middlebury stereo vision page (2001), `http://vision.middlebury.edu/stereo/`
23. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision 47, 7–42 (2002)
24. Sun, J., Zheng, N.N., Shum, H.Y.: Stereo matching using belief propagation. IEEE Trans. PAMI 25(7), 787–800 (2003)
25. van der Mark, W., Gavrila, D.M.: Real-time dense stereo for intelligent vehicles. IEEE Trans. on Intelligent Transportation Systems 7(1), 38–50 (2006)
26. Zinner, C., Humenberger, M., Ambrosch, K., Kubinger, W.: An optimized software-based implementation of a census-based stereo matching algorithm. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Porikli, F., Peters, J., Klosowski, J., Arns, L., Chun, Y.K., Rhyne, T.-M., Monroe, L. (eds.) ISVC 2008, Part I. LNCS, vol. 5358, pp. 216–227. Springer, Heidelberg (2008)