

Enhanced Phase Correlation for Reliable and Robust Estimation of Multiple Motion Distributions

Matthias Ochs¹(✉), Henry Bradler¹, and Rudolf Mester^{1,2}

¹ VSI Lab, Goethe University, Frankfurt am Main, Germany
{ochs,bradler,mester}@vsi.cs.uni-frankfurt.de

² CVL, ISY, Linköping University, Linköping, Sweden

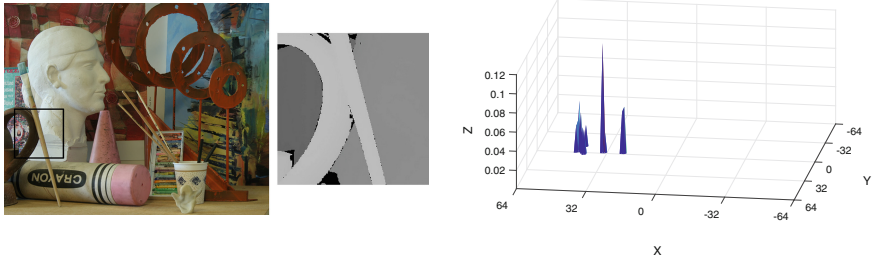
Abstract. Phase correlation is one of the classic methods for sparse motion or displacement estimation. It is renowned in the literature for high precision and insensitivity against illumination variations. We propose several important enhancements to the phase correlation (PhC) method which render it more robust against those situations where a motion measurement is not possible (low structure, too much noise, too different image content in the corresponding measurement windows). This allows the method to perform self-diagnosis in adverse situations. Furthermore, we extend the PhC method by a robust scheme for detecting and classifying the presence of multiple motions and estimating their uncertainties. Experimental results on the Middlebury Stereo Dataset and on the KITTI Optical Flow Dataset show the potential offered by the enhanced method in contrast to the PhC implementation of OpenCV.

Keywords: Optical flow · Motion estimation · Phase correlation

1 Introduction

Phase Correlation (PhC) is one of the four classical methods for local motion estimation, together with *discrete matching* (a.k.a. block matching), *differential matching* and *spatio-temporal optical flow measurement* (structure tensor). Although the fundamentals of PhC date back to the 1970s [2,4,5], the precise relations between the listed families of approaches have not been analyzed thoroughly in the literature so far. Depending on the characteristics of the data to be processed and to some degree also depending on the scientific community which is regarded (computer vision, geophysical data analysis, time delay estimation, ...), different families of methods are preferred for the task of estimating displacements or 2D motion. For instance, [6,7] are early papers proposing the normalized cross correlation metric. How to optimize the metric is another issue. *Discrete matching* is opposed to *differential approaches* lead by the classic Lucas&Kanade approach [1].

Since Fourier transform is the main element of the PhC method, it shows strong robustness against geometrical and photometric distortions [8–10].



(a) Cell position and disparity map of this cell (b) Thresholded delta array

Fig. 1. Exemplary result of our enhanced PhC on data from the Middlebury Stereo Dataset. Three different motions (depths) are present in the marked cell (brush, bust and background), causing 3 distinct peaks in the delta array.

Like the classical differential matching schemes, the PhC method, with certain extensions, can achieve subpixel matching accuracy [11]; accuracy of better than $1/100$ pixel was claimed by [12, 13]. Additional details regarding different variants of PhC algorithms can be found in [10, 14–17, 21, 23]. Some of them describe different ways to achieve subpixel matching accuracy, some others emphasize the advantages of PhC for estimating homogenous displacements for larger images (image registration). Some recent papers considered the use of PhC-based stereo algorithms for remote sensing tasks applied to aerial imagery [18] and for interferometric SAR image co-registration [19]. We refer also to [20] where the PhC stabilizes video sequences against illumination changes and camera shaking. Besides some completely novel approaches, we extend in the present paper several ideas that appear already in [20] and put them on a more systematic basis.

We emphasize that the method presented here does *not* aim at the computation of dense motion fields, but a) makes the classical PhC robust, and b) extends the PhC method towards being able to obtain *distributions* of motion vectors that appear in a given patch. In applications where the patch is assumed to be subjected to a homogeneous translation motion (image registration), this is already the desired result, whereas for complex motion fields these distributions give valuable *prior information* that allows to systematically *initialize* and *guide* a subsequent sparse or (semi-)dense motion estimation procedure.

2 Approach

This section embeds the plain PhC method as it is described in the literature into a framework that checks for potential problematic situations (due to invalid or ambiguous input data) and performs a series of self-checks and filtering steps that are necessary to employ the method in an autonomous mode without user intervention. We provide solid and proven procedures for tuning the different parameters that appear in the enhanced PhC method. The presentation of the PhC method and the proposed extensions are described here for one-dimensional signals; the generalization to more dimensions is straightforward.

Let $y[x_n]$ and $z[x_n]$ be two observations of the same discrete signal $s[x_n]$, where $z[x_n]$ contains a shift by a displacement d :

$$y[x_n] = s[x_n] \quad \text{and} \quad z[x_n] = s[x_n] * \delta[x_n - d] = s[x_n - d]. \tag{1}$$

The orthonormal Fourier transform over a discrete area of size N yields:

$$Y[f_k] = S[f_k] \quad \text{and} \quad Z[f_k] = S[f_k] \cdot \frac{1}{\sqrt{N}} \cdot \exp(-2\pi i \cdot \frac{f_k \cdot d}{N}). \tag{2}$$

For further examination, we isolate the displacement and frequency dependent phase shift between the two signals and introduce the cross-power spectrum $P[f_k]$ and its inverse Fourier transform, the *delta array* $p[x_n]$:

$$P[f_k] \stackrel{\text{def}}{=} \frac{Z[f_k] \cdot Y^*[f_k]}{|Z[f_k]| \cdot |Y[f_k]|} = \exp(-2\pi i \cdot \frac{f_k \cdot d}{N}), \tag{3}$$

$$p[x_n] \stackrel{\text{def}}{=} \mathcal{F}^{-1}(P[f_k]) = \sqrt{N} \cdot \delta[x_n - d]. \tag{4}$$

The delta array $p[x_n]$ consists of an ideal δ -impulse which indicates the relative shift between the two signals $y[x_n]$ and $z[x_n]$. In a realistic setting, with noise¹, multiple motions² and without periodicity of the images³, the delta array is more complex and needs to be analyzed in detail to obtain reliable results.

In the following Sects. 2.1 — 2.4 we introduce several checks and filtering steps which must be performed to let the PhC actually yield reliable and precise results. Steps which need to be applied separately for both patches ($y[x_n]$, $Y[f_k]$ or $z[x_n]$, $Z[f_k]$) are only denoted for the first patch (second patch accordingly).

2.1 Structure Check

First we check if both image patches show sufficient structure to allow the displacement estimation. We compute the gray scale variance of the patches in a weighted manner using the weights $w[x_n]$ of the anti-leakage window:

$$\hat{\sigma}^2 = \left(\sum_{n=1}^N w[x_n] \right)^{-1} \sum_{n=1}^N w[x_n] \cdot (y[x_n] - \hat{\mu})^2 \quad \text{with} \tag{5}$$

$$\hat{\mu} = \left(\sum_{n=1}^N w[x_n] \right)^{-1} \sum_{n=1}^N w[x_n] \cdot y[x_n]. \tag{6}$$

Then we compare it against a threshold τ_1 which was experimentally determined:

$$\hat{\sigma}^2 \begin{matrix} \text{structured} \\ \geq \\ < \\ \text{unstructured} \end{matrix} \tau_1. \tag{7}$$

¹ $y[x_n] \rightarrow y[x_n] + u[x_n]$, where $u[x_n]$ is assumed to be $\mathcal{N}(0, \sigma_u^2)$ i.i.d.

² Due to independent motions within the image or geometric effects (e.g. zoom).

³ $y[x_n] \rightarrow w[x_n] \cdot y[x_n]$, where $w[x_n]$ is an anti-leakage window (e.g. Tukey).

In our experiments with different datasets⁴ we found $\tau_1 \approx 90$ to be a good threshold to distinguish between structured and unstructured patches. Of course this value varies with the noise level of the input images. Due to the normalization of the weights, $w[x_n]$, it is independent of the chosen patch size.

2.2 Spectral Significance Filtering

After the transition to the frequency domain, we need to identify those significant spectral coefficients $Y[f_k]$ and $Z[f_k]$ which represent the main structure of the image patches and thus allow us to determine the displacement d . Therefore we need to suppress the influence of the DC ($f_k = 0$) spectral component of the signal (mean value compensation) as well as the components whose spectral magnitudes are dominated by noise (noise suppression).

Mean Compensation. Since most of the structural information of the image is encoded in the low frequency AC ($f_k \neq 0$) spectral components, it is important to compensate for the gray scale mean *before* the anti-leakage window $w[x_n]$ is applied. Otherwise, these low frequency components would be superimposed by the gray scale mean of the original image patch when the convolution with the Fourier transform of the anti-leakage window $w[x_n]$ is performed⁵.

$$Y[f_k] = \mathcal{F}(w[x_n] \cdot (y[x_n] - \langle y[x_n] \rangle)). \tag{8}$$

Noise Suppression. We also need to suppress those spectral components of $Y[f_k]$ and $Z[f_k]$ whose magnitudes are in the order of magnitude of the noise floor because their phases are only dominated by noise and do not contain any information. To do so, we compute the frequency distributions of $|Y[f_k]|$ and $|Z[f_k]|$ and look for the first interval which is mainly dominated by noise. For a fast approximation, we compute the mean τ_2 of those magnitudes which lie in the smaller half of the frequency distribution. Generally τ_2 might be too large, but this is negligible.

$$|Y[f_k]| \begin{matrix} \text{significant} \\ > \\ < \\ \text{insignificant} \end{matrix} \tau_2. \tag{9}$$

2.3 Delta Array Check

After significance filtering has been applied, the cross-power spectrum $P[f_k]$ and the delta array $p[x_n]$ are computed for significant components (see Eqs. 3 and 4). The inverse Fourier transform is an orthonormal transformation, thus:

⁴ We used a 0.5 Tukey window on Middlebury Stereo and KITTI Optical Flow datasets.

⁵ Convolution theorem: $\mathcal{F}(w[x_n] \cdot y[x_n]) = \mathcal{F}(w[x_n]) * \mathcal{F}(y[x_n]) = W[f_k] * Y[f_k]$.

$$\sum_{k=1}^N |P[f_k]|^2 = \overbrace{\quad}^{\text{no. of significant componentens } N_{\text{sig}}} = \sum_{n=1}^N |p[x_n]|^2 \leq N. \quad (10)$$

In an ideal case all the energy should concentrate on one δ -impulse which represents the displacement d . Hence we are only interested in those values of $p[x_n]$ which hold a significant amount of the energy known in beforehand (see Eq. 10) and thus represent a dominant motion. The other values which possess a much lower energy are suppressed by computing a threshold τ_3 based on the histogram of the distribution of $|p[x_n]|^2$. We set the histogram range to $[0, N_{\text{sig}}]$ (see Eq. 10), the number of bins to the geometric mean m_{win} of the lengths of the window and the right border of the first bin to be τ_3 . In our experiments we verified that energies which represent a relevant motion always lie above this threshold. This check fails if the energies of *all* spectral components are below τ_3 .

$$|p[x_n]| \begin{matrix} \text{significant} \\ > \\ < \\ \text{insignificant} \end{matrix} \tau_3 = \sqrt{\frac{N_{\text{sig}}}{m_{win}}} \quad (11)$$

2.4 Delta Array Clustering

So far, the tests were described for the one-dimensional case, but for the next check we need the actual two-dimensional representation of the signal. Therefore the delta array is written as $p[\mathbf{x}_n]$. In the absence of noise, the inverse Fourier transform of $P[f_k]$ contains a single δ peak, or multiple δ peaks in case of multiple motion. For real data, this / these peak(s) get smeared out and there will be some background noise in the delta array. Hence we only examine the significant (Eq. 11) values of the delta array $p[\mathbf{x}_n]$. We define the sets:

$$\mathcal{X} \stackrel{\text{def}}{=} \{\mathbf{x}_n : |p[\mathbf{x}_n]| > \tau_3\} \text{ and } \mathcal{P} \stackrel{\text{def}}{=} \{|p(\mathbf{x}_n)| : \mathbf{x}_n \in \mathcal{X}\}, |\mathcal{X}| = |\mathcal{P}| = N_{\text{sig}}. \quad (12)$$

These two sets will serve as input for a weighted K -means clustering algorithm.

Initial Phase. The first mean chosen is the point with the largest weight. We iteratively determine $K - 1$ more candidates as the ones with the largest cumulative euclidean distance to the already chosen ones. A set of K covariance matrices Σ_k is initialized as two dimensional identity matrices.

Labeling Phase. For each point we calculate the Mahalanobis distance to the current K means \mathbf{m}_k and assign the point to the cluster of the mean with minimum distance. Subsequently, we compute means and covariance matrices of the updated clusters using the values of the delta array as weights. This is repeated until either the clusters converge or a predefined maximum number of iterations is reached.

Algorithm 1. weighted k-means using Mahalanobis distance

```

1: function GET_LABELS( $k, \{\mathbf{x}_n\}_{n=1, \dots, N_{\text{sig}}}, \{p(\mathbf{x}_n)\}_{n=1, \dots, N_{\text{sig}}}$ )
2:    $\{\mathbf{m}_k\}_{k=1, \dots, K} \leftarrow \text{INITIALIZE\_WEIGHTED\_MEANS}(\{\mathbf{x}_n\}, \{p(\mathbf{x}_n)\})$ 
3:    $\{\Sigma_k\}_{k=1, \dots, K} \leftarrow \text{INITIALIZE\_COVARIANCE\_MATRICES}()$ 
4:   repeat
5:     for  $n \in \{1, \dots, N_{\text{sig}}\}$  do
6:        $\{d_k\}_{k=1, \dots, K} \leftarrow \text{GET\_MAHALANOBIS\_DISTANCES}(\mathbf{x}_n)$ 
7:        $\text{label}_n \leftarrow \arg \min_k \{d_k\}_{k=1, \dots, K}$ 
8:     end for
9:      $\text{UPDATE\_WEIGHTED\_MEANS}()$ 
10:     $\text{UPDATE\_COVARIANCE\_MATRICES}()$ 
11:   until LABELS_CONVERGED() or MAX_ITER_REACHED()
12:   return  $\{\mathbf{m}_k\}_{k=1, \dots, K}, \{\Sigma_k\}_{k=1, \dots, K}$ 
13: end function

```

This algorithm returns K means \mathbf{m}_k and covariance matrices Σ_k which describe the distribution within each cluster. To find the optimal K , the algorithm is run for different values of K and a cost function which sums up the areas of the covariance ellipses and penalizes large values of K (Occam's razor) is minimized:

$$K_{opt} = \arg \min_K \sum_{k=1}^K \det(\Sigma_k) + a \cdot \exp(b \cdot K). \quad (13)$$

We determined the values of the parameters in our experiments to be

$$a \approx 2.5 \cdot \det(\Sigma_0) \quad \text{and} \quad b \approx 0.5, \quad (14)$$

where $\det(\Sigma_0)$ is the area of the covariance ellipse in the case of only one cluster ($K = 1$).

2.5 Multiresolution

Since the estimation of a relative displacement of the signal in two regarded patches is limited by the patch size⁶ and works best when most of the image content is present in both patches, the previously presented steps are performed iteratively on different resolution scales of the image. We employed a Gaussian pyramid with two levels and a scaling of 2 for each image dimension. We used the same patch size on both pyramid levels, performed a first motion estimation on the upper (=lower resolution) pyramid level and transferred the result to the original scale by shifting the patch windows relative to each other according to the (correctly scaled) motion vector determined in the upper pyramid level. This way we ensure that we can deal also with large displacements.

⁶ Displacements of at most half the patch size are detectable.

3 Experiments

Our enhanced PhC approach allows us to estimate multiple motion distributions. The proposed method is evaluated at the optical flow dataset from the KITTI Vision Benchmark Suite [22] and the Middlebury Stereo Dataset [3]. Due to the fact that the PhC, by construction, aims at determining the *distribution* of motion vectors but not a dense motion field, we could not apply the metrics of these benchmarks which expect a dense motion field. Therefore we can only compare our method against the PhC implementation of OpenCV, which is based on the work of Stone et al. [16], and the ground truth data of the training datasets of the two mentioned benchmarks.

3.1 Middlebury Stereo Dataset

In this experiment, we intend to show that our proposed approach is able to estimate multiple motions within a defined patch. We also want to demonstrate that these estimates are correct and precise. However, PhC is of course only able to detect motions if the moving objects show enough structure. Therefore, we chose the dataset from 2001 as its images exhibit well structured elements.

The 6 image pairs of this stereo dataset are divided into 6 centered patches, each of size 128×128 pixels. These patches are shown as black rectangles in Fig. 2. Since this dataset was originally created for a stereo benchmark, the images are recorded by a left and right camera. Thus we can assume that the captured scene is only translated horizontally, although of course the PhC is not aware of this. The provided disparity maps express exactly this described behavior. Objects which are more far away from the camera exhibit a lower displacement than objects in the near field. The disparity values (represented by the gray level values) describe the ‘motion’ of an object between two images. For example, two different motions are present in the first patch of the disparity map 2a. The aim of this experiment is to detect exactly these multiple motions within a patch. The total quantity of available displacements of all patches of a specific image pair is listed in the second column of Table 1. Another aspect which has to be considered is that the objects do not necessarily lie in a frontal plane w.r.t. the camera and hence the translation of the object cannot be described with one single disparity value. This means that we observe disparity value *ranges*, not singular values. In our experiment, we computed all such ranges in each patch, which serve as ground truth ranges.

For each patch we executed our enhanced PhC algorithm. The results are shown in Table 1, where they are compared against the OpenCV version of the PhC. Obviously our enhanced PhC is able to estimate a significantly larger amount of motions than the OpenCV PhC. Moreover, the calculated displacements are more precise and more reliable than the OpenCV ones. We estimated every single detected motion correctly, which means that our computed displacements fall within the above stated range of ground truth data. In contrast to that, the OpenCV version does not determine all its detected motions correctly, as it can be seen in the last column of Table 1.

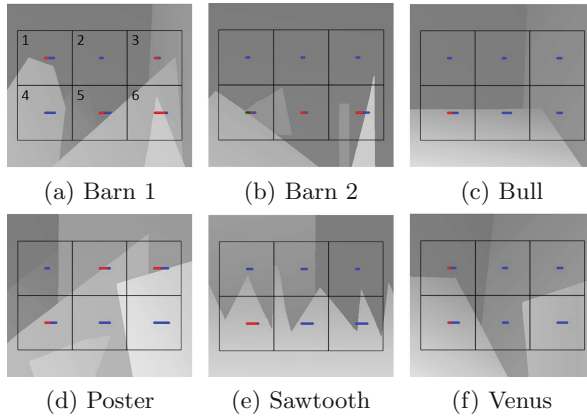


Fig. 2. All disparity maps from the Middlebury Stereo Dataset 2001 [3] divided into patches. In each patch, the estimated motion is encoded by a colored line. Note: The length of the line (respectively the motion) is scaled by a factor of two for better visualization.

Table 1. We compare the quantity of right estimated motions by our and the OpenCV PhC [16] w.r.t. the occurring motions within each dataset.

Dataset	# gt. motions	est. motions		correct est. motions	
		OpenCV	our PhC	OpenCV	our PhC
Barn 1	15	40 %	67 %	100 %	100 %
Barn 2	13	46 %	77 %	83 %	100 %
Bull	11	54 %	63 %	83 %	100 %
Poster	19	32 %	47 %	100 %	100 %
Sawtooth	13	46 %	54 %	100 %	100 %
Venus	13	46 %	62 %	100 %	100 %
Overall	84	44 %	62 %	94 %	100 %

Using the Middlebury Stereo Dataset, we showed that our enhanced PhC can detect and correctly estimate multiple motions within a patch if the individual objects possess enough structure and cover a reasonable percentage of the patch.

3.2 KITTI Optical Flow Dataset

In the second part of our experiments, we use real world driving scenes and show that our algorithm outperforms the OpenCV PhC in terms of precision and reliability and also simultaneously measures the uncertainties of the estimated motions. Furthermore we show that our four self-diagnostic checks are both useful and work correctly. For the evaluation of our new approach, we took all 194 image pairs from the KITTI Optical Flow Training Dataset [22].

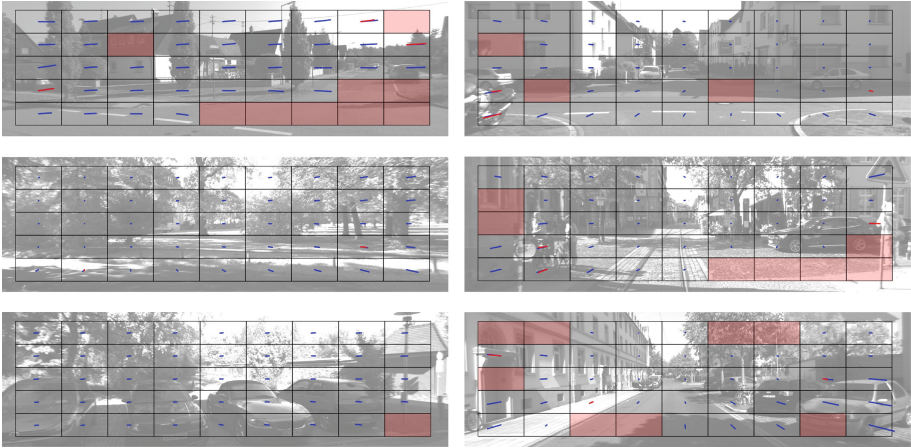


Fig. 3. 6 exemplary results taken from the KITTI Training Dataset which show the performance of our method. The colored lines indicate the length of estimate motions. A red overlay on a patch indicates that one of our checks yielded a negative result and thus no reliable and precise motion estimation is possible.

We did not evaluate our PhC method on the Test Dataset, because ground truth data is not available and the benchmark only accepts sparse or dense motion fields. We cannot provide this because we can only estimate motion distribution within a defined patch. For that reason we analyze the performance of our PhC by dividing each image into 45 non-overlapping patches of 128×64 pixels (cf. Fig. 3) and compare the results of our PhC against the ones from OpenCV and the ground truth from the training dataset. Unfortunately, KITTI does not provide ground truth for the entire image because their *LIDAR* scanner has only a limited field of view. This is why we could only do the evaluation on 6942 of the 8730 possible patches. The KITTI data provides an almost dense motion field within a patch, but we can only compare motion distributions characterized by a mean $\mathbf{m}_{n,gt}$ and a covariance matrix $\Sigma_{n,gt}$. Therefore, we determine the parameters of the motion distribution from the ground truth data for the patches $\{c_n\}_{n=1,\dots,3752}$ where K motions \mathbf{x}_i occur in a patch c_n , in the following way:

$$\hat{\mathbf{m}}_{n,gt} = \frac{1}{K} \sum_{i=1}^K \mathbf{x}_i \quad \text{and} \quad \hat{\Sigma}_{n,gt} = \frac{1}{K-1} \sum_{i=1}^K (\mathbf{x}_i - \hat{\mathbf{m}}_{n,gt})(\mathbf{x}_i - \hat{\mathbf{m}}_{n,gt})^T. \quad (15)$$

With the described self-diagnosis checks, we determined that on 3752 of the 6942 patches, the PhC can provide a reliable motion estimate. On the other patches, one of our proposed checks failed due to low structure, too much noise or too different image patches caused by large displacements. In the Fig. 4a and b, the ordered displacements in horizontal respectively vertical direction are shown for all possible patches. These plots show that our PhC is located much closer to the trend of the ground truth than the OpenCV implementation. Many of the

motions computed by OpenCV are either outliers or lie close to the zero line. As opposed to this our PhC produces only a few outliers. Consider to the given integer precision, our results comply well to the ground truth.

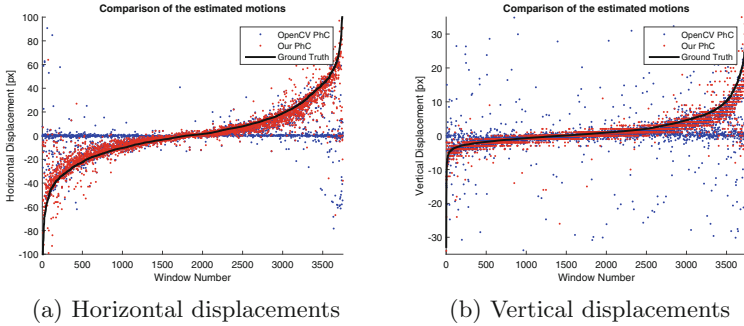


Fig. 4. Ordered horizontal and vertical displacements which occur through the 3752 patches where ground truth data and a motion estimate is available.

In the last part of this experiment, we want to show that our motion distribution parameters⁷ are well estimated. We chose a relatively pessimistic approach by evaluating $\mathbf{m}_{n,eval}$ and $\Sigma_{n,eval}$ for each patch c_n in the following way:

$$\mathbf{m}_{n,eval} = \hat{\mathbf{m}}_{n,gt} - \mathbf{m}_{n,k,PhC} \quad \text{and} \quad \Sigma_{n,eval} = \hat{\Sigma}_{n,gt} + \Sigma_{n,k,PhC}. \quad (16)$$

Figure 5a and b show the histogram of the euclidean lengths of the deviation $\{\mathbf{m}\}_{n=1,\dots,3752,eval}$ between the ground truth and our PhC and the OpenCV PhC respectively. The results show that our PhC provides fewer estimates than the OpenCV PhC, simply because we recognize patches where no reliable motion estimate is possible. Secondly, slightly more displacements are estimated correctly, as it can be seen in histogram bins $[0, 5]$. However, the main advantage of our PhC is that only very few estimates deviate more than 30 pixels to the ground truth. The OpenCV PhC, on the other hand, yields more than 2000 motion estimates which exhibit a deviation of more than 30 pixels w.r.t the ground truth. To evaluate the uncertainty of the estimate, expressed by the ‘size’ of the covariance matrix, we compute the area A_n of $\Sigma_{n,eval}$ as $A_n = \det(\Sigma_{n,eval})$, which corresponds to the 1σ -area covered by the covariance ellipse. Figure 5c shows the histogram of $\{A\}_{n=1,\dots,3752}$. Some motions have a relatively high uncertainty (large A_n), but most of the estimated motion distributions are compact (small A_n), which means that their displacements are reliable and precise.

We have evaluated our work on two different datasets. The performance of our enhanced PhC clearly outperforms the OpenCV one. We achieve good estimates of motion distributions if the moving objects possess enough structure and cover a significant part of the patch. As already stated, the purpose of our approach is

⁷ We do not assume any specific probability distribution.

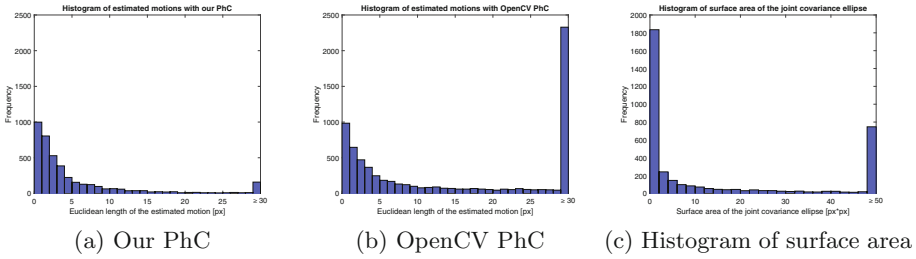


Fig. 5. In the Figs. 5a and b, the euclidean length of the deviation between the ground truth and the estimated motions of two used PhCs is shown. Figure 5c shows the 1σ -area covered by the covariance ellipse of the PhC result.

not to compute a dense optical flow field, but to estimate the dominant motions and their uncertainties. The particular advantage of our approach is that we achieve with very moderate computational effort reliable information about the distribution of the optical flow vectors within a patch - including the case of multiple motions. The runtime of our PhC is roughly 1 ms for a 256×256 pixel patch without any use of multithreading and GPU support on a common PC.

4 Summary and Conclusion

We have shown that the classical PhC method can be made significantly more robust against different sources of malfunction. This has been achieved by a systematic analysis of the effects of noise and the conditioning of the input data (texture, similarity). Obviously, the spatial precision of the method can be extended into the subpixel range by using existing schemes for providing subpixel resolution to phase correlation [11–13]. This, however, is independent from the method improvements presented here. We refrained from using any of these schemes in order to present the effects of our modifications in ‘clean room conditions’, unaffected by other modifications. We emphasize that the standard PhC is a good motion estimator for patches with a homogeneous translational motion field, whereas our extended PhC provides *distributions* for multiple motions in a patch which can be used for local methods that need a good initialization.

References

1. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: proceedings of the International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
2. Knapp, C., Carter, G.: The generalized correlation method for estimation of time delay. *IEEE Trans. Acoust. Speech Signal Process.* **24**(4), 320–327 (1976)
3. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Intern. J. C.V.* **47**(1–3), 7–42 (2002)

4. Pratt, W.K.: Correlation techniques of image registration. *IEEE Trans. Aerosp. Electron. Syst.* **AES-10**(3), 353–358 (1974)
5. Kuglin, C.D., Hines, D.C.: The Phase Correlation Image Alignment Method. In: *Proceedings of the International Conference on Cybernetics and Society*, pp. 163–165 (1975)
6. Kumar, B.V.K., Hassebrook, L.: Performance measures for correlation filters. *Appl. Opt.* **29**(20), 2997–3006 (1990)
7. Greenfeld, J.S.: An operator-based matching system. *Photogram. Eng. Remote Sens.* **8**(57), 1049–1055 (1991)
8. Tian, Q., Huhns, M.N.: Algorithms for subpixel registration. *Comput. Vis. Graph. Image Process.* **35**(2), 220–233 (1986)
9. Brown, L.G.: A survey of image registration techniques. *ACM Comput. Surv.* **24**(4), 325–376 (1992)
10. Zitová, B., Flusser, J.: Image registration methods: a survey. *Image Vis. Comput.* **21**(11), 977–1000 (2003)
11. Pearson, J.J., Hines, D.C., Golosman, S., Kuglin, C.D.: Video-rate image correlation processor. In: *Proceedings of Application of Digital Image Processing*, pp. 191–205 (1977)
12. Takita, K., Sasaki, Y., Higuchi, T., Kobayashi, K.: High-accuracy subpixel image registration based on phase-only correlation. *IEICE Trans. Fundam. Electron. Commun.* **86**(8), 1925–1934 (2003)
13. Takita, K., Muquit, M.A., Aoki, T., Higuchi, T.: A subpixel correspondence search technique for computer vision applications. *IEICE Trans. Fundam. Electron. Commun.* **E87-A**(8), 1913–1923 (2004)
14. Kirichuk, V.S., Peretjagin, G.I.: Establishing similarity between fragments and a standard. *Optoelectron. Instrum. Data Process.* **22**(4), 83–87 (1986)
15. Fleet, D.J.: Disparity from local weighted phase-correlation. In: *proceedings of the International Conference on Systems, Man and Cybernetics*, pp. 48–56 (1994)
16. Stone, H.S., Orchard, M., Chang, E.C., Martucci, S.: A fast direct Fourier-based algorithm for subpixel registration of images. *IEEE Trans. Geosci. Remote Sens.* **39**(10), 2235–2243 (2001)
17. Guizar-Sicairos, M., Thurman, S.T., Fienup, J.R.: Efficient subpixel image registration algorithms. *Opt. Lett.* **33**(2), 156–158 (2008)
18. Jung, I.K., Lacroix, S.: High resolution terrain mapping using low altitude aerial stereo imagery. In: *proceedings of the International Conference on Computer Vision*, pp. 946–951 (2003)
19. Abdelfattah, R., Nicolas, J.M., Tupin, F.: Interferometric SAR image coregistration based on the Fourier Mellin invariant descriptor. *IEEE Int. Geosci. Remote Sens. Symp.* **3**, 1334–1336 (2002)
20. Eisenbach, J., Mertz, M., Conrad, C., Mester, R.: Reducing camera vibrations and photometric changes in surveillance video. In: *proceedings of the International Conference on Advanced Video and Signal Based Surveillance*, pp. 69–74 (2013)
21. Foroosh, H., Zerubia, J.B., Berthod, M.: Extension of phase correlation to subpixel registration. *IEEE Trans. Image Process.* **11**(3), 188–200 (2002)
22. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: *proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361 (2012)
23. Morgan, G.L.K., Jian, G.L., Yan, H.: Precise subpixel disparity measurement from very narrow baseline stereo. *IEEE Trans. Geosci. Remote Sens.* **48**(9), 3424–3433 (2010)