High-Speed Analysis Of Speckle-Based Imaging Data With Unified Modulated Pattern Analysis (UMPA)

Fabio De Marco[1,2,a)], Sara Savatović[1,2], Mirko Riedel[3,4], Ronan Smith[5], Vittorio Di Trapani[1,2], Marco Margini[1,2], Ginevra Lautizi[1,2], Julia Herzen[3] and Pierre Thibault[1,2]

[1]*Department of Physics, University of Trieste, Via A. Valerio 2, 34127 Trieste, Italy*
[2]*Elettra Sincrotrone S.C.p.A., Strada Statale 14 - km 163.5, 34149 Basovizza, Italy*
[3]*Physics Department, Technical University of Munich, James-Franck-Str. 1, 85748 Garching, Germany*
[4]*Institute for Materials Physics, Helmholz-Zentrum Hereon, 21502 Geesthacht, Germany*
[5]*Department of Physics, University of Southampton, University Road, Southampton SO17 1BJ, United Kingdom*

[a)]Corresponding author: fabiodomenico.demarco@units.it

Abstract. When a partially coherent X-ray source illuminates an object with an irregular surface, a near-field speckle pattern may appear at some distance downstream. Speckle-based X-ray, a relatively novel imaging technique, exploits this effect to extract information about attenuation, refraction, and small-angle scatter induced by a sample. Over the last ten years, different acquisition and image processing techniques have been developed to extract this information from the image data. One of these techniques, Unified Modulated Pattern Analysis (UMPA), uses a speckle-tracking approach, implemented by the least-squares minimization of a cost function that simultaneously models all three image modalities. We here present a new implementation of UMPA. By shifting from Python to C++ and Cython, execution speed was increased by a factor of about 125. Furthermore, a new acquisition modality, "sample-stepping", was introduced. Finally, we discuss the origin and mitigation of two types of image artifacts that may arise during image processing with UMPA.

INTRODUCTION

Over the last three decades, a wide variety of X-ray phase-contrast imaging methods has been developed. Propagation-based imaging (PBI) employs no optical elements, but extracts information about the sample from propagation fringes which occur due to self-interference of the beam [1, 2]. Alternatively, rocking curves can be acquired using analyzer crystals, which allows isolating rays with a specific angle of refraction, yielding "differential-phase" data [3, 4]. In grating-based imaging (GBI), a third approach, attenuating and/or phase-shifting transmission gratings can be used to produce grating self-images, whose deflection can be detected very sensitively by arranging two or three gratings into a Talbot- or Talbot-Lau interferometric arrangement [5, 6]. Closely related to this is the edge-illumination technique, which instead uses masks of larger pitch that do not exploit interference effects [7, 8]. Finally, a range of "wavefront marking" techniques have been developed which introduce an (ordered or unordered) optical element into the beam and directly measure the resulting intensity modulation on a detector. The properties of an imaged specimen are then determined by analyzing its impact on the modulation pattern (i.e., attenuation, distortion, or blurring).

Among the more recent developments of wavefront marking techniques is "speckle-based X-ray imaging" (SBI), where the intensity modulation is achieved by an object with an irregular microstructure (such as layers of sandpaper, or biological membranes). The random microstructure acts as an unordered array of refracting lenses and produces a similarly unordered intensity pattern on the detector (near-field X-ray speckle) [9].

Numerous measurement techniques and signal extraction approaches have been developed for SBI. X-ray speckle tracking (XST) [10, 11], the oldest approach, was adapted from digital image correlation (DIC), a widespread optical-light technique used to measure strain and deformation in planar objects [12]. An image is acquired with and without the object in the beam. A pixel neighborhood or "window" is defined for every pixel in one of the images, and a matching window is sought in the other image. Information about refraction is encoded in the distance between the locations of the two windows, attenuation is determined from the decrease of mean intensity, and small-angle scattering is given by the decrease of the mean-normalized speckle amplitudes within the window (upon introducing the object).

Further refinements to this approach were later introduced, such as "X-ray Speckle Scanning" (XSS) [13], where the analysis window is replaced by a displacement of the diffuser in small steps, thus yielding better resolution and sensitivity, at the cost of slower measurements. Later, "X-ray Speckle Vector Tracking" (XSVT) combined arbitrary diffuser displacements [14] trajectories with the optional use of analysis windows [15].

The "Unified Modulated Pattern Analysis" (UMPA) approach, although developed independently from XSVT, uses a relatively similar approach: images are acquired with and without the sample, optionally at different diffuser positions. In contrast to XSVT, which uses zero-normalized cross-correlation (ZNCC) as the metric of similarity between the sample and reference images, UMPA uses a cost function written as a sum of squared differences (SSD), which also directly incorporates attenuation and dark-field (see the section below) [16].

Besides this speckle-tracking approach, more recent publications have introduced alternative techniques for extracting information from speckle images, such as the "Multimodal Intrinsic Speckle Tracking" (MIST) approach, which solves the transport-of-intensity equation [17] or the Fokker-Planck equation [18], and does not require the use of analysis windows. Additionally, "Ptychographic X-ray Speckle Tracking" (PXST) has been recently introduced for use in setups with very strong geometric magnification. This approach does not necessitate a diffuser, and instead requires transverse movement of the sample between frames. Image information is retrieved in an iterative process reminiscent of ptychography [19].

Nearly all signal extraction methods for SBI are quite computationally intensive, e.g., when compared to PBI or GBI. This is especially problematic since the resolution and frame rate of X-ray detectors (and thus, the rate of generated data) is steadily increasing. In order to maintain UMPA as a practical and convenient solution for SBI imaging tasks, especially for large computed tomography acquisitions, we have re-implemented the model using C++ and Cython, which yielded significant improvements in execution speed.

In the following section, we introduce the mathematics behind the UMPA model, give an overview of the re-implementation of the model in Cython and C++, quantify the achieved speedup, and introduce a new acquisition technique called "sample-stepping". In the section "Artifacts and their mitigation", we introduce two types of artifacts that may arise while processing image data with UMPA, and we present techniques to reduce their impact. We finish with a brief conclusion.

STRUCTURE OF UNIFIED MODULATED PATTERN ANALYSIS

Cost function definition

UMPA determines image modalities by comparing "sample images" and "reference images", i.e., speckle images with and without a sample in the beam. This has also been called the "differential mode" of speckle imaging [10]. It can be applied to a set of $M \geq 1$ sample and reference images with different relative lateral displacements between the speckle pattern and the sample. This relative displacement is usually achieved by lateral displacement of the diffuser by a motorized stage, but this can also be achieved by translating the sample. Our new implementation allows directly processing datasets acquired in this manner (so-called "sample-stepping", see below).

This matching of sample and reference images is achieved by defining an analysis window, centered on a pixel of interest in the sample images I_m, and comparing these to an analysis window centered on another pixel of interest in the reference images $I_{0,m}$ ($m = 1, \ldots, M$). The similarity between the windows is calculated as a sum of squared differences (SSD) of gray values in the two windows. In order to also include attenuation and small-angle scattering, a physical model is first applied to the reference image values before calculating the SSD. In short, the cost function

$$L(\mathbf{r}; \mathbf{u}, T, D) = \sum_{m=1}^{M} \sum_{w_x, w_y=-N}^{N} \Gamma(\mathbf{w}) \left[I_m^{(\text{model})}(\mathbf{r} + \mathbf{w} - \mathbf{S}_m; \mathbf{u}, T, D) - I_m(\mathbf{r} + \mathbf{w} - \mathbf{S}_m) \right]^2 . \tag{1}$$

is minimized, where $\mathbf{r} = (r_x, r_y)$ is the pixel of interest, $\mathbf{u} = (u_x, u_y)$ is the shift between the two analysis windows, and $\mathbf{w} = (w_x, w_y)$ is a summation variable over the analysis window. The parameter N characterizes the size of the analysis window [which is $(2N+1)\times(2N+1)$ pixels]. $\Gamma(\mathbf{w})$ is a 2D Hamming window which decreases to zero towards the edges of the analysis window and thus produces a smoother appearance of the output images. The quantity $\mathbf{S}_m$ represents the lateral translation of the sample if the "sample-stepping" mode is used (see below), and is otherwise $(0, 0)$ for all m. Finally, $I_m^{(\mathrm{model})}(\mathbf{r}; \mathbf{u}, T, D)$ represents the application of the physical model for attenuation and small-angle scattering, applied to the reference images $I_{0,m}$:

$$I_m^{(\mathrm{model})}(\mathbf{r}; \mathbf{u}, T, D) = T \left\{ D \left[I_{0,m}(\mathbf{r} - \mathbf{u}) - \langle I_{0,m} \rangle (\mathbf{r} - \mathbf{u}) \right] + \langle I_{0,m} \rangle (\mathbf{r} - \mathbf{u}) \right\}, \tag{2}$$

$$\langle I_{0,m} \rangle (\mathbf{r}) = \frac{\sum_{w_x, w_y = -N}^{N} \Gamma(\mathbf{w}) I_{0,m}(\mathbf{r} + \mathbf{w})}{\sum_{w_x, w_y = -N}^{N} \Gamma(\mathbf{w})}. \tag{3}$$

D is a scaling factor for the amplitude of the speckle pattern, whereas T is a scaling factor for the overall intensity of the gray values. $\Gamma(\mathbf{w})$ is also used as a weighting factor for the local average of the reference, $\langle I_{0,m} \rangle$.

Minimization

Image information is extracted by minimizing $L(\mathbf{r}; \mathbf{u}, T, D)$ from Eq. (1) for every value of $\mathbf{r}$:

$$\hat{\mathbf{u}}(\mathbf{r}), \hat{T}(\mathbf{r}), \hat{D}(\mathbf{r}) = \underset{\mathbf{u}, T, D}{\operatorname{argmin}} \, L(\mathbf{r}; \mathbf{u}, T, D). \tag{4}$$

Note, however, that $\mathbf{u}$ is only well-defined for integer multiples of the pixel size, whereas T and D can assume any real value (and make physical sense for the interval $[0, 1]$). In order to simplify the cost function space over which to minimize, a partial minimization of L with respect to T and D is performed for every new evaluated value of $\mathbf{u}$. This is done by setting $\partial L / \partial T = \partial L / \partial D = 0$, and solving for T and D (which can be done analytically). This thus yields the partially minimized cost function $\hat{L}$, as well as the optimized transmittance and dark-field values $\tilde{T}$, $\tilde{D}$ (which still depend on $\mathbf{u}$):

$$\hat{L}(\mathbf{r}; \mathbf{u}) = \min_{T, D} L(\mathbf{r}; \mathbf{u}, T, D), \tag{5}$$

$$\tilde{T}(\mathbf{r}; \mathbf{u}), \tilde{D}(\mathbf{r}; \mathbf{u}) = \underset{T, D}{\operatorname{argmin}} \, L(\mathbf{r}; \mathbf{u}, T, D). \tag{6}$$

The remaining minimization thus takes place entirely in $\mathbf{u}$-space. For each $\mathbf{r}$, the first step of this minimization is determining a local minimum of $\hat{L}(\mathbf{r}; \mathbf{u})$. This is achieved by one-dimensional optimization performed alternately in the u_x- and u_y-directions, yielding the discrete minimum $\hat{\mathbf{u}}_{\mathrm{d}}(\mathbf{r})$, as well as the transmittance and dark-field values $\hat{T}(\mathbf{r})$, $\hat{D}(\mathbf{r})$:

$$\hat{\mathbf{u}}_{\mathrm{d}}(\mathbf{r}) = \underset{\mathbf{u}}{\operatorname{argmin}} \, \hat{L}(\mathbf{r}; \mathbf{u}), \;\; \hat{T}(\mathbf{r}) = \tilde{T}(\mathbf{r}; \hat{\mathbf{u}}_{\mathrm{d}}(\mathbf{r})), \;\; \hat{D}(\mathbf{r}) = \tilde{D}(\mathbf{r}; \hat{\mathbf{u}}_{\mathrm{d}}(\mathbf{r})). \tag{7}$$

Since the amount of shift $\hat{\mathbf{u}}$ is rarely more than one pixel, $\hat{\mathbf{u}}_{\mathrm{d}}(\mathbf{r})$ is not of sufficient precision. We therefore "interpolate" the cost function landscape to a continuous range of $\mathbf{u}$ values. To this aim, we have determined a convolution kernel which yields the equivalent result of first bilinearly interpolating the raw data, and then performing the cost function calculation on this data. We find that the kernel to use for this purpose is $B \star B$, i.e. the autocorrelation function of the kernel for bilinear interpolation.

$$\hat{L}_{\mathrm{cont}}(\mathbf{r}; \mathbf{u}) = \hat{L}(\mathbf{r}; \mathbf{u}') \otimes (B \star B)(\mathbf{u}') \text{ [for } \mathbf{u} \text{ values in the vicinity of } \hat{\mathbf{u}}_{\mathrm{d}}(\mathbf{r})\text{]}, \tag{8}$$

$$\hat{\mathbf{u}}(\mathbf{r}) = \underset{\mathbf{u}}{\operatorname{argmin}} \, \hat{L}_{\mathrm{cont}}(\mathbf{r}; \mathbf{u}), \tag{9}$$

where $\otimes$ denotes convolution, and $\star$ denotes cross-correlation. Minimization according to Eq. (9) in the resulting cost function landscape is then performed using the Newton-Raphson algorithm.

IMPROVEMENTS BY THE NEW IMPLEMENTATION

Speedup due to re-implementation in Cython and C++

The new implementation of UMPA uses C++ libraries for the computationally intensive parts of the calculation, namely the calculation of the cost function, as well as its minimization. The main program structure (program flow, memory management, etc.) is written in Cython, which serves as an interface between the C++ libraries and a Python interface. Like its predecessor, the new version of UMPA can thus be used like an ordinary Python module.

In a simple benchmark (measuring processing time for a simulated speckle dataset of 20 sample and reference images of 1000 × 1000 pixels), we measured a single-thread performance increase of about 125 compared to the previous implementation (old version: 3864 s, new version: 30.9 s). This speedup factor is further increased by the use of multithreading, although the previous implementation can also be parallelized. With this speedup, applying UMPA to large tomographic datasets with thousands of projections becomes feasible with limited processing time. This new implementation has previously been used in [20] for signal extraction from both speckle data and Talbot images of two-dimensional gratings (Talbot array illuminators).

Sample-stepping

Combining image data from multiple diffuser positions has the primary benefit of eliminating spurious minima from the cost function landscape, thus reducing the frequency of failed minimization procedures and significantly increasing image quality. Usually, this is achieved by displacing the diffuser in the plane orthogonal to the beam, but it works equally well by displacing the sample instead (which is what we call "sample-stepping"). This latter approach has the added benefit of increasing the effective field of view (if the movement steps of the sample are sufficiently large, see Figure 1). In principle, a "sample-stepping" dataset can always be turned into a "diffuser-stepping" dataset by shifting the individual image frames to undo the translation of the sample. However, it must be considered that, unlike in a real "diffuser-stepping" dataset, the number of image frames contributing to the cost function (the "coverage") varies with the position in the field of view, which must be taken into account for image reconstruction. The new implementation of UMPA calculates this coverage from the lateral displacement of the sample in each frame of the dataset (in pixel multiples) and calculates the resulting images without requiring further manual input.

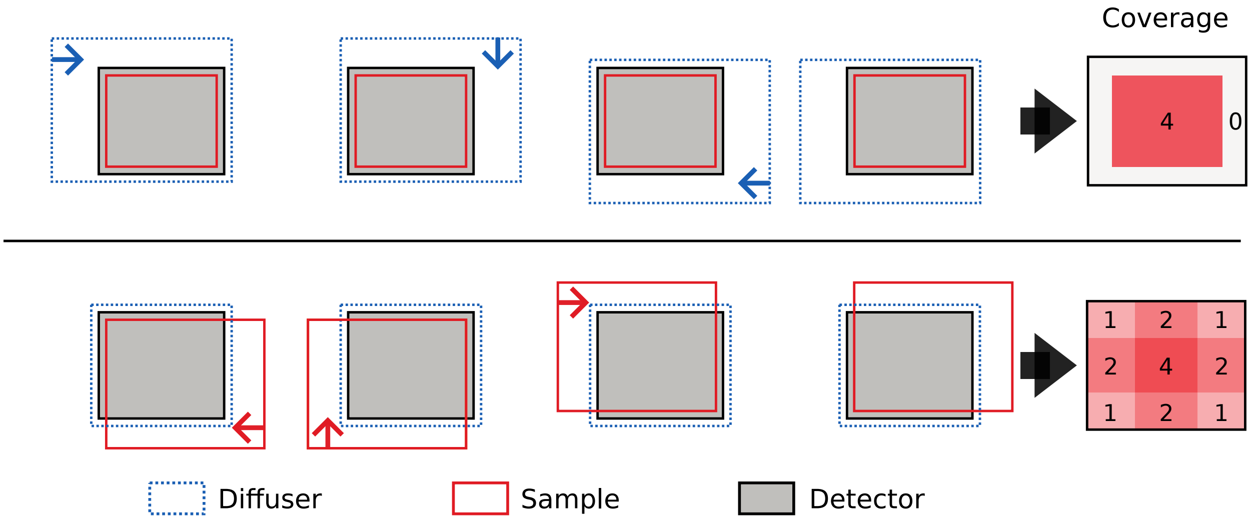

FIGURE 1. Comparison of multi-frame acquisition methods. Top row: In the conventional "diffuser-stepping" method, the sample remains stationary and the diffuser is displaced, yielding a constant coverage across the field of view of the detector. Bottom row: In the "sample-stepping" approach, the diffuser remains stationary, and the sample is displaced. This yields a larger measured field of view, albeit at the expense of a decrease of coverage towards the edges.

To demonstrate the practical use of this technique, we measured a small flower at the P05 beamline at PETRA III, DESY (Hamburg, Germany) (Figure 2). The "sample-stepping" technique is especially useful when the field of view must be increased beyond the size of the detector, since it requires much fewer motor displacements than an equivalent "tiled" acquisition with the "diffuser-stepping" method. Furthermore, this technique is compatible with a continuous displacement of the sample, which would completely eliminate the measurement overhead due to motor movements.

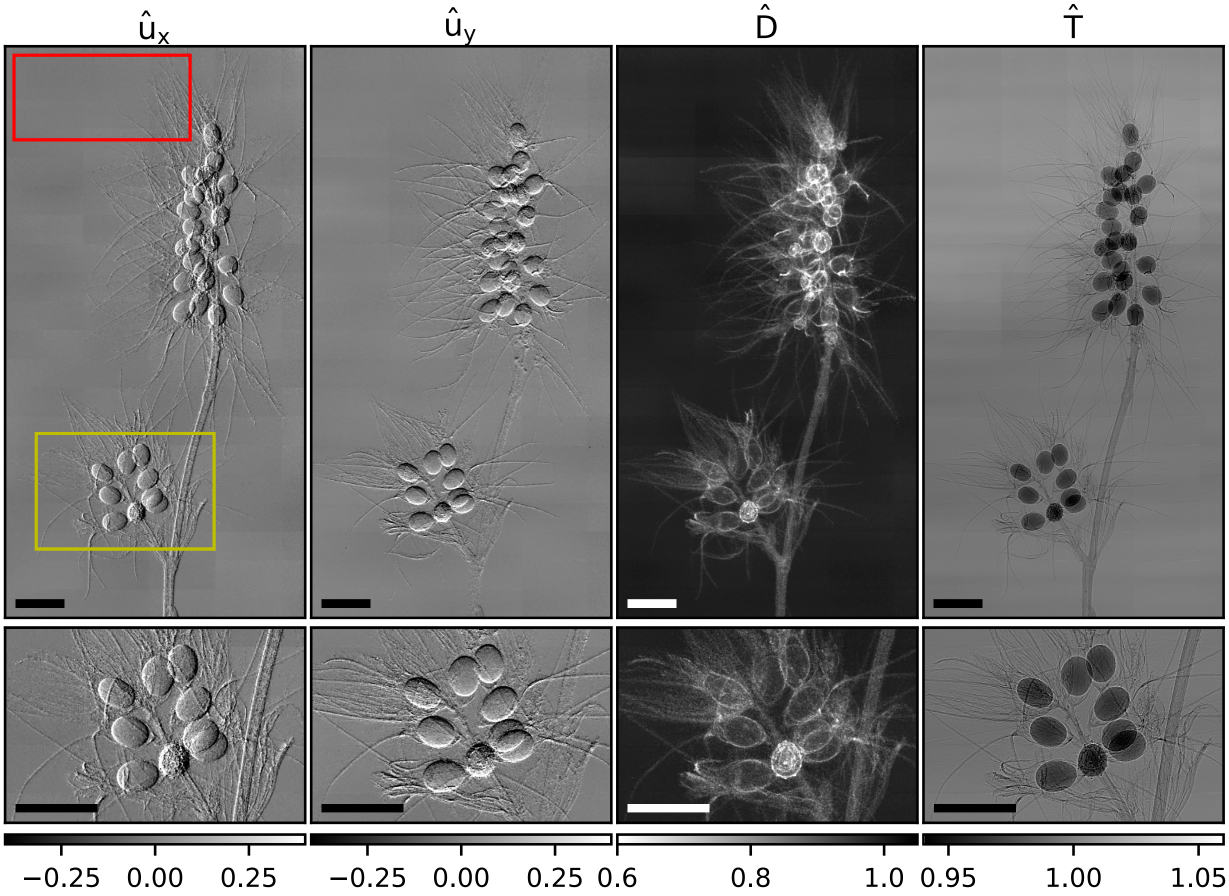

FIGURE 2. Small flower imaged with the "sample-stepping" technique at the P05 beamline at PETRA III, DESY, and processed by the new implementation of UMPA. From the left: horizontal differential phase $\hat{u}_x$, vertical differential phase $\hat{u}_y$, dark-field $\hat{D}$ and transmittance $\hat{T}$. The effective size of the detector field of view (3.48 mm × 7.25 mm) is shown by the red rectangle. The second row of images shows zoomed-in sections of the images (indicated by the yellow rectangle). The sample was displaced on a 2-dimensional grid of 25 × 10 positions. 9 layers of 400-grit sandpaper were used as a diffuser. Photon energy: 35 keV, pixel size: 0.916 μm. Sample-detector distance was 20 cm, and the diffuser was 11.5 cm upstream of the sample. The full size of the processed images is 13500 × 25700 pixels. Scalebars are 2 mm.

ARTIFACTS AND THEIR MITIGATION

Estimation bias in differential phase

We have identified the presence of a bias in the calculation of the differential phase signal $\mathbf{\hat{u}}$. This effect is visible as a speckle-like background pattern even in sample-free areas of the field of view. We believe that this bias occurs due to a slight inaccuracy in the derivation used for the convolution kernel $B \star B$. The calculation assumes that the cost function can be written as a sum of cross-correlation terms involving I_m, $I_{0,m}$, and $\langle I_{0,m} \rangle$. This is accurate for the original cost function L, but during the calculation of the partially minimized cost function $\hat{L}$, a non-linear step is introduced to the calculation, invalidating this assumption, and probably leading to the observed behavior.

However, we have determined a simple method to find an estimate for this bias, namely: using UMPA to compare the reference images to themselves. In other words, the input variable $I_m(\mathbf{r})$ meant for the sample image stack is simply replaced by the reference image stack $I_{0,m}(\mathbf{r})$. The differential-phase images resulting from this are a good approximation to the differential-phase bias, and can be simply subtracted from the normally retrieved differential phase images of the sample.

Ambiguity in the attribution of UMPA output to pixel coordinates

In the presence of refraction, the UMPA model compares reference and analysis windows centered on different pixels. This is apparent from Eq. (1) and (2): the cost function $L(\mathbf{r};\mathbf{u},T,D)$ is calculated from the reference window centered on $\mathbf{r}-\mathbf{u}$, while the sample window is centered on $\mathbf{r}$. This prompts the question of which pixel the found signal values $\hat{\mathbf{u}}$, $\hat{T}$, and $\hat{D}$ should be assigned to. Assigning found values to the center of the reference window $(\mathbf{r}-\mathbf{u})$ is the physically accurate solution, since this location coincides with the interaction point of beam and sample, disregarding shifts due to refraction occurring after the interaction. However, the previously published implementation of UMPA instead assigns the values to the center of the sample window.

The new implementation supports both assignment methods, but we find that the "old" approach leads to a better image quality near sharp edges or strong propagation fringes. The reason for this is that the cost function landscapes (as a function of $\mathbf{u}$) that result from the two approaches differ: If values are assigned to the reference window, the location of the matched sample window varies with $\mathbf{u}$ during the optimization. In regions of sample images with strong edges or propagation fringes, UMPA yields high cost function values (since these effects are not well described by the UMPA model). The optimization thus consistently avoids matching with these regions of the sample image(s), leading to a black-and-white artifact in the differential-phase maps.

If output values are instead assigned to the center of the sample window, UMPA can not avoid matching certain regions of the sample images. The presence of edges or fringes in a sample image region leads to an increase in cost function values with *all* reference window positions, and thus mostly acts as an added offset of the cost function landscape, and convergence is only weakly affected. In most cases, the misattribution of positions due to this physically incorrect assignment is negligible, since the magnitude of $\hat{\mathbf{u}}(\mathbf{r})$ is far below one pixel. In cases of very strong refraction, however, a correction method is imaginable that applies an image warping transform to the images produced by UMPA, the magnitude of which is determined by $\hat{\mathbf{u}}(\mathbf{r})$.

CONCLUSION

We have introduced the mathematical basis of the Unified Modulated Pattern Analysis (UMPA) approach for processing speckle-based X-ray imaging data, and presented an improved implementation of UMPA. This new version features a much higher computational efficiency, and also includes a number of new features, most prominently the ability to perform "sample-stepping" measurements. Furthermore, we have discussed two important artifacts which may arise during UMPA processing and presented approaches to minimize their impact. The program code for this new implementation will be made publicly available on GitHub in the context of an upcoming publication [21]. This implementation forms the basis for ongoing work, such as its extension to directional small-angle scattering, which is the focus of an additional upcoming publication [22].

ACKNOWLEDGMENTS

This publication is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant agreement No. 866026).

REFERENCES

[1] A. Snigirev, I. Snigireva, V. Kohn, S. Kuznetsov, and I. Schelokov, Rev. Sci. Instrum. **66**, 5486–5492 (1995).

[2] D. Paganin, S. C. Mayo, T. E. Gureyev, P. R. Miller, and S. W. Wilkins, J. Microsc. **206**, 33–40 (2002).

[3] D. Chapman, W. Thomlinson, R. E. Johnston, D. Washburn, E. Pisano, N. Gmür, Z. Zhong, R. Menk, F. Arfelli, and D. Sayers, Phys. Med. Biol. **42**, 2015–2025 (1997).

[4] O. Oltulu, Z. Zhong, M. Hasnah, M. N. Wernick, and D. Chapman, J. Phys. D: Appl. Phys. **36**, 2152–2156 (2003).

[5] A. Momose, S. Kawamoto, I. Koyama, Y. Hamaishi, K. Takai, and Y. Suzuki, Jpn. J. Appl. Phys. **42**, L866–L868 (2003).

[6] F. Pfeiffer, T. Weitkamp, O. Bunk, and C. David, Nat. Phys. **2**, 258–261 (2006).

[7] A. Olivo, F. Arfelli, G. Cantatore, R. Longo, R. H. Menk, S. Pani, M. Prest, P. Poropat, L. Rigon, G. Tromba, E. Vallazza, and E. Castelli, Med. Phys. **28**, 1610–1619 (2001).

[8] P. R. Munro, K. Ignatyev, R. D. Speller, and A. Olivo, Proc. Natl. Acad. Sci. U.S.A. **109**, 13922–13927 (2012).

[9] R. Cerbino, L. Peverini, M. A. C. Potenza, A. Robert, P. Bösecke, and M. Giglio, Nat. Phys. **4**, 238–243 (2008).

[10] S. Bérujon, E. Ziegler, R. Cerbino, and L. Peverini, Phys. Rev. Lett. **108**, 158102 (2012).

[11] K. S. Morgan, D. M. Paganin, and K. K. W. Siu, Appl. Phys. Lett. **100**, 124102 (2012).

[12] B. Pan, K. Qian, H. Xie, and A. Asundi, Meas. Sci. Technol. **20**, 062001 (2009).

[13] S. Berujon, H. Wang, and K. Sawhney, Phys. Rev. A **86**, 063813 (2012).

[14] S. Berujon and E. Ziegler, Phys. Rev. A **92**, 013837 (2015).

[15] S. Berujon and E. Ziegler, Phys. Rev. Appl. **5**, 044014 (2016).

[16] M.-C. Zdora, P. Thibault, T. Zhou, F. J. Koch, J. Romell, S. Sala, A. Last, C. Rau, and I. Zanette, Phys. Rev. Lett. **118**, 203903 (2017).

[17] D. M. Paganin, H. Labriet, E. Brun, and S. Berujon, Phys. Rev. A **98**, 053813 (2018).

[18] K. M. Pavlov, D. M. Paganin, H. T. Li, S. Berujon, H. Rougé-Labriet, and E. Brun, J. Opt. **22**, 125604 (2020).

[19] A. J. Morgan, H. M. Quiney, S. Bajt, and H. N. Chapman, J. Appl. Crystallogr. **53**, 760–780 (2020).

[20] A. Gustschin, M. Riedel, K. Taphorn, C. Petrich, W. Gottwald, W. Noichl, M. Busse, S. E. Francis, F. Beckmann, J. U. Hammel, J. Moosmann, P. Thibault, and J. Herzen, Optica **8**, 1588–1595 (2021).

[21] F. De Marco, S. Savatović, R. Smith, V. Di Trapani, M. Margini, G. Lautizi, and P. Thibault, “High-speed processing of speckle-based imaging data with Unified Modulated Pattern Analysis (UMPA)” (submitted).

[22] R. Smith, F. De Marco, L. Broche, M.-C. Zdora, N. W. Phillips, R. Boardman, and P. Thibault, “X-ray directional dark-field imaging using Unified Modulated Pattern Analysis” (submitted).