



Iterative Phase Correlation Algorithm for High-precision Subpixel Image Registration

Zdenek Hrazdír¹ , Miloslav Druckmüller¹, and Shadia Habbal²

¹ Institute of Mathematics, Faculty of Mechanical Engineering, Brno University of Technology, 616 69 Brno, Czech Republic; hrazdira.zdenek@google.com

² Institute for Astronomy, University of Hawaii, Honolulu, HI 96822, USA

Received 2019 November 7; revised 2019 December 14; accepted 2019 December 17; published 2020 February 20

Abstract

Many astrophysical observations and measurement techniques that rely on data from images include an image registration step. The results of such techniques thus heavily rely on the precision of the registration. We present an Iterative Phase Correlation (IPC) algorithm, which is an extension of the well-known phase correlation method of image registration and is ideally suited for problems, where the subpixel registration accuracy plays a crucial role. Furthermore, a sophisticated and reliable method of optimal IPC parameter estimation is described. The paper includes examples of such optimized parameters for *Solar Dynamics Observatory* (SDO)/Helioseismic and Magnetic Imager, SDO/Atmospheric Imaging Assembly, and *Solar Terrestrial Relations Observatory A/B* Sun Earth Connection Coronal and Heliospheric Investigation images. The new method (both with or without the parameter optimization step) significantly outperforms standard image registration methods, such as (non-iterative) phase correlation or (normalized) cross correlation in the sense of subpixel accuracy. A step-by-step pseudocode implementation is also included.

Unified Astronomy Thesaurus concepts: [Astronomical techniques](#) (1684); [Direct imaging](#) (387); [Astronomy software](#) (1855); [CCD observation](#) (207); [Publicly available software](#) (1864); [Fast Fourier transform](#) (1958); [Astrometry](#) (80); [Algorithms](#) (1883); [Astronomical methods](#) (1043); [Astronomy data analysis](#) (1858); [Computational methods](#) (1965); [Analytical mathematics](#) (38)

1. Introduction

The need to effectively solve the image registration problem arises in many domains (Foroosh et al. 2002), ranging from medical (Jenkinson & Smith 2001) and satellite imaging (Do et al. 2019; Li et al. 2019; Mahmood & Lee 2019) to optical flow (Lefebure & Cohen 2001), experimental mechanics (Bing et al. 2006), 3D reconstruction (Gravel et al. 2012; Li et al. 2018; Casella et al. 2019), and astrophysics (Shapiro et al. 2013; Chen et al. 2014; Zhou & Yu 2018). Algorithms based on a notion of correlation are one of the most frequently used and widely known techniques in this domain (Leng et al. 2019).

The problem of image registration consists of finding the exact image geometric alignment, which in turn consists of determining the scale, rotation, and shift between two input images. If the *log-polar transform* is applied to the amplitude spectra of both input images, rotation and scale can be inferred from the shift between them (Reddy & Chatterji 1996; Druckmüller 2009). Thus one needs to be concerned only with determining the image shift, without any loss of generality.

In the standard cross correlation approach, two real-valued functions $f(t)$, $g(t)$ and their cross correlation (\times) and convolution ($*$), defined as

$$(f \times g)(\tau) = \int_{-\infty}^{\infty} \overline{f(t)} g(t + \tau) dt, \quad (1)$$

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau, \quad (2)$$

are considered. Because of the ease of computing convolutions via the convolution theorem and Fourier transforms, it is desirable to express the cross correlation in the form of convolution. From these definitions we can readily see that

$$f(t) \times g(t) = \overline{f(-t)} * g(t), \quad (3)$$

which is the sought cross correlation/convolution relation. One of the basic properties of the Fourier transform (here denoted \mathcal{F}), namely, the conjugate property

$$\mathcal{F}\{\overline{f(t)}\} = \overline{\mathcal{F}\{f(-t)\}} \quad (4)$$

allows the computation of cross correlation in the frequency domain as

$$\mathcal{F}\{f \times g\} = \mathcal{F}\{f\} \cdot \overline{\mathcal{F}\{g\}}. \quad (5)$$

Applying the inverse Fourier transform (denoted \mathcal{F}^{-1}), we get

$$f \times g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \overline{\mathcal{F}\{g\}}\}. \quad (6)$$

If we consider functions f and g as two-dimensional discrete-valued functions over a finite region (e.g., images) and replace the continuous Fourier transforms with their appropriate discrete two-dimensional counterparts, the coordinates of the cross correlation maximum will result in the corresponding pixel shift between input images:

$$[\Delta x, \Delta y] = \arg \max_{x,y} (\mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \overline{\mathcal{F}\{g\}}\}). \quad (7)$$

The phase correlation approach (Reddy & Chatterji 1996) differs from cross correlation only by having the cross-power spectrum normalized element-wise, thus removing the magnitude component and correlating phases only (hence the name *phase correlation*). Theoretically, this approach results in a Dirac impulse δ ($\Delta x, \Delta y$) at the location of the image shift. Since input images are finite and noisy, this will not be the case in practice. One can then simply estimate the image shift as the phase correlation maximum location, similarly to the cross correlation approach. The phase correlation image shift

can then be computed as

$$[\Delta x, \Delta y] = \arg \max_{x,y} \left(\mathcal{F}^{-1} \left\{ \frac{\mathcal{F}\{f\} \cdot \overline{\mathcal{F}\{g\}}}{|\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}|} \right\} \right). \quad (8)$$

The Iterative Phase Correlation (IPC) algorithm presented here is an extension of the standard phase correlation method.

2. The IPC Algorithm

In the IPC algorithm, the improved subpixel accuracy is a result of the iterative character of the method, together with bilinear upscaling. It is important to mention that the exact same extension can be applied to the cross correlation method or any of its variants. However, the standard phase correlation technique frequently outperforms cross correlation (in the sense of subpixel accuracy—[Feroosh et al. 2002](#)), and thus is a better candidate for further accuracy refinement. The required steps are given in what follows.

2.1. Image Alignment

In the field of astrophotography and microscopy, for example, it is often necessary to compose images taken with different instruments and varying wavelengths. These images can be noticeably misaligned due to differences in location of the instruments, varying times of acquisition, and/or varying optics. Misalignments typically include a combination of image translation in both directions, image rotation, and scaling. To accurately align two such images, four parameters need to be precisely estimated—pixel shifts, Δx and Δy , in the x , y directions, rotation angle φ , and a scale factor s . If these four parameters are computed with good subpixel accuracy, the resulting composed image can have distinctly better characteristics than an image with standard pixel accuracy methods. This subpixel composition is possible via the use of bilinear (or any other) interpolation.

2.2. IPC Image Alignment Process

The IPC image alignment procedure first evaluates the rotation and scale parameters φ , s by computing the shift between log-polar transformed normalized logarithms of Fourier transform magnitude components of both input images with the IPC algorithm. The scale ratio of Fourier transform magnitude components is inversely proportional to the image scale ratio, according to the Fourier transform scale property

$$\mathcal{F}\{f(ax)\}(\xi) = \frac{1}{|a|} \mathcal{F}\{f(x)\}\left(\frac{\xi}{a}\right). \quad (9)$$

This expression can be intuitively interpreted as the fact that to increase the size of a feature in an image, the basic corresponding spatial frequencies need to be decreased and vice versa. One of the input images is then rotated and scaled back by the computed amount, followed by another run of IPC, this time to estimate the subpixel shift Δx and Δy , which is again applied to the corresponding image, therefore completing the alignment. All mentioned geometric transformations are done subpixel-wise via bilinear interpolation.

2.3. Filtering

Let $C(x, y)$ be the normalized cross-power spectrum of images $I_1(x, y)$ and $I_2(x, y)$, which are optionally spatially windowed, as windowing can help reduce spectral leakage present in the discrete Fourier transform ([Harris 1978](#)). The characteristics of the resulting phase correlation peak and its neighborhood can be improved by first applying frequency domain filtering to the cross-power spectrum. Higher frequencies of $C(x, y)$ mostly contain noise, and for this reason correlating their phases will most likely lower the precision of the result. Therefore, ideally a low-pass filter is first applied to $C(x, y)$. Likewise, the lowest frequencies mostly contain the overall brightness of the image. Hence to make the method more resilient to different levels of brightness between the two input images and to allow for more parameter optimization, these lower frequencies can also be dampened. However, it is important to realize that the phase correlation algorithm is correlating phases only, contrary to cross correlation, and thus has some level of brightness resilience built in. The resulting ideal frequency domain filter for the proposed method is then a bandpass filter, whose optimal parameters depend on the individual image characteristics (levels of noise, image size, difference in exposure times, sharpness, etc.). The method presented here performed well with a *Gaussian bandpass filter* constructed by multiplying a *Gaussian low-pass filter* and an *inverse Gaussian high-pass filter*. The equation for the applied bandpass filter $B(x, y)$ can be written as

$$B(x, y) = \exp \left(-\frac{\left(x - \frac{w}{2}\right)^2}{2\left(\frac{w}{\sigma_L}\right)^2} - \frac{\left(y - \frac{h}{2}\right)^2}{2\left(\frac{h}{\sigma_L}\right)^2} \right) \times \left(1 - \exp \left(-\frac{\left(x - \frac{w}{2}\right)^2}{2\left(\frac{w}{\sigma_H}\right)^2} - \frac{\left(y - \frac{h}{2}\right)^2}{2\left(\frac{h}{\sigma_H}\right)^2} \right) \right), \quad (10)$$

where w and h are the width and height of the cross-power spectrum in pixels, and σ_L and σ_H are parameters determining the amount of low-pass and high-pass filtering. An example of such a Gaussian bandpass filter with $\sigma_L = 2$ and $\sigma_H = 7$ is given in [Figure 1](#). The low-pass frequency damping (darker regions near the edges of the spectrum) and the high-pass frequency damping (a dark region near the center of the spectrum) can clearly be seen.

2.4. Subregions and Upscaling

Multiplying the cross-power spectrum $C(x, y)$ with an appropriate bandpass filter $B(x, y)$ element-wise, and computing the inverse Fourier transform $\mathcal{F}^{-1}\{B \cdot C\}$, result in a phase correlation landscape $r(x, y)$. Standard phase correlation algorithms terminate here by returning the location of the maximum element of $r(x, y)$, or alternatively, selecting a small rectangular neighborhood around the maximum element (usually 5×5 pixels) and returning the subpixel coordinates of a centroid computed on that neighborhood ([Druckmüller 2009](#)). First, to improve the accuracy and further boost the effectiveness of the following iterative refinement, the original phase correlation landscape $r(x, y)$ (for later purposes and brevity also denoted as L_3) is upsampled. It is unnecessary and very costly to

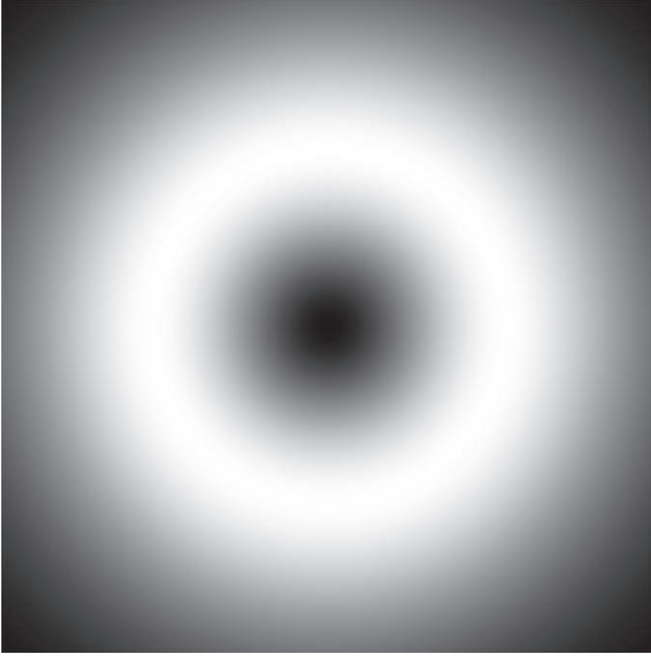


Figure 1. Gaussian bandpass filter mask $B(x, y)$ with $\sigma_L = 2$ and $\sigma_H = 7$, displayed as different shades of gray in linear brightness scale.

upscale the entire landscape L_3 ; therefore, only a subregion around the maximum element is chosen, and then bilinearly upscaled (denoted as L_2). Other interpolation methods were also tested (bicubic interpolation, Lanczos interpolation), but yielded no statistically significant accuracy improvement over the bilinear interpolation. Thus, bilinear interpolation was chosen, as it is the computationally fastest method and does not change the overall region brightness. This is a desirable property for the following iterative refinement, as we are only aiming to get a more accurate maximum correlation location from given values of L_3 , rather than modifying the phase correlation values themselves. After bilinearly upscaling a chosen subregion around the maximum correlation value, another subregion is chosen (this time around the maximum correlation value in L_2), which we denote as L_1 . Contrary to the rectangular regions L_3 and L_2 , region L_1 is chosen to be circular. This is a significantly more natural choice for the final centroid calculation compared to the frequently used rectangular 5×5 window, as the horizontal sections of the correlation peak are always of a circular shape (as seen in Figure 2).

2.5. Iterative Refinement

After computing the circular region L_1 and the rectangular regions of the phase correlation L_3 , L_2 , we begin to iteratively refine the resulting L_2 centroid location. If the usual image coordinate system (x, y) with the origin at the top left corner is considered, and the width/height of the image region L_i are denoted as w_i and h_i , the zero relative shift in the L_i subregion corresponds to the maximum element located at the region center $L_{\text{imid}}[x, y] = \left[\frac{w_i}{2}, \frac{h_i}{2} \right]$. For better clarity, we also denote the current integer x, y pixel coordinates of the correlation peak in a given coordinate system of subregion L_i as $L_{\text{imax}}[x, y]$. The iterative shift refinement process is then described by Algorithm 1.

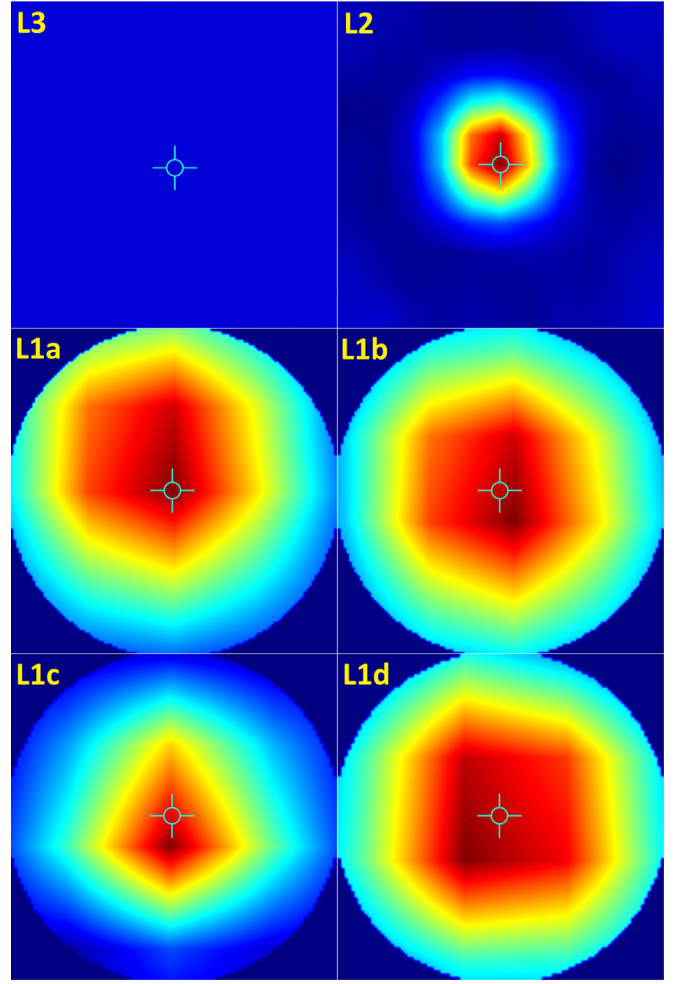


Figure 2. Heatmap of phase correlation maximum neighborhood regions L_i with marked $L_{\text{imax}}[x, y]$ locations (alignment of 4096×4096 full-disk *SDO* 171 Å and 304 Å images, see Section 3.5, L_2 region size equal to 9 px). Left to right, top to bottom: original phase correlation L_3 , rectangular bilinearly upscaled subregion L_2 , circular subregion L_1 (a) before iterative refinement process, L_1 (b) after shift refinement process, L_1 (c) obtained from two (similar) 171 Å images (sharp peak), L_1 (d) obtained from (dissimilar) 171 Å and 304 Å images (broad peak).

Algorithm 1. $L_{2\text{max}}[x, y]$ Iterative Refinement Process

-
1. Initialize $L_{\text{imax}}[x, y]$ for $i = 1, 2, 3$ as maximum correlation coordinates in the corresponding coordinate system of subregion L_i
 2. **while** $|C_x - L_{\text{imid}}[x]| > 0.5$, $|C_y - L_{\text{imid}}[y]| > 0.5$ and iterations $< \text{maximum_iterations}$ **do**
 - replace the previous L_1 circular subregion by a new one, centered around the current $L_{2\text{max}}[x, y]$
 - calculate L_1 centroid coordinates C_x, C_y
 - calculate $L_{1\text{max}}[x, y]$ as rounding C_x, C_y to the nearest integer
 - update $L_{2\text{max}}[x, y]$ by adding the current relative L_1 shift $L_{1\text{max}}[x, y] - L_{\text{imid}}[x, y]$
 - update the number of iterations
- end**
- Result:** return the final $L_{2\text{max}}[x, y]$ along with final L_1 centroid coordinates C_{xf} and C_{yf} (these are now both less than 0.5 pixels away from the L_1 region center)
-

Intuitively, the procedure simply moves the L_1 circular upscaled subregion in the direction of higher correlation, allowing it to selectively explore the L_2 landscape around $L_{2\text{max}}[x, y]$, thus converging to a more accurate solution, as

opposed to a single centroid calculation with neighborhood centered around the maximum correlation value in the case of non-iterative methods. The loop terminates when the difference between L_1 centroid coordinates and $L_{1\text{mid}}[x, y]$ is less than 0.5 pixel in both directions, because one can only move the upscaled L_1 subregion by an integer number of pixels. This is the desired behavior, and the centroid converges. If the reason behind loop termination is to reach the maximum iterations value, the centroid diverges. This is a pathological case that happens only if the resulting phase correlation landscape $r(x, y)$ does not contain a distinct peak to begin with (possible reasons include very low signal-to-noise ratio, wrong bandpass filter parameters).

After obtaining the position of the final L_1 circular subregion ($L_{2\text{max}}[x, y]$) and the final L_1 centroid coordinates C_{xf} and C_{yf} , it is possible to calculate the refined image shift $[\Delta x_r, \Delta y_r]$. Because upscaling was used, refinements need to be scaled down by the amount of upscaling. The resulting refined pixel shift between the two input images can then be evaluated as

$$[\Delta x_r, \Delta y_r] = L_{3\text{max}}[x, y] - L_{3\text{mid}}[x, y] + \frac{1}{U} \times (L_{2\text{max}}[x, y] - L_{2\text{mid}}[x, y] + [C_{xf}, C_{yf}] - L_{1\text{mid}}[x, y]), \quad (11)$$

where $U = 2k+1$, $k \in \mathbb{N}$, $k \geq 1$ is the upscaling coefficient and a centered discrete Fourier transform is assumed (lowest frequencies in the vicinity of $L_{3\text{mid}}[x, y]$).

3. Parameter Optimization

3.1. Universal Parameter Setting

The main parameters of the presented method include L_2 and L_1 region size, upscaling coefficient U , bandpass filter parameters σ_L and σ_H , and input image window type. The L_2 region size should be large enough to allow the circular L_1 region to explore the close proximity of $L_{2\text{max}}[x, y]$ during the iterative phase. The L_1 region size should be chosen according to the phase correlation landscape characteristics—ideally, the entire correlation peak should fit tightly inside the circular region, for the best results. In the example of Figure 2, the size is equal to 35% of the upscaled version of L_2 . The sizes of both regions L_2 and L_1 should be an odd number, so that there exists a central pixel, which corresponds to the maximum correlation amplitude. The upscaling coefficient U should also be an odd number (for the same reason) and large enough to allow for finer refinement. However, there is a limit above which the IPC accuracy does not improve any further with increasing values of U (limit dependent on signal-to-noise ratio). Bandpass filter parameters σ_L and σ_H should be chosen according to the input image characteristics—naturally, if the image contains a high amount of noise, a more pronounced low-pass filtering achieves better results. For the input image windowing method, either rectangular (for images whose edges are generally similar) or Hann window (for images whose edges are generally dissimilar) are recommended.

3.2. Parameter Optimization Process

The universal IPC parameter values, set according to Section 3.1, are general recommended parameter values, as they usually work sufficiently well. However, for applications, where the highest possible alignment accuracy is desired, a

mathematical optimization procedure tailored to a specific kind of input image is essential. Different kinds of input images usually require different IPC parameters. Therefore, ideally, with each type of image data, optimal parameters are found with a suitable optimization algorithm. Because the objective function (as described in Section 3.4) is essentially a *black-box*-type function, an evolutionary method called *Differential Evolution* (Storn & Price 1997) was chosen for this task. This evolutionary continuous parameter optimization method performed sufficiently in a global objective function minimum search.

3.3. The Domain Transforming Function

The main parameters worth optimizing include L_2 (or L_1) size, bandpass filter parameters σ_L , σ_H , and the input image windowing method. These parameters have different domains—bandpass filter coefficients are continuously nonzero, L_2 size is a nonnegative integer, and the window is either rectangular or Hann window (Boolean variable, one of two choices). The standard differential evolution algorithm works with continuous variables only—a construction of transforming function, which transforms the optimized virtual continuous unbounded parameters into their real usable counterparts in corresponding domains is therefore needed. The most natural choice of such a function can be defined as

$$W(x, y) = \begin{cases} H(x, y), & \text{if } w^* > 0 \\ R(x, y), & \text{if } w^* \leq 0, \end{cases} \quad (12)$$

$$\sigma_L = \max(\epsilon, |\sigma_L^*|), \quad (13)$$

$$\sigma_H = \max(\epsilon, |\sigma_H^*|), \quad (14)$$

$$L_2 = \max(3, \lceil L_2^* \rceil), \quad (15)$$

where w^* , σ_L^* , σ_H^* , and L_2^* denote virtual continuous unbounded variables optimized by the differential evolution algorithm, R and H denote rectangular/Hann windows, the symbol $\lceil x \rceil$ denotes rounding x to the nearest integer value, and ϵ denotes a small positive value present to avoid problems arising while dividing by a value close to zero. Passing the virtual continuous parameters through such a function ensures their usability in the IPC algorithm.

3.4. The Objective Function

The last but most important part of correct parameter optimization is the construction of a suitable objective function. The IPC parameters are optimized to improve the subpixel accuracy of the method—it is then natural to define the objective function as an average subpixel error over a fixed amount of *registration trials*. The computation process of an appropriate objective function value is described in Algorithm 2, and an example objective function landscape of a simplified two-parameter version can be seen in Figure 3. This particular 2D simplified objective function landscape was obtained by fixing the L_2 and w^* parameters, while smoothly varying the other two parameters σ_L and σ_H (used as the x -axis and y -axis values). The z -axis value was then set as the registration error measure for the corresponding values of σ_L and σ_H . The global minimum ($\sigma_L = 4.83$, $\sigma_H = 12.14$) of this simplified objective function for a 32×32 *Solar Dynamics Observatory* (SDO)/Helioseismic and Magnetic Imager (HMI) image can clearly be seen.

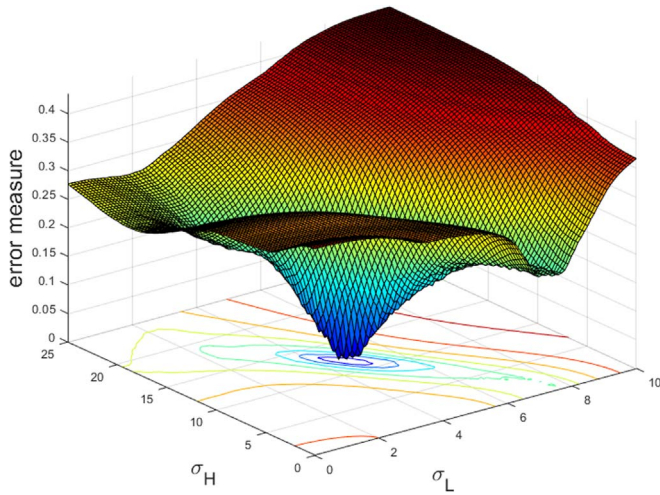


Figure 3. Example of a 2D cut of the four-dimensional IPC parameter objective function described in Algorithm 2 for a 32×32 *SDO*/HMI image (see Section 3.5). The objective function is plotted for varying bandpass parameters σ_L and σ_H with a constant IPC subregion size $L_2 = 15$, and with a Hann window applied to both input images ($w^* > 0$).

Algorithm 2. Objective Function Used for IPC Parameter Optimization

1. Load the appropriate input image
2. Initialize usable IPC parameters by passing their continuous optimized counterparts through a transforming function
3. Extract subregion 1 with predefined size (e.g., 32×32)
4. (optional) Generate artificial noise with characteristics corresponding to the input image wavelength and add it to subregion 1
5. Initialize total subpixel error as 0
6. **while** trials < maximum_trials **do**
 - generate the artificial shift value, either randomly or from an equidistant grid
 - extract subregion 2 with the same size as subregion 1, artificially shifted by a known non-integer amount
 - (optional) generate artificial noise with characteristics corresponding to the input image wavelength and add it to subregion 2
 - calculate the shift between subregion 1 and subregion 2 with IPC
 - update the total subpixel error

end

Result: Return the average subpixel registration error

The artificial subpixel shifts used in the computation of the objective function value can either be randomly distributed uniformly over a given region of interest or arranged in a predefined equidistant $[x, y]$ grid. For non-integer (subpixel) shifts, the artificially shifted image can be obtained with the use of a bilinear interpolation. Artificially generated noise can also be applied to both input images (separately generated for each one) to simulate the presence of real noise between input images.

3.5. Parameter Optimization for *SDO* Images

Concrete examples, for which the IPC parameters were optimized in this paper, include the *SDO* (Pesnell et al. 2011)/HMI images (see Schou et al. 2012), the Atmospheric Imaging Assembly (AIA) images (see Boerner et al. 2011; Lemen et al. 2011), and the *Solar Terrestrial Relations Observatory* (*STEREO* A/B) Sun Earth Connection Coronal and Heliospheric Investigation (SECCHI) images (see Howard et al. 2008).

Table 1

Optimal IPC Parameters (Described in Section 3), Found by a Differential Evolution Algorithm, Optimizing the Objective Function Described in Algorithm 2 with Different Types of Astronomical Input Images (See Section 3.5), $32 \times 32/512 \times 512$ Input Image Size

Instrument	λ	σ_L	σ_H	L_2	W
<i>SDO</i> /HMI	WL	1.2/5.4	70.3/76.2	11/19	Hann
<i>SDO</i> /AIA					
	94 Å	5.5/5.1	8.7/98.1	11/17	Hann
	131 Å	4.9/5.2	9.1/92.3	11/19	Hann
	171 Å	6.5/5.5	1.4/199.0	11/19	Hann
	193 Å	3.8/5.4	10.1/82.9	11/19	Rect
	211 Å	4.9/5.9	1.6/30.3	9/17	Rect
	304 Å	5.9/5.2	4.1/95.9	11/19	Hann
	335 Å	4.9/5.2	9.8/84.3	11/19	Hann
<i>STEREO</i> A SECCHI					
	171 Å	5.1/6.1	1.4/30.1	9/15	Hann
	195 Å	4.8/5.2	1.9/197.1	9/19	Hann
	284 Å	4.9/9.4	1.2/7.4	9/17	Hann
	304 Å	5.1/6.0	1.7/32.7	9/15	Hann
<i>STEREO</i> B SECCHI					
	171 Å	4.9/5.9	1.6/35.0	9/17	Hann
	195 Å	4.5/7.0	1.5/18.1	9/15	Hann
	284 Å	5.1/8.0	1.1/12.8	9/17	Hann
	304 Å	4.9/5.8	2.0/34.6	9/17	Hann

Standard deviations of the applied artificial noise are calculated from dark regions in the corners of input images, as they do not receive solar emission from the optics (being shaded by the filter-wheel mechanism, therefore containing noise only) as described in Lemen et al. (2011). These measured standard deviations (denoted by σ_{MES}) can be different for each detector, and thus are computed separately for each wavelength. Furthermore, the raw maximum and minimum pixel intensities also vary with wavelength, and for every image, the histogram was stretched over the entire 16-bit range (0–65,535) prior to being processed by IPC. Standard deviations therefore need to be normalized equally (denoted σ_N), resulting in a more accurate (relative) measure of noise in the input image, following

$$\sigma_N = \frac{65,535}{\max I(x, y) - \min I(x, y)} \sigma_{MES}. \quad (16)$$

Usable IPC parameters optimized by the differential evolution algorithm for *SDO*/HMI, *SDO*/AIA, and *STEREO* A/B SECCHI images can be seen in Table 1. The use of a bandpass filter proved beneficial for all input image wavelengths. A typical subpixel accuracy improvement, obtained with the use of the optimal IPC parameters (optimized with a mathematical optimization algorithm, such as differential evolution), can be seen in Figure 4, for an *SDO*/AIA 32×32 input image. The optimized IPC registration errors (colored green) are visibly significantly closer to zero, compared to the raw IPC (set by hand) registration errors (colored blue). This illustrates the effectiveness and accuracy improvement achieved by the parameter optimization step.

4. Results

To determine the level of improvement in accuracy of the presented method over the standard methods, a subregion of a

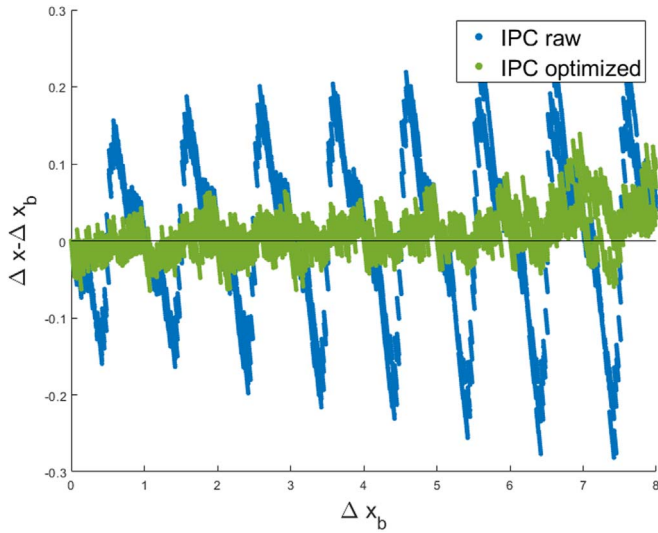


Figure 4. Subpixel errors as a function of actual shifts Δx_b for IPC with optimized parameters and IPC with parameters set by hand for a 32×32 *SDO*/AIA 94 Å image.

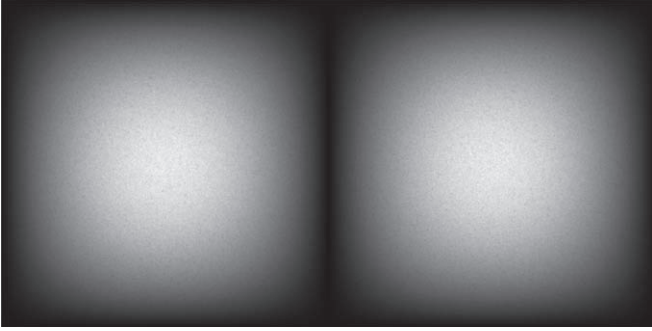


Figure 5. Example of the homogeneous input *SDO*/HMI white-light image subregions $I_1(x, y)$ and $I_2(x, y) = I_1(x + \Delta x_b, y + \Delta y_b)$ with applied Hann window, used for the IPC method subpixel performance analysis.

sample image is manually shifted by a random (but known) amount $[\Delta x_b, \Delta y_b]$. The known image shift is then estimated by the IPC method (and others for comparison). If done many times for many different values of $[\Delta x_b, \Delta y_b]$, this allows for objective method performance evaluation. To also measure subpixel accuracy, the random shift applied to the image is chosen uniformly over an entire finite interval (this is again done with the use of bilinear interpolation, as in Algorithm 2). For this purpose, the chosen sample input data in this paper are *SDO*/HMI and *SDO*/AIA images (similarly to Section 3.5), because of their varying characteristics. For example, *SDO*/HMI WL images are very homogeneous, i.e., they have no apparent distinct features and are impossible for a human eye to correctly align (as can be seen Figure 5). This is contrary to the relatively feature-rich *SDO*/AIA images (Figure 6).

Standard methods of image registration, which the presented method is compared to, include the cross correlation method (CC), phase correlation (PC), and a frequently used non-iterative subpixel version of phase correlation with the centroid calculated on a 5×5 rectangular window around the pixel with maximum correlation (sPC). This version of phase correlation is implemented in the very popular *OpenCV*—*Open source Computer Vision C++* library. The same bandpass filter

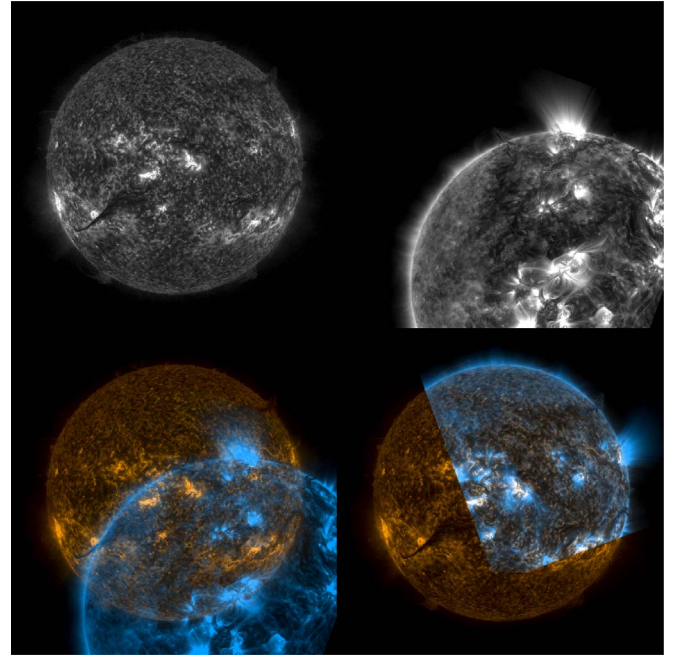


Figure 6. Alignment of artificially shifted, scaled, and rotated *SDO*/AIA 304 Å and 171 Å images. Top to bottom: *SDO*/AIA 304 Å first input image (left), artificially misaligned *SDO*/AIA 171 Å second input image (right), raw first and second image color composition (left), first and second image color composition after the IPC alignment (right).

Table 2
Rounded Sum of 1000 Absolute Subpixel Image Registration Errors of Different Methods for Different Input Image Sizes, *SDO*/HMI WL Images

Image Size	CC	PC	sPC	IPC
32×32	1165	1022	168	36
64×64	256	270	175	36
128×128	253	252	167	18
256×256	250	248	154	12
512×512	245	245	170	12
1024×1024	238	238	165	11

Note. The bold values in the final column are rounded sums of 1000 absolute subpixel image registration errors of the optimized Iterative Phase Correlation (IPC) method calculated with different input image sizes (from 32×32 to 1024×1024 pixels), tested on 1000 (for each image size) uniformly randomly artificially shifted *SDO*/HMI WL images.

and spatial window settings are used for all four tested methods.

The results show that the upscaling and iterative refinement procedures included in the IPC algorithm significantly decrease the bias of resulting shifts being close to an integer. This hugely boosts the overall accuracy of the method (see Table 2). The effect can clearly be seen in Figure 7—the shifts predicted by the IPC algorithm (denoted Δx) are significantly closer to the actual correct shifts (denoted Δx_b), as opposed to the shifts predicted by the frequently used sPC algorithm. This fact is evident at first glance—the IPC registration predictions form a very consistent line, which is very close to the perfect line $\Delta x = \Delta x_b$. The resulting shifts of the IPC method are substantially more accurate for every input image size setting (tested from 32×32 to 4096×4096), making it a good overall replacement for standard correlation methods.

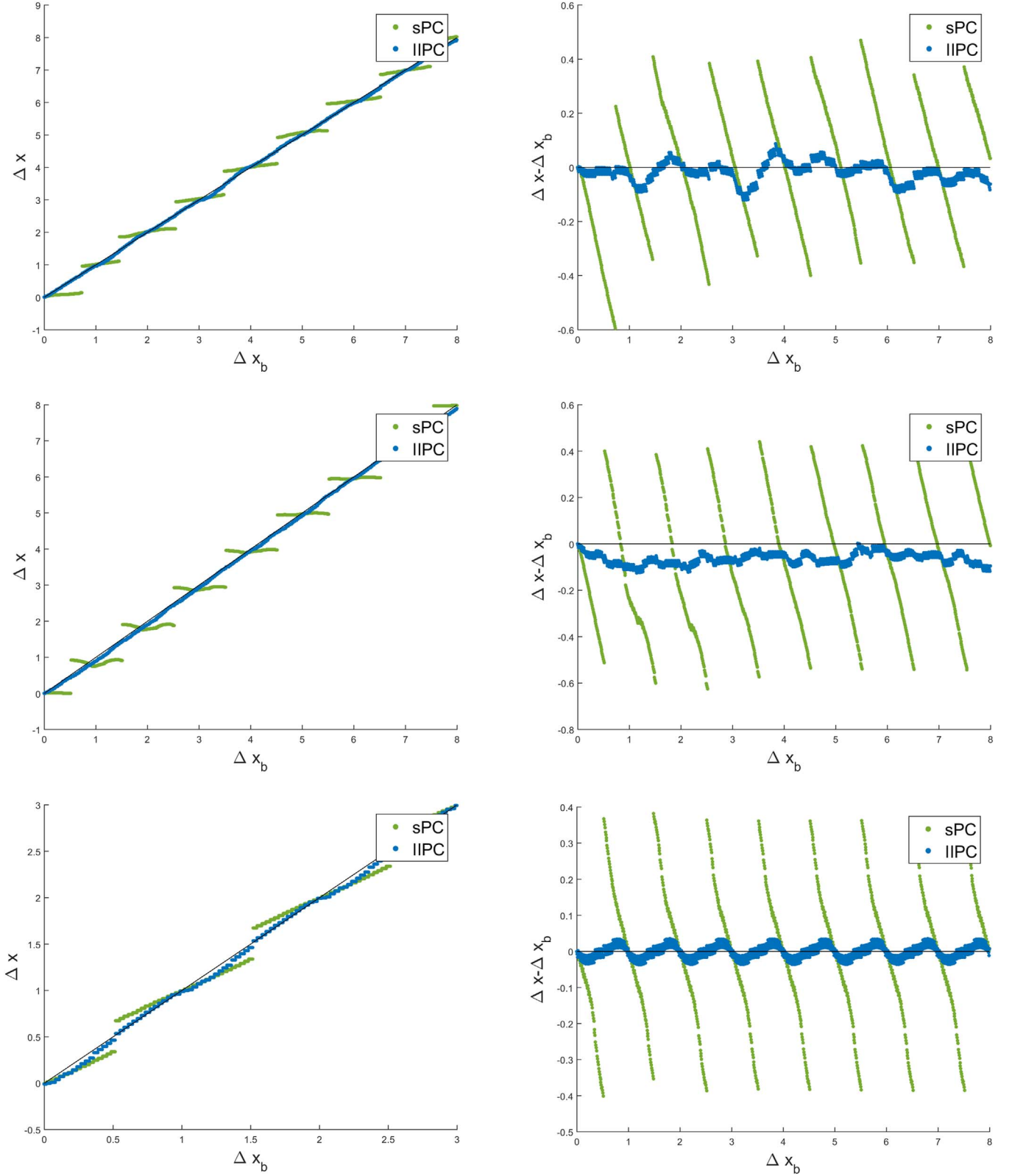


Figure 7. Estimated shifts Δx and subpixel registration errors $\Delta x - \Delta x_b$ as a function of actual shifts Δx_b for different image registration methods, different types of input images, and different input image sizes. Top to bottom: shifts and errors between two 32×32 SDO/HMI white-light images, shifts and errors between two 32×32 SDO/AIA 171 Å images, shifts between (significantly dissimilar) 2048×2048 AIA 171 Å and 304 Å images (left), subpixel registration errors between 2048×2048 HMI white-light images (right). Example source images with applied Hann window, from which the 1000 subregions used for subpixel shift estimation accuracy were extracted, as can be seen in Figure 5.

The real shift, rotation, and scale misalignments of consecutive *SDO* images (e.g., 171 and 304 Å) are visible, but quite small. However, the IPC method is very resilient even to much larger input misalignments. It was also tested to align artificially rotated, shifted, and scaled images, this time by much larger amounts. The quality of the resulting alignment is indistinguishable from the high-precision results achieved with less misaligned images. An example of a high-precision subpixel alignment of extremely artificially misaligned images with very different characteristics and features (*SDO*/AIA 171 and 304 Å) can be seen in Figure 6. In this alignment/composition example, the red color channel is determined only by the first image (304 Å), the blue color channel is determined only by the extremely artificially misaligned second image (171 Å), and the green color channel is set as the average value of the two input images. This color composition channel assignment theoretically results in a perfect black and white composition for identical images (which is not the case in Figure 6), which are aligned precisely.

The IPC method of image alignment is an integral part of a workflow, which is routinely used for the creation of *SDO*/AIA PM-NAFE processed videos. The archive of these videos can be found at <http://www.zam.fme.vutbr.cz/~druck/Sdo/Pm-nafe/0-info.htm>.

5. Summary

The IPC algorithm described in this paper is a powerful, robust, and very accurate method for image registration. This phase correlation extension is useful primarily for tasks where small subpixel differences play a crucial role in the quality of the result. The method was extensively tested and compared with other algorithms for various astrophysical examples, yielding superior results for all tested input image types and sizes, ranging from 32×32 to 4096×4096 16-bit images. The algorithm, implemented in C++ with an easy to use command line interface, through which the user can specify all the IPC parameters (described in Section 3), is available as freeware, together with a manual and all required dependencies at <https://github.com/zdenyhraz/IPC>.

The work of Z.H. and M.D. was supported by the Grant Agency of Brno University of Technology, project No. FSI-S-14-2290.

ORCID iDs

Zdenek Hrazdřa  <https://orcid.org/0000-0001-7362-4947>

Shadia Habbal  <https://orcid.org/0000-0003-4089-9316>

References

- Bing, P., Hui-Min, X., Bo-Qin, X., & Fu-Long, D. 2006, *MeScT*, **17**, 1615
- Boerner, P., Edwards, C., Lemen, J., et al. 2011, *The Solar Dynamics Observatory* (Berlin: Springer), 41
- Casella, A., De Falco, I., Della Cioppa, A., Scafuri, U., & Tarantino, E. 2019, *JPCD*, **133**, 307
- Chen, J., Ji, K., Deng, H., & Feng, S. 2014, *Unifying Electrical Engineering and Electronics Engineering* (Berlin: Springer), 1241
- Do, H. N., Chin, T.-J., Moretti, N., Jah, M. K., & Tetlow, M. 2019, *AdSpR*, **64**, 733
- Druckmüller, M. 2009, *ApJ*, **706**, 1605
- Foroosh, H., Zerubia, J. B., & Berthod, M. 2002, *ITIP*, **11**, 188
- Gravel, P., Verhaeghe, J., & Reader, A. J. 2012, *PMB*, **58**, 105
- Harris, F. J. 1978, *IEEEP*, **66**, 51
- Howard, R., Moses, J., Vourlidas, A., et al. 2008, *SSRv*, **136**, 67
- Jenkinson, M., & Smith, S. 2001, *Medical Image Analysis*, **5**, 143
- Lefebvre, M., & Cohen, L. D. 2001, *Journal of Mathematical Imaging and Vision*, **14**, 131
- Lemen, J. R., Akin, D. J., Boerner, P. F., et al. 2011, *The Solar Dynamics Observatory* (Berlin: Springer), 17
- Leng, C., Zhang, H., Li, B., et al. 2019, *IEEE Access*, **7**, 6424
- Li, C., Lu, B., Zhang, Y., Liu, H., & Qu, Y. 2018, *Neural Processing Letters*, **48**, 1281
- Li, S., Peng, M., Zhang, B., Feng, X., & Wu, Y. 2019, *IJRS*, **40**, 5635
- Mahmood, M. T., & Lee, I. H. 2019, *Applied Sciences*, **9**, 3487
- Pesnell, W. D., Thompson, B. J., & Chamberlin, P. 2011, *The Solar Dynamics Observatory* (Berlin: Springer), 3
- Reddy, B. S., & Chatterji, B. N. 1996, *ITIP*, **5**, 1266
- Schou, J., Scherrer, P., Bush, R., et al. 2012, *SoPh*, **275**, 229
- Shapiro, C., Rowe, B., Goodsall, T., et al. 2013, *PASP*, **125**, 1496
- Storn, R., & Price, K. 1997, *Journal of Global Optimization*, **11**, 341
- Zhou, H., & Yu, Y. 2018, *Frontiers of Computer Science*, **12**, 1013