



Published in [Image Processing On Line](#) on YYYY-MM-DD.
 Submitted on YYYY-MM-DD, accepted on YYYY-MM-DD.
 ISSN 2105-1232 © YYYY IPOL & the authors [CC-BY-NC-SA](#)
 This article is available online with supplementary materials,
 software, datasets and online demo at
<http://www.ipol.im/>

A survey on shift estimation methods for image registration

Martin Rais ¹, Second Author²

¹ CMLA, ENS-Cachan, France (martin.rais@cmla.ens-cachan.fr)

² Department, Institution, Country (username@mailserver.edu)

PREPRINT Thursday 29th March, 2018

Abstract

The problem of global subpixel shift estimation appears frequently on several applications related to image processing. When high accuracy is not required, most authors tend to use the well-known cross-correlation method which usually meets the performance requirements for such tasks. However, under more demanding accuracy constraints, cross-correlation approaches are either not accurate enough or their computational costs are prohibitive. Moreover, problems such as time constraints, varying illumination conditions, aliasing, among several others, make the shift estimation task harder. What is worse, there is little evidence about which shift estimation method to use depending on the circumstances. Our objective in this article is therefore to give an in depth study on shift estimation methods, offering a practical recipe showing which methods are more suited depending on the conditions of the task. By discretizing the distinct varying conditions, we evaluate on each of them more than a thousand variants of shift estimation methods, and summarize the results here. We observed that gradient-based shift estimation methods, used in a multi-scale and iterative form, achieve in general the most accurate results, without requiring considerable computational resources.

Source Code

The source code section briefly explains what the source code published with the article contains, all in a single paragraph. For example: The reviewed source code and documentation for this algorithm are available from the [web page of this article](#)¹. Compilation and usage instruction are included in the `README.txt` file of the archive.

Supplementary Material

The supplementary files section provides explanations about other files published with the article, all in a single paragraph. Mention clearly if they are reviewed or not. For example: A reference dataset, to be used for further comparisons, is provided with the article and peer-reviewed. A Matlab interface (not reviewed) is also available for convenience.

Keywords: first, second, third, fourth

¹<http://www.ipol.im/>

1 Introduction

In this article we study the problem of global sub-pixel shift estimation between images. Given two observations I_1, I_2 of the same image $I(x, y)$, shifted by an unknown displacement $\mathbf{v} = \{v_x, v_y\}$ and affected by noise, the problem of shift estimation is to find this displacement so as to align both images. Applications for shift estimation include medical image registration [52, 42, 58], motion tracking [41], microscopic biology [48], digital image stabilization [28], 3D reconstruction [61], video analysis [8], and are particularly developed in the field of remote sensing [38, 69, 51, 77].

The shift estimation problem presents several challenges. The conditions for this problem to be solved may vary depending on the sensor, the underlying noise model, the effective image size, the magnitude of the displacement, varying illumination between frames, occlusions, aliasing, among others, making shift estimation methods prone to considerable errors depending on their underlying assumptions. Furthermore, the shift estimation task is usually part of a more complex image processing application, which usually constrains its execution time, forcing the shift estimation method to be both fast and accurate.

In this article we tackle the case where both images I_1 and I_2 have the same $N \times N$ size, are both contaminated by white Gaussian noise with the same standard deviation σ . We always assume that the 2D displacement is smaller than one pixel in both dimensions and that the brightness constancy constrain holds, i.e.,

$$I_2(x, y) = I_1(x + v_x, y + v_y). \quad (1)$$

Observing the behavior of each method under these ideal conditions is sufficient since most problems mentioned above can be reduced to this setting. In particular,

- **Signal dependent noise.** Heteroscedastic noise can be turned homoscedastic by prefiltering the images with a Variance Stabilization Transform (VST) [85]. For example, the well-known Anscombe transform [5], as well as the Generalized Anscombe transform convert Poisson or Poisson + Gaussian noise into approximately standard Gaussian noise making the standard deviation almost constant [85]. Other well-known variance stabilization transformations include the Freeman-Tukey [33], Barlett [14], Curtiss [22] and Eisenhart [26]. In fact, there is a whole family of VSTs for the Poisson distribution, described by Bar-Lev and Enis [13].
- **Varying illumination between frames.** A histogram matching method should be performed previous to the shift estimation. In many contexts, such as satellite imaging, the change in illumination is multiplicative, therefore a fast multiplicative mean equalization should be done. For non-linear changes in illumination, images can be equalized by using the midway equalization method [24]. We refer the reader to further histogram matching approaches [37, 57].
- **Different image sizes.** In template matching applications [18], an image patch of size $N \times N$ has to be aligned with a template or reference image of size $M \times M$ with $N < M$. This is in general a less challenging problem due to the availability of image borders on the reference image. In such situations, any fast method achieving pixel precision [2, 65] could be applied in order to extract a $N \times N$ subimage from the reference, followed by applying a subpixel shift estimation method.
- **Large shift magnitudes.** As in the previous case, large shift magnitudes could be estimated in two steps. After performing an initial pixel-wise shift estimation, any of the methods discussed in this article could be used on the intersection of both pixel-aligned images. This includes projection-based methods [2, 65], pyramidal multi-scale approaches [88, 68] or fast periodic cross correlation methods in the Fourier domain [66].

While several methods exist up to date, not many achieve acceptable results on every possible condition. To our understanding, several survey papers exist summarizing these methods [89, 97, 58, 17]. However, none of them offer a performance comparison under the different challenging conditions that may occur, which ends up with the eventual user to pick the wrong method. The contribution of this article is therefore twofold. First, an in-depth study of the state-of-the-art shift estimation methods is given, followed by a practical recipe defining which methods to use depending on the particular conditioning of the problem.

1.1 Subpixel shift estimation approaches

As mentioned in Tian *et al.* [89], there are mainly four types of shift estimation methods that achieve subpixel accuracy: correlation interpolation, intensity interpolation, differential methods and phase correlation. Recently, feature-based image registration has become extremely popular since the introduction of local feature detection/description methods.

Correlation Interpolation Methods. In order to achieve subpixel accuracy using discrete correlation, an interpolation surface is fitted to samples of this function, and then, the maximum of this surface is searched. When the images are sampled at a high enough frequency, the corresponding discrete correlation function is quite smooth and using a second-order interpolation function can obtain accurate results. This methodology not only implies calculating the discrete correlation between images, which is a resource consuming task, but also to interpolate it.

Intensity Interpolation Methods. Another way to achieve subpixel results is to interpolate selected parts of the input images in order to create a much denser grid. Then, the task is to match these grids between images. This task requires knowing beforehand, which part of the input images to interpolate and then match, something which is not always available. Several searches have to be made in order to find good matches, making this approach time consuming.

Differential Methods. Also known as Gradient-Based Shift Estimation (GBSE), the idea behind this methodology is to relate the difference between two successive frames to the spatial intensity gradient of the first image. This relationship is obtained by approximating the shifted image by its Taylor development up to the first order. Since the higher terms of the Taylor approximation are removed, this relation holds only when the displacement is shorter than a pixel. By performing such an approximation, the shift can then be easily estimated using linear least squares minimization. However, more elaborate methods can be applied to achieve higher precision. This type of methods have linear complexity and are in general considerably faster than both interpolation-based methods. Furthermore, several extensions allow it to become more robust against noise and larger displacements.

Phase correlation Methods. This methodology is based on the fact that most of the information about the relative displacements of objects between two images is contained in the phase of their cross-power spectrum. Several methods exploit this to estimate the displacement both in the Fourier and the spatial domain. These methods require the computation of the discrete Fourier transform (DFT), which could be prohibitive in certain scenarios. Nevertheless, they are able to achieve very accurate results without further computational effort, and they can be used when images are seriously distorted, in either geometry or intensity. Due to this reason, several methods based on this idea have been recently published [70, 86, 32, 87, 39, 6, 7, 93, 92, 73, 72]. As a drawback, phase correlation approaches assume circular shifts between images, i.e. $I_1(x, y) = I_2((x - v_x) \bmod M, (y - v_y) \bmod N)$ (where the images are $M \times N$ in size). In fact, the cross-power spectrum in the frequency domain is the analog of the cross-correlation operator between two signals. Since in general, the shift between both images is a simple linear shift rather than a circular shift, windowing functions are usually employed to mitigate problems related to image edges, by performing apodization of the input images. For images that have repeating objects, phase correlation may also yield ambiguous results with

several correlation peaks, and its use on those cases is not advisable.

Feature Based Methods. This approach, as pointed out by [97], is based on the extraction of salient structures or features, that are supposed to be stable. Then, these features are matched between images by either comparing feature descriptors, or similarity measures combined with spatial relationships. Once the matches are calculated, a transformation model is estimated in order to align both images, using the computed matches. The idea for the features is to be distinct, spread all over the image, and efficiently detectable in both images. They are expected to be stable in time to stay at fixed positions during the whole experiment. One algorithm used to describe and match features is the famous SIFT method proposed by Lowe [55]. It gets subpixel accuracy of the features by interpolating the response of the estimated Laplacian over the pixels.

The stability of the features decreases in the presence of noise and in consequence poor registration results are obtained. What is more, although these methods are efficient when they work on a multiscale approach, they are not aimed at achieving accurate subpixel measurements and because of this reason, are not considered in this article.

1.2 Summary

This article offers a detailed analysis on fast and accurate shift estimation methods. With few exceptions, we discard every shift estimation method that augments the images resolutions using interpolation to reach sub-pixel accuracy. Although these methods may in practice achieve accurate results, they are computationally expensive. For this reason, we focus only on differential methods and phase correlation approaches, offering a complete study on both methodologies.

We observed that gradient-based shift estimation (GBSE) approaches proved to be more accurate and tolerant to noise than phase-correlation methods. They are also computationally cheaper and more stable. However, when the estimated shifts are large, they must be used within pyramidal multiscale approaches which tend to reduce their stability. Although more computationally intensive, gradient correlation methods offer more accurate and consistent results than phase correlation approaches. Their accuracy suffers less under lower SNR scenarios. We refer the reader directly to the conclusions yielded from sections 4.2.3, 4.4.1, 4.5 and 4.6 for a brief summary.

We begin this article by giving an overview of the most important fast and accurate shift estimation methods, starting with gradient-based approaches (section 2) followed by phase-correlation-based methods (section 3). We perform a full evaluation of the analyzed methods under different challenging situations in section 4 and give a practical recipe by giving the best approaches to use under those. We conclude and offer some envisaged future work in section 5.

2 Gradient-Based Shift Estimation methods

Given two images where the brightness constraint holds, namely $I_2(x, y) = I_1(x + v_x, y + v_y)$ where v_x and v_y are the unknown shift coefficients, then using the first order Taylor approximation and assuming the displacement to be small, we have that

$$\begin{aligned} I_2(x, y) - I_1(x, y) &\approx v_x \frac{\partial I_1(x, y)}{\partial x} + v_y \frac{\partial I_1(x, y)}{\partial y}, \\ I_t(x, y) &\approx \nabla I_1(x, y)^T \mathbf{v} \end{aligned} \tag{2}$$

where $I_t = I_2 - I_1$ is the discrete temporal derivative, $\nabla I_1(x, y) = \left[\frac{\partial I_1(x, y)}{\partial x}, \frac{\partial I_1(x, y)}{\partial y} \right]$ the image gradient and $\mathbf{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$ corresponds to the unknown shift.

Note that even though in practice the brightness constraint rarely holds exactly, it works remarkably well in real-life applications [31]. Eq. (2) relates the difference between the two successive frames to the spatial intensity gradient of the first image for a single pixel. This equation is known as the optical flow equation or the gradient constraint equation. Since there is one equation and two unknowns, the shift cannot be determined, therefore yielding the necessity to add additional constraints. In an optical flow application, the flow for each pixel may be different, therefore many variational approaches based on different possible constraints exist [96, 15, 20]. However, for the shift estimation problem between two images, since all pixels supposedly share the same displacement, each (x, y) location becomes an equation and the problem becomes over-determined.

2.1 Optical Flow equation with Least Squares minimization

The Lucas-Kanade algorithm [56] is probably the most widely known gradient-based method used to estimate the optical flow between two images. Based on Eq. (2), it assumes a constant displacement for every pixel around its neighborhood, which allows the construction of an overdetermined system $\mathbf{Av} = \mathbf{b}$, where \mathbf{A} is composed of spatial intensity derivatives and \mathbf{b} has temporal derivatives

$$\mathbf{A} = \begin{bmatrix} \frac{\partial I_1}{\partial x}(p_1) & \frac{\partial I_1}{\partial y}(p_1) \\ \vdots & \vdots \\ \frac{\partial I_1}{\partial x}(p_N) & \frac{\partial I_1}{\partial y}(p_N) \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} I_t(p_1) \\ \vdots \\ I_t(p_N) \end{bmatrix} \quad (3)$$

and p_i with $i = 1 \dots N$ represents the i th pixel and N the number of pixels. The solution \mathbf{v} to this overdetermined system is obtained by performing the linear least squares method, using the Moore-Penrose pseudo-inverse. Then, let I_x, I_y be an abbreviation for $\frac{\partial I_1}{\partial x}$ and $\frac{\partial I_1}{\partial y}$ respectively, the shift is computed by

$$\hat{\mathbf{v}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}, \quad (4)$$

where

$$\mathbf{A}^T \mathbf{A} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \quad (5)$$

is the second moment matrix, and

$$\mathbf{A}^T \mathbf{b} = \begin{bmatrix} \sum I_t I_x \\ \sum I_t I_y \end{bmatrix} \quad (6)$$

is a spatio-temporal gradient correlation term. To solve this system, the matrix $\mathbf{A}^T \mathbf{A}$ must be invertible. Although this method was designed to compute the optical flow between two images, this same idea could be directly used as a shift estimation method by simply considering the neighborhood of each pixel as the whole image.

It is not a coincidence that the results of the method depend on the inversion of this second moment matrix since the determinant of this matrix is crucial for determining the limits on the estimation performance. A study on this matrix before actually performing the shift estimation could potentially be used to discard ill-posed cases, in which, for example, the gradient occurs on its majority on a single direction and thus we are dealing with a potentially unsolvable situation, known as the aperture problem.

Note that, since the Taylor development is centered at zero, this method performs well only when the translation is subpixel, i.e., estimated shifts larger than 1 would not be correctly estimated. For dealing with cases in which the shift is larger, the estimation must be recursively performed on zoomed-out versions of both images, followed by subsampling on the next scale. This method will be addressed in section 2.3.

2.2 Optical Flow Equation with Total Least Squares Minimization

To solve the optical flow equation (2), the least squares (LS) method can be employed yielding accurate results by using a fast closed form solution. It allows for noise in the independent term \mathbf{b} , yet it does not take into account both the noise in the spatial derivative matrix \mathbf{A} , coming from the image acquisition and sampling processes, as well as the fact that the derivatives themselves are also approximations (i.e. estimated as differences between neighboring pixels).

In order to deal with noise in the spatial derivatives and to account for approximation errors, an improved technique, named total least squares (TLS) [45, 59], can be used. This method assumes perturbations both in the temporal derivative (the independent term \mathbf{b}) as well as in the spatial derivatives of the first image (matrix \mathbf{A}). To explain this in a more intuitive way, we shall rewrite first the Least Squares (LS) problem as

$$\{\hat{\mathbf{x}}_{LS}, \Delta\mathbf{b}_{LS}\} := \arg \min_{\mathbf{x}, \Delta\mathbf{b}} \|\Delta\mathbf{b}\|_2 \quad \text{subject to } \mathbf{Ax} = \mathbf{b} + \Delta\mathbf{b}, \quad (7)$$

where the idea is to allow for an error in the independent term \mathbf{b} by minimizing over $\Delta\mathbf{b}$, so that the corrected system of equations $\mathbf{Ax} = \hat{\mathbf{b}}, \hat{\mathbf{b}} := \mathbf{b} + \Delta\mathbf{b}$ has an exact solution. Then if $\mathbf{A}^T\mathbf{A}$ is invertible, the unique solution $\mathbf{x}_{LS} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ of the optimally corrected system of equations $\mathbf{Ax} = \hat{\mathbf{b}}_{LS}, \hat{\mathbf{b}}_{LS} := \mathbf{b} + \Delta\mathbf{b}_{LS}$ is by definition the least squares approximate solution of the original incompatible system of equations.

As can be seen from the above example, \mathbf{b} is corrected while \mathbf{A} is not, thus ignoring possible errors and noise in the calculation of the derivatives. To deal with them, the total least squares method assumes uncorrelated noise in both the independent term \mathbf{b} as well as in the spatial derivative matrix \mathbf{A} . Indeed, total least squares assumes the error in each element of matrix \mathbf{A} and the vector \mathbf{b} to be independent and identically distributed (i.e. the error matrix is white), while performing no assumptions on the distribution of the noise. If however the noise turns out to be correlated, total least squares can actually perform worse than standard least squares. However, TLS finds the true solution when the amount of optical flow equations N goes to infinity, and it out-performs LS estimation as N is sufficiently large [94].

It is well-known that the least-squares approximation is statistically motivated as a maximum likelihood estimator in a linear regression model under zero mean and a normally distributed residual with a covariance matrix that is a multiple of the identity [40]. Similarly, the total least-squares approximation is a maximum likelihood estimator in the errors-in-variables (EIV) model, namely

$$\tilde{\mathbf{A}} = \mathbf{A} + \Delta\tilde{\mathbf{A}}, \quad \tilde{\mathbf{b}} = \mathbf{b} + \Delta\tilde{\mathbf{b}}, \quad \exists \mathbf{x} \text{ such that } \mathbf{Ax} = \mathbf{b}. \quad (8)$$

under the assumption that the augmented matrix $[\Delta\tilde{\mathbf{A}} | \Delta\tilde{\mathbf{b}}]$ is a zero mean, normally distributed random vector with a covariance matrix that is a multiple of the identity [59].

Theorem 1. *The solution to the total least squares problem consisting in calculating*

$$\begin{aligned} \{\hat{\mathbf{x}}_{TLS}, \Delta\mathbf{A}_{TLS}, \Delta\mathbf{b}_{TLS}\} &:= \arg \min_{\mathbf{x}, \Delta\mathbf{A}, \Delta\mathbf{b}} \|[\Delta\mathbf{A} \ \Delta\mathbf{b}]\|_F \\ &\text{subject to } (\mathbf{A} + \Delta\mathbf{A})\mathbf{x} = \mathbf{b} + \Delta\mathbf{b} \end{aligned} \quad (9)$$

where $\|[\Delta\mathbf{A} \ \Delta\mathbf{b}]\|_F$ is the Frobenius norm of the augmented matrix with matrix $\Delta\mathbf{A}$ and the vector $\Delta\mathbf{b}$ side by side, is given by

$$\hat{\mathbf{x}}_{TLS} = -\frac{(v_1, v_2)^T}{v_3} \quad (10)$$

where $V = (v_1, v_2, v_3)^T$ is the 3×1 right singular vector associated with the smallest singular value of the augmented matrix $[\mathbf{A} | \mathbf{b}]$.

Proof. Minimizing Eq. (9) is the same as minimizing $\|[\Delta\mathbf{A} \ \Delta\mathbf{b}]\|_F$ subject to $\mathbf{b} + \Delta\mathbf{b} \in \text{Range}(\mathbf{A} + \Delta\mathbf{A})$. Once the minimum $(\widehat{\Delta\mathbf{A}}, \widehat{\Delta\mathbf{b}})$ is found, any x satisfying

$$(\mathbf{A} + \widehat{\Delta\mathbf{A}})x = \mathbf{b} + \widehat{\Delta\mathbf{b}} \quad (11)$$

is said to solve the TLS problem. To seek for a solution, we can generalize the problem for \mathbf{b} to be a matrix \mathbf{B} of $n \times k$ elements. Then Eq. (11) can be rewritten as

$$[(\mathbf{A} + \Delta\mathbf{A}) \ (\mathbf{B} + \Delta\mathbf{B})] \begin{bmatrix} x \\ -I_k \end{bmatrix} = 0, \quad (12)$$

where I_k is the $k \times k$ identity matrix. We shall now show that solving the problem is equivalent to finding $[\Delta\mathbf{A} \ \Delta\mathbf{B}]$ that reduces the rank of $[\mathbf{A} \ \mathbf{B}]$ by k . Define $[\mathbf{U}][\Sigma][\mathbf{V}]^*$ to be the singular value decomposition of the augmented matrix $[\mathbf{A} \ \mathbf{B}]$

$$[\mathbf{A} \ \mathbf{B}] = [\mathbf{U}_A \ \mathbf{U}_B] \begin{bmatrix} \Sigma_A & 0 \\ 0 & \Sigma_B \end{bmatrix} \begin{bmatrix} V_{AA} & V_{AB} \\ V_{BA} & V_{BB} \end{bmatrix}^*, \quad (13)$$

where V is partitioned into blocks corresponding to the shape of \mathbf{A} and \mathbf{B} . Using the Eckart-Young-Mirsky theorem [36, 25], the approximation minimizing the Frobenius norm of the error in Eq. (9) is such that matrices \mathbf{U} and \mathbf{V} are unchanged, while the k -smallest singular values are replaced with zeroes. That is

$$[(A + \Delta\mathbf{A}) \ (B + \Delta\mathbf{B})] = [\mathbf{U}_A \ \mathbf{U}_B] \begin{bmatrix} \Sigma_A & 0 \\ 0 & 0_{k \times k} \end{bmatrix} \begin{bmatrix} V_{AA} & V_{AB} \\ V_{BA} & V_{BB} \end{bmatrix}^*, \quad (14)$$

so by subtracting Eq. (13) with Eq. (14), we get

$$[\Delta\mathbf{A} \ \Delta\mathbf{B}] = -[\mathbf{U}_A \ \mathbf{U}_B] \begin{bmatrix} 0_{n \times n} & 0 \\ 0 & \Sigma_B \end{bmatrix} \begin{bmatrix} V_{AA} & V_{AB} \\ V_{BA} & V_{BB} \end{bmatrix}^*. \quad (15)$$

We can then remove blocks from the \mathbf{U} and Σ matrices, simplifying to

$$[\Delta\mathbf{A} \ \Delta\mathbf{B}] = -\mathbf{U}_B \Sigma_B \begin{bmatrix} V_{AB} \\ V_{BB} \end{bmatrix}^* = -[A \ B] \begin{bmatrix} V_{AB} \\ V_{BB} \end{bmatrix} \begin{bmatrix} V_{AB} \\ V_{BB} \end{bmatrix}^*. \quad (16)$$

This provides $\Delta\mathbf{A}$ and $\Delta\mathbf{B}$ so that

$$[(A + \Delta\mathbf{A}) \ (B + \Delta\mathbf{B})] \begin{bmatrix} V_{AB} \\ V_{BB} \end{bmatrix} = 0. \quad (17)$$

Now if V_{BB} is nonsingular, we can then right multiply both sides by $-V_{BB}^{-1}$ to bring the bottom block of the right matrix to the negative identity, giving

$$\begin{aligned} & [(A + \Delta\mathbf{A}) \ (B + \Delta\mathbf{B})] \begin{bmatrix} -V_{AB}V_{BB}^{-1} \\ -V_{BB}V_{BB}^{-1} \end{bmatrix} = \\ & = [(A + \Delta\mathbf{A}) \ (B + \Delta\mathbf{B})] \begin{bmatrix} x \\ -I_k \end{bmatrix} = 0, \end{aligned} \quad (18)$$

and so the solution of the TLS problem for this case is

$$\hat{\mathbf{x}}_{TLS} = -V_{AB}V_{BB}^{-1}. \quad (19)$$

Since in our particular problem the matrix B is the vector \mathbf{b} of $N \times 1$, then V_{BB} is a scalar and V_{AB} is a 2×1 vector, therefore the overall cost of this procedure is dominated by the singular value decomposition and the solution becomes

$$\hat{\mathbf{x}}_{TLS} = -\frac{(v_1, v_2)^T}{v_3}. \quad (20)$$

□

Optical flow estimation using total least squares instead of the conventional least squares method is not new. Already in 1995, Weber and Malik [95] used total least squares to solve the over-determined optical flow problem. They also used as a reliability measure, the consistency ratio $\frac{\sigma_3}{\sigma_2}$ which is the division between the two smallest singular values of the augmented matrix.

Tsai et al. [91] applied total least squares to estimate stereo optical flow and used σ_3 , the smallest singular value of the augmented matrix $[\mathbf{A} \ \mathbf{b}]$, as a reliability measure of the estimates. They also regularized the estimated flow field based on the confidence provided by the value of σ_3 . Similarly Bab-Hadiashar and Suter [9, 10] used the Least Median of Squares Orthogonal Distances (LMSOD) to identify the outliers and then total least squares to solve the optical flow problem. Finally, more recently Fashandi et al. [30] have proposed to estimate an optical flow field based on a wavelet decomposition and to use total least squares because of approximations performed on both sides of their over determined equation system.

2.3 Bias minimization through iterative and multiscale gradient-based shift estimation

In a least squares problem configuration, the linear least squares produces robust but not very accurate solutions since it ignores the noise in the spatial derivative matrix \mathbf{A} . Total least squares takes into account the noise implicitly contained in the matrix \mathbf{A} . However, when the noise is not independent or identically distributed or when the system is highly inconsistent (i.e. the determinant of the second order matrix $\mathbf{A}^T \mathbf{A}$ is small), TLS tends to give noisier estimates greatly affecting its robustness [62]. This is because TLS deals with the errors in \mathbf{A} and \mathbf{b} symmetrically. If all the errors in \mathbf{A} and \mathbf{b} are identical and independent, or their ratio can be obtained, then the TLS estimation is asymptotically unbiased [46]. However, since \mathbf{A} is composed of spatial gradients which are estimated using numerical differentiation, the noise we expect on both variables \mathbf{A} and \mathbf{b} is correlated between neighboring pixels, which causes further problems for TLS. In fact, by ignoring the noise, the estimator proposed in (4) is systematically biased, meaning that its expected value is not equal to the true shift. Notice, there are no particular assumptions on the noise; it only needs to be symmetric around the true value.

A second source of bias for the estimator (4) is the Taylor approximation. Indeed, this method is derived from a Taylor approximation by truncating the Taylor series after the first order derivative. This approximation is accurate only when the second and the higher order derivatives are small. As a result, there is a systematic bias that depends on the image content and the displacement itself. We hereon present differential shift estimation methods that try to reduce its influence indirectly, without explicitly dealing with each of the mentioned reasons.

Instead of dealing with the bias explicitly, it was shown in Pham et al. [64] that both bias sources depend linearly on the shift magnitude. This justifies the use of an iterative method, which is able to significantly reduce the bias, provided an appropriate resampling method is used. This algorithm is described in Alg. 1, where k is the maximum amount of iterations, *findshift* uses (4) to solve for $v[i]$ and *Resample* resamples an image by interpolation. The selected interpolation algorithm

Algorithm 1 Iterative GBSE method.

```

1: procedure ILK( $I_1, I_2$ )                                 $\triangleright$  Receives a pair of images
2:    $i \leftarrow 0$ ;  $I_2[0] \leftarrow I_2$ ;  $w \leftarrow 0$ ;
3:   while  $i \leq k$  do
4:      $v[i] \leftarrow \text{findshift}(I_1, I_2[i])$             $\triangleright$  Eq. (4)
5:      $w \leftarrow w + v[i]$                                  $\triangleright$  Accumulate total shift
6:      $I_2[i + 1] \leftarrow \text{Resample}(I_2, -w)$          $\triangleright$  Use original  $I_2$ 
7:      $i \leftarrow i + 1$ 
8:   end while
9:   return  $w$                                           $\triangleright$  Return accumulated shift
10: end procedure

```

could become a limiting factor to achieve high final accuracy, therefore this decision should not be underrated.

Note that while in the original formulation of Lucas and Kanade, the resampling was performed on the same image from where they computed the gradient (I_1), in the present formulation the gradient of image I_1 is computed only once, while the resampling is always performed on the second image, thus avoiding to recalculate it for every iteration. This method is known as the Inverse Compositional Algorithm [11]. It should also be noted that instead of resampling using the already resampled image, it is always performed taking the original I_2 , thus reducing the negative effects of inexact resampling methods and accelerating the process.

It was proven in Pham et al. [64] that this iterative scheme is able to significantly reduce the bias and to make the iterative GBSE estimator practically unbiased. Nevertheless, resampling of highly aliased images could end up violating the brightness constancy constraint, making a single iteration to outperform an iterative scheme, as shown in [63]. However, when a correct resampling is possible and with enough iterations, this method is the only one capable, to the best of our knowledge, to achieve optimal results with no bias.

Another element to consider is when the underlying displacements are larger than one pixel. In this case and as mentioned before, the presented GBSE method fails. However, by building a pyramid representation of the input images, Eq. (4) can be applied on each scale to estimate the shift between images, and this estimated shift can in turn be used to resample the second image on the following level of the pyramid [88]. If more accuracy is desired, an iterative scheme (as of Algorithm 1) could be used to better estimate the shift in each scale, although this comes with an increase in computational cost. We used in our implementation a dyadic Gaussian pyramid approximation [19], however we also tried an exact dyadic Gaussian pyramid [60] filtering with $\sigma = 1.4$ before subsampling, but we found out that the results were similar. Starting from the coarse image at scale $s > 1$, the method is presented in Algorithm 2.

2.4 Minimizing the bias through corrected gradient estimation

Assuming that the noise is Gaussian white noise with variance σ^2 and does not depend on the image intensity, Ji et al. [46] proposed to use a simple and straightforward bias correction technique. In their work, they show that the bias of the least squares estimator amounts to

$$\begin{aligned}
\text{bias}(\hat{\mathbf{x}}_{LS}) &= \lim_{n \rightarrow \infty} E(\hat{\mathbf{x}}_{LS} - \mathbf{x}) \\
&= -\sigma^2 \left(\lim_{n \rightarrow \infty} \left(\frac{1}{N} \mathbf{A}^T \mathbf{A} \right) \right)^{-1} \mathbf{x},
\end{aligned} \tag{21}$$

Algorithm 2 Multiscale GBSE method.

```

1: procedure MSSE( $I_1, I_2, s$ )                                 $\triangleright$  Receives a pair of images and amount of scales
2:    $I_1^{1 \dots s} \leftarrow \text{BuildPyramid}(I_1, s)$            $\triangleright$  Burt&Adelson's Gaussian Pyramid [19]
3:    $I_2^{1 \dots s} \leftarrow \text{BuildPyramid}(I_2, s)$            $\triangleright$  i.e., IMPYRAMID function from Matlab
4:    $i \leftarrow s; w \leftarrow 0$ 
5:   while  $i > 0$  do
6:      $v(i) \leftarrow \text{findshift}(I_1^i, I_2^i)$                    $\triangleright$  Eq. (4)
7:     or  $v(i) \leftarrow \text{ILK}(I_1^i, I_2^i)$                    $\triangleright$  Alg. 1
8:      $w \leftarrow w * 2 + v(i) * 2$ 
9:      $I_2^{i-1} \leftarrow \text{Resample}(I_2^{i-1}, -w)$ 
10:     $i \leftarrow i - 1$ 
11:   end while
12:    $v(i) = \text{findshift}(I_1^1, I_2^1)$ 
13: end procedure

```

where \mathbf{A} and \mathbf{x} are obtained assuming noiseless images and N is the total amount of pixels in the image. To reduce the influence of the noise on the computations, they propose the corrected least squares estimator (CLS), given by

$$\mathbf{x}_{\text{CLS}} = \left(\tilde{\mathbf{A}}^T \tilde{\mathbf{A}} - N\sigma^2 \mathbf{I} \right)^{-1} \left(\tilde{\mathbf{A}}^T \tilde{\mathbf{b}} \right), \quad (22)$$

where $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{b}}$ are the noisy versions of \mathbf{A} and \mathbf{b} respectively, and \mathbf{I} is the 2×2 identity matrix. Thus, provided a correct noise estimation is possible, then by simply subtracting an approximation of the error introduced by the noise from the biased components of the second order matrix, it is possible to attenuate the bias. In practice, however, this is a difficult task because noise in $\tilde{\mathbf{A}}$ is not always Gaussian nor is it uncorrelated with the signal. Furthermore, the noise-corrected second-order matrix in Eq. (22) is not necessarily a good approximation of the noiseless matrix $\mathbf{A}^T \mathbf{A}$ when the underlying image is aliased or has been pre-filtered [63]. Last but not least, the noise variance is not always known or easy to estimate.

2.5 Bidirectional bias correction for Gradient-Based Shift Estimation

To improve the accuracy of GBSE methods using a single iteration, an approach by Pham and Duggan [63] tries to estimate the unbiased second moment matrix $\mathbf{A}^T \mathbf{A}$ from the noisy images. Let $\tilde{I} = I + n$ be the noisy image where n denotes additive white Gaussian noise. Rewriting the least squares solution of the optical flow equation (4) as $\tilde{\mathbf{A}}\tilde{\mathbf{v}} = \tilde{\mathbf{b}}$ where

$$\underbrace{\begin{bmatrix} \sum \tilde{I}_x^2 & \sum \tilde{I}_x \tilde{I}_y \\ \sum \tilde{I}_x \tilde{I}_y & \sum \tilde{I}_y^2 \end{bmatrix}}_{\tilde{\mathbf{A}}} \underbrace{\begin{bmatrix} \tilde{v}_x \\ \tilde{v}_y \end{bmatrix}}_{\tilde{\mathbf{v}}} = \underbrace{\begin{bmatrix} \sum \tilde{I}_t \tilde{I}_x \\ \sum \tilde{I}_t \tilde{I}_y \end{bmatrix}}_{\tilde{\mathbf{b}}}, \quad (23)$$

and the tilde indicates the noisy version, the authors point out that, given n_x, n_y the directional derivatives of the noise n , the expected value of matrix $\tilde{\mathbf{A}}$ is given by

$$E[\tilde{\mathbf{A}}] = \begin{bmatrix} \sum I_x^2 + \text{var}(n_x) & \sum I_x I_y + \text{cov}(n_x, n_y) \\ \sum I_x I_y + \text{cov}(n_x, n_y) & \sum I_y^2 + \text{var}(n_y) \end{bmatrix} \quad (24)$$

therefore $E[\tilde{\mathbf{A}}] \neq \mathbf{A}$ making the computation of the derivative matrix \mathbf{A} to be usually overestimated. Since $E(\tilde{\mathbf{b}}) = \mathbf{b}$, it makes sense to assume $\tilde{\mathbf{b}} \approx \mathbf{b}$, so we have that

$$\tilde{\mathbf{A}}\tilde{\mathbf{v}} = \tilde{\mathbf{b}} \approx \mathbf{b} = \mathbf{A}\mathbf{v} \quad (25)$$

from which we conclude that

$$\tilde{\mathbf{A}}\tilde{\mathbf{v}} = \mathbf{A}\mathbf{v}. \quad (26)$$

Given that both $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{v}}$ can be estimated using the noisy image and by solving for $\tilde{\mathbf{v}}$ in Eq. (23), the method then estimates \mathbf{A} by performing three more shift estimations using shifted versions by an integer translation of the second image. By denoting

$$\mathbf{A} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}, \quad (27)$$

and $\tilde{\mathbf{v}}_I = \text{findshift}(\tilde{I}_1(x, y), \tilde{I}_2(x, y))$, this yields an overdetermined system with three unknowns and six equations. For example, by assuming $\tilde{v}_x < 0$ and $\tilde{v}_y < 0$, the system is given by

$$\tilde{\mathbf{v}}_{00} = \tilde{\mathbf{v}}_I = \text{findshift}(\tilde{I}_1, \tilde{I}_2(x, y)) = \mathbf{M}\tilde{\mathbf{v}} \quad (28)$$

$$\tilde{\mathbf{v}}_{10} = \text{findshift}(\tilde{I}_1, I_2(x+1, y)) = \mathbf{M}(\tilde{\mathbf{v}} + [1 \ 0]^T)$$

$$\tilde{\mathbf{v}}_{01} = \text{findshift}(\tilde{I}_1, I_2(x, y+1)) = \mathbf{M}(\tilde{\mathbf{v}} + [0 \ 1]^T)$$

$$\tilde{\mathbf{v}}_{11} = \text{findshift}(\tilde{I}_1, I_2(x+1, y+1)) = \mathbf{M}(\tilde{\mathbf{v}} + [1 \ 1]^T), \quad (29)$$

where $\mathbf{M} = \tilde{\mathbf{A}}^{-1}\mathbf{A}$. Then subtracting (28) and pre-multiplying both sides by $\tilde{\mathbf{A}}$ gives

$$[p_1, q_1] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{10} - \tilde{\mathbf{v}}_{00}) \approx \mathbf{A}[1 \ 0]^T = [a \ b]^T \quad (30)$$

$$[p_2, q_2] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{01} - \tilde{\mathbf{v}}_{00}) \approx \mathbf{A}[0 \ 1]^T = [b \ c]^T \quad (31)$$

$$[p_3, q_3] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{11} - \tilde{\mathbf{v}}_{00}) \approx \mathbf{A}[1 \ 1]^T = [a + b \ b + c]^T. \quad (32)$$

Finally, the unbiased matrix \mathbf{A} is obtained by weighted least squares, minimizing the following functional

$$\epsilon = w_1(a - p_1)^2 + w_1(b - q_1)^2 + w_2(b - p_2)^2 + w_2(c - q_2)^2 + w_3(a + b - p_3)^2 + w_3(b + c - q_3)^2 \quad (33)$$

where the weights are chosen so that smaller shifts are given more importance

$$w_1 = \|\tilde{\mathbf{v}}_{10}\|^{-2}, w_2 = \|\tilde{\mathbf{v}}_{01}\|^{-2} \text{ and } w_3 = \|\tilde{\mathbf{v}}_{11}\|^{-2} \quad (34)$$

The justification for the weights is that GBSE center their Taylor development in zero, and therefore the smaller the shift, the more accurate it will be estimated. Finally, the weighted least squares system becomes the solution of

$$\begin{bmatrix} w_1 + w_3 & w_3 & 0 \\ w_3 & w_1 + w_2 + 2w_3 & w_3 \\ 0 & w_3 & w_2 + w_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} w_1 p_1 + w_3 p_3 \\ w_1 q_1 + w_2 p_2 + w_3(p_3 + q_3) \\ w_2 q_2 + w_3 q_3 \end{bmatrix}, \quad (35)$$

were a, b and c were the values of the unbiased matrix \mathbf{A} . As can be seen, functional (33) as well as the weights definitions came from the assumption that $\tilde{\mathbf{v}}_I < \mathbf{0}$. Therefore, the method begins by estimating $\tilde{\mathbf{v}}_I$, the shift between both original images \tilde{I}_1 and \tilde{I}_2 . Based on this result, four cases arise depending on the sign of both values of $\tilde{\mathbf{v}}_I$, each one yielding different equations which in turn define distinct weighted least squares systems to minimize as well as weight definitions (in the given example, $\tilde{\mathbf{v}}_I = \tilde{\mathbf{v}}_{00}$).

By assuring the subpixel condition of the underlying GBSE method, the other three cases, depending on the initial shift estimation $\tilde{\mathbf{v}}_I = [\tilde{v}_x, \tilde{v}_y]^T$, are

- If $\tilde{v}_x > 0$ and $\tilde{v}_y > 0$, then

$$\begin{aligned}
 \tilde{\mathbf{v}}_{00} &= \text{findshift}(\tilde{I}_1, \tilde{I}_2(x-1, y-1)) = \mathbf{M}(\tilde{\mathbf{v}} - [1 \ 1]^T), \\
 \tilde{\mathbf{v}}_{10} &= \text{findshift}(\tilde{I}_1, \tilde{I}_2(x-1, y)) = \mathbf{M}(\tilde{\mathbf{v}} + [-1 \ 0]^T), \\
 \tilde{\mathbf{v}}_{01} &= \text{findshift}(\tilde{I}_1, \tilde{I}_2(x, y-1)) = \mathbf{M}(\tilde{\mathbf{v}} + [0 \ -1]^T), \\
 \tilde{\mathbf{v}}_{11} &= \tilde{\mathbf{v}}_I = \text{findshift}(\tilde{I}_1, \tilde{I}_2(x, y)) = \mathbf{M}\tilde{\mathbf{v}}.
 \end{aligned} \tag{36}$$

- If $\tilde{v}_x > 0$ and $\tilde{v}_y < 0$, then

$$\begin{aligned}
 \tilde{\mathbf{v}}_{00} &= \text{findshift}(\tilde{I}_1, \tilde{I}_2(x-1, y)) = \mathbf{M}(\tilde{\mathbf{v}} + [-1 \ 0]^T), \\
 \tilde{\mathbf{v}}_{10} &= \tilde{\mathbf{v}}_I = \text{findshift}(\tilde{I}_1, \tilde{I}_2(x, y)) = \mathbf{M}\tilde{\mathbf{v}}, \\
 \tilde{\mathbf{v}}_{01} &= \text{findshift}(\tilde{I}_1, \tilde{I}_2(x-1, y+1)) = \mathbf{M}(\tilde{\mathbf{v}} + [-1 \ 1]^T), \\
 \tilde{\mathbf{v}}_{11} &= \text{findshift}(\tilde{I}_1, \tilde{I}_2(x, y+1)) = \mathbf{M}(\tilde{\mathbf{v}} + [0 \ 1]^T).
 \end{aligned} \tag{37}$$

- If $\tilde{v}_x < 0$ and $\tilde{v}_y > 0$, then

$$\begin{aligned}
 \tilde{\mathbf{v}}_{00} &= \text{findshift}(\tilde{I}_1, \tilde{I}_2(x, y-1)) = \mathbf{M}(\tilde{\mathbf{v}} + [0 \ -1]^T), \\
 \tilde{\mathbf{v}}_{10} &= \text{findshift}(\tilde{I}_1, \tilde{I}_2(x+1, y-1)) = \mathbf{M}(\tilde{\mathbf{v}} + [1 \ -1]^T), \\
 \tilde{\mathbf{v}}_{01} &= \tilde{\mathbf{v}}_I = \text{findshift}(\tilde{I}_1, \tilde{I}_2(x, y)) = \mathbf{M}\tilde{\mathbf{v}}, \\
 \tilde{\mathbf{v}}_{11} &= \text{findshift}(\tilde{I}_1, \tilde{I}_2(x+1, y)) = \mathbf{M}(\tilde{\mathbf{v}} + [1 \ 0]^T).
 \end{aligned} \tag{38}$$

Once we have the four shift estimates, given by either one of (29), (36), (37) or (38), a weighted least squares minimization scheme is used to compute a, b and c , the values of the unbiased matrix \mathbf{A} .

Following the same reasoning, the systems to solve for these three cases become

- If $\tilde{\mathbf{v}}_I = \tilde{\mathbf{v}}_{11}$ then

$$w_1 = \|\tilde{\mathbf{v}}_{00} + \tilde{\mathbf{v}}_{11}\|^{-2}, \quad w_2 = \|\tilde{\mathbf{v}}_{01} + \tilde{\mathbf{v}}_{11}\|^{-2}, \quad w_3 = \|\tilde{\mathbf{v}}_{10} + \tilde{\mathbf{v}}_{11}\|^{-2} \tag{39}$$

$$[p_1, q_1] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{00} - \tilde{\mathbf{v}}_{11}), \quad [p_2, q_2] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{01} - \tilde{\mathbf{v}}_{11}), \quad [p_3, q_3] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{10} - \tilde{\mathbf{v}}_{11}) \tag{40}$$

and the system to solve for a, b and c is

$$\begin{bmatrix} -w_1 - w_2 & -w_1 & 0 \\ -w_1 & -2w_1 - w_2 - w_3 & -w_1 \\ 0 & -w_1 & -w_1 - w_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} w_1 * p_1 + w_2 * p_2 \\ w_1 * (p_1 + q_1) + w_2 * q_2 + w_3 * p_3 \\ w_1 * q_1 + w_3 * q_3 \end{bmatrix}. \tag{41}$$

- If $\tilde{\mathbf{v}}_I = \tilde{\mathbf{v}}_{01}$ then

$$w_1 = \|\tilde{\mathbf{v}}_{00} + \tilde{\mathbf{v}}_{01}\|^{-2}, \quad w_2 = \|\tilde{\mathbf{v}}_{10} + \tilde{\mathbf{v}}_{01}\|^{-2}, \quad w_3 = \|\tilde{\mathbf{v}}_{11} + \tilde{\mathbf{v}}_{01}\|^{-2} \tag{42}$$

$$[p_1, q_1] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{00} - \tilde{\mathbf{v}}_{01}), \quad [p_2, q_2] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{10} - \tilde{\mathbf{v}}_{01}), \quad [p_3, q_3] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{11} - \tilde{\mathbf{v}}_{01}) \tag{43}$$

and the system to solve for a, b and c is

$$\begin{bmatrix} w_2 + w_3 & -w_2 & 0 \\ -w_2 & w_1 + 2w_2 + w_3 & -w_2 \\ 0 & w_2 & -w_1 - w_2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} w_2 * p_2 + w_3 * p_3 \\ -w_1 * p_1 + w_2 * (q_2 - p_2) + w_3 * q_3 \\ w_1 * q_1 + w_2 * q_2 \end{bmatrix}. \tag{44}$$

- If $\tilde{\mathbf{v}}_I = \tilde{\mathbf{v}}_{10}$ then

$$w_1 = \|\tilde{\mathbf{v}}_{00} + \tilde{\mathbf{v}}_{10}\|^{-2}, \quad w_2 = \|\tilde{\mathbf{v}}_{01} + \tilde{\mathbf{v}}_{10}\|^{-2}, \quad w_3 = \|\tilde{\mathbf{v}}_{11} + \tilde{\mathbf{v}}_{10}\|^{-2} \quad (45)$$

$$[p_1, q_1] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{00} - \tilde{\mathbf{v}}_{10}), \quad [p_2, q_2] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{01} - \tilde{\mathbf{v}}_{10}), \quad [p_3, q_3] = \tilde{\mathbf{A}}(\tilde{\mathbf{v}}_{11} - \tilde{\mathbf{v}}_{10}) \quad (46)$$

and the system to solve for a, b and c is

$$\begin{bmatrix} -w_1 - w_2 & w_2 & 0 \\ -w_2 & w_1 + 2w_2 + w_3 & -w_2 \\ 0 & -w_2 & w_2 + w_3 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} w_1 * p_1 + w_2 * p_2 \\ -w_1 * q_1 + w_2 * (p_2 - q_2) + w_3 * p_3 \\ w_2 * q_2 + w_3 * q_3 \end{bmatrix}. \quad (47)$$

Therefore, depending on the initial shift estimation result $\tilde{\mathbf{v}}_I$, one of the four possible cases is used to calculate the values of the unbiased derivative matrix \mathbf{A} . At last, the real unbiased shift is calculated as

$$\mathbf{v} = \mathbf{A}^{-1} \tilde{\mathbf{b}}. \quad (48)$$

2.6 Gradient computation and image prefiltering

By focusing on the design of the filters used to estimate the image gradient under noise, studied in conjunction with prefiltering the input images, Simoncelli [82] was able to reduce the bias by using a gradient filter that approximates the derivatives of the prefilter. The proposed pre-smoothing filters are forced to be symmetric, to guarantee they are linear phase filters, while the gradient filters are selected to be anti-symmetric in order to preserve the property of being a differentiator. Farid and Simoncelli [29] also proposed a set of both prefilters and differentiators obtained by minimizing the errors in the gradient direction, for a fixed size kernel.

This approach was later followed by Elad et al. [27] where they specifically studied the problem for GBSE. In their work, they noted that by designing a set of pre-smoothing filters and gradients filters minimizing the modelling error for each particular image, the estimator performance could be further improved. In fact, each filter is designed based on the spectral form of the first image and the *a priori* knowledge of a maximal motion. Their objective is then to find filter parameters for a set of filters by minimizing a cost function in order to reduce the error of the shift estimates. Although this solution achieves improved accuracy over previous approaches, it does in fact decrease the bias caused by the noise in an indirect way, completely ignoring the statistical performance of the GBSE. Furthermore, the minimization must be done for every image pair, thus radically augmenting the computational burden.

Followed by a precise study on the bias in [75], Robinson and Milanfar proposed designing a gradient filter based on the selection of pre-filters, on the prior knowledge of the image spectrum and some constraint about the shift [76]. Surprisingly, this work proposed to minimize the estimator bias by attacking the approximation error in the data model due to the linear signal approximation performed by the Taylor development, while completely ignoring the noise. In fact, low SNR conditions are discarded even though the bias due to the noise dominates the overall estimator bias [64]. For this reason, they achieve poor results on images with SNR lower than 20dB. Furthermore, none of these previous approaches work under aliased situations or badly sampled images, which is possible (yet undesired) on computer vision problems. To this end, Christmas [21] proposed a differentiator kernel that optimises the fit of the estimator with respect to the ideal differentiator $D(f) = i2\pi f$ at low frequencies, giving less importance to higher frequencies usually more affected by aliasing. This approach is suitable on cases where the image contents appear on the lower frequencies.

Note also that using a large kernel for image derivatives computation imply not only excessive blurring of the image, but also discarding more boundary pixels, thus leaving fewer equations to

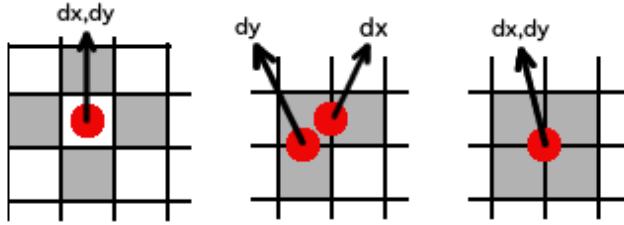


Figure 1: Used pixels (gray background) for fast gradient estimation methods and their exact localizations (red spots). **Left:** Centered differences. **Center:** 1D backward difference for both ∂x and ∂y . **Right:** 2D backward difference.

estimate the shift on (4), constraining the gradient kernel to be compact, precise and robust to the presence of noise. In fact, the impact on the accuracy and the robustness to noise of the gradient computation is a key factor to the final performance of the GBSE method. What is more, this computation must be fast in order to accelerate the algorithm.

Emulating the work of Simoncelli [82], in order to increase the accuracy of the method by minimizing noise or aliasing influence, we look for two kernel functions: an asymmetric kernel \mathbf{d} to estimate the image gradients and a symmetric kernel \mathbf{k} to prefilter the images. Using both kernels, matrix \mathbf{A} and vector \mathbf{b} from Eq. (4) become

$$\mathbf{A} = \begin{pmatrix} (\mathbf{d}_x * I_1)(p_1) & (\mathbf{d}_y * I_1)(p_1) \\ \vdots & \vdots \\ (\mathbf{d}_x * I_1)(p_n) & (\mathbf{d}_y * I_1)(p_n) \end{pmatrix} \text{ and } \mathbf{b} = - \begin{pmatrix} (\mathbf{k} * (I_1 - I_2))(p_1) \\ \vdots \\ (\mathbf{k} * (I_1 - I_2))(p_n) \end{pmatrix}, \quad (49)$$

where $*$ denotes convolution.

Because the method should be computationally fast, a straightforward candidate for gradient estimation is the well-known centered differences method using a $[1, 0, -1]$ kernel, however since the central pixel is ignored in the computation, its performance is usually poor under high precision constraints. For this reason, a backward difference method using a $[1, -1]$ kernel would seem more appropriate, however, this derivative corresponds to the center between both pixels and not in the pixels itself, as seen from the middle image of Fig. 1.

A more exact gradient estimation method is shown on the right of Fig. 1. Using the original image I_1 , the derivatives are calculated by convolving it with \mathbf{d}_x and \mathbf{d}_y given by

$$\mathbf{d}_x = \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & -1/2 \end{bmatrix}, \quad \mathbf{d}_y = \begin{bmatrix} 1/2 & 1/2 \\ -1/2 & -1/2 \end{bmatrix} \quad (50)$$

and resampled versions of both images I_1 and I_2 are used to calculate vector \mathbf{b} of (3), shifting them by half a pixel to the bottom right using bilinear interpolation, finally taking

$$\mathbf{k} = \begin{bmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{bmatrix}. \quad (51)$$

This gradient estimation trick, which we shall call the *hypemode*, despite being simplistic usually improves the accuracy obtained by GBSE methods using finite difference gradient estimation. This is because it slightly blurs the input images, which alleviates both aliasing and noise, and because of its accurate gradient localization.

Another considered smoothing kernel is the 2D Gaussian kernel given by sampling from

$$\mathbf{k} = g(x, y, \sigma_g) = \frac{1}{2\pi\sigma_g^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_g^2}\right) \quad (52)$$

and its derivatives

$$\mathbf{d}_x = \frac{\partial g(x, y, \sigma_g)}{\partial x} = -\frac{x}{2\pi\sigma_g^4} \exp\left(-\frac{x^2 + y^2}{2\sigma_g^2}\right), \quad (53)$$

$$\mathbf{d}_y = \frac{\partial g(x, y, \sigma_g)}{\partial y} = -\frac{y}{2\pi\sigma_g^4} \exp\left(-\frac{x^2 + y^2}{2\sigma_g^2}\right). \quad (54)$$

The kernel support is determined by the σ_g value, which defines the amount of blur applied to each image before performing the computations, therefore higher noise values imply a higher σ_g . While a too low value would be less tolerant to noise, a too high value would imply losing potentially valuable textures which usually aid the shift estimation method. In our experiments, we evaluated using $\sigma_g = \{0.3, 0.6, 1\}$ leading to supports 3, 5 and 7 respectively.

Other evaluated image gradient estimation methods in the context of GBSE were the 3×3 and the 5×5 kernels from Simoncelli [82], and the 3×3 , 5×5 and 7×7 from Farid [29]. As mentioned above, these approaches are the result of finding a smoothing prefilter, which is assumed to be a separable product of identical symmetric 1D functions, together with a derivative antisymmetric filter. Simoncelli minimizes the following energy

$$E(P, D) = \int d\omega W^2(\omega) [j\omega P(\omega) - D(\omega)]^2 \quad (55)$$

where $P(\omega)$ and $D(\omega)$ are the Fourier transform of a prefilter and a derivative filter respectively. The weights W are taken to mimic the expected spectral content of natural images

$$W(\omega) = \frac{1}{\sqrt{|\omega|}}. \quad (56)$$

Therefore, the method fixes a kernel size and minimizes Eq. (55) by defining some constraints on both kernels. On the other side, Farid minimizes

$$E(P, D) = \frac{\int_{\omega} |j\omega P(\omega) - D(\omega)|^2}{\int_{\omega} P^2(\omega)} \quad (57)$$

in its discrete form

$$E(\mathbf{p}, \mathbf{d}) = \frac{|j\omega F_s \mathbf{p} - F_a \mathbf{d}|^2}{|F_s \mathbf{p}|^2}, \quad (58)$$

where \mathbf{p} is the prefilter vector of length $(L + 1)/2$, \mathbf{d} the derivative filter vector of size $(L - 1)/2$, F_s and F_a are matrices whose columns contain the real and imaginary components of the discrete Fourier basis of size $K \gg L$ such that $F_s \mathbf{p}$ gives the DFT of the prefilter and $F_a \mathbf{d}$ gives the DFT of the antisymmetric derivative filter.

The gradient estimators from Christmas [21] were also included in the evaluation. As mentioned before, they are obtained by fitting the kernel with the ideal differentiator $D(f) = i2\pi f$ at low frequencies, avoiding high frequency information. Specifically, for an n th order estimator, they equate the first n derivatives of the estimator frequency response to those of the ideal differentiator. Note that Christmas' approach does not require prefiltering the images, yielding faster processing times, although usually coming with a decrease in accuracy.

All evaluated gradient estimators kernels are shown in table 1. Given image I , the gradient along the horizontal direction is estimated by convolving each column of I with the prefilter vector \mathbf{k} and then convolving each row of the result with the differentiator vector \mathbf{d} . The vertical gradient is obtained by swapping \mathbf{d} and \mathbf{k} yielding

$$I_x = \mathbf{d} * \mathbf{k}^T * I, \quad (59)$$

$$I_y = \mathbf{k} * \mathbf{d}^T * I. \quad (60)$$

Kernel		Sample Number						
		-3	-2	-1	0	1	2	3
Hypemode 2×2	k				0.5	0.5		
	d				1	-1		
Gaussian $\sigma = 0.3$	k			0.003865	0.999990	0.003865		
	d			0.707110	0.000000	-0.707110		
Gaussian $\sigma = 0.6$	k		0.003645	0.235160	0.943070	0.235160	0.003645	
	d		0.021915	0.706770	0.000000	-0.706770	-0.021915	
Gaussian $\sigma = 1$	k	0.008343	0.101650	0.455560	0.751090	0.455560	0.101650	0.008343
	d	0.035436	0.287800	0.644920	0.000000	-0.644920	-0.287800	-0.035436
Simoncelli 3×3	k			0.224209	0.551580	0.224209		
	d			0.455271	0.000000	-0.455271		
Simoncelli 5×5	k		0.035697	0.248874	0.430855	0.248874	0.035697	
	d		0.107662	0.282671	0.000000	-0.282671	-0.107662	
Farid 3×3	k			0.229879	0.540242	0.229879		
	d			0.425287	0.000000	-0.425287		
Farid 5×5	k		0.037659	0.249153	0.426375	0.249153	0.037659	
	d		0.109604	0.276691	0.000000	-0.276691	-0.109604	
Farid 7×7	k	0.004711	0.069321	0.245410	0.361117	0.245410	0.069321	0.004711
	d	0.018708	0.125376	0.193091	0.000000	-0.193091	-0.125376	-0.018708
Christmas 3×3	k			1	1			
	d			0	0	-1		
Christmas 5×5	k		-1/12	2/3	1			
	d				0	-2/3	1/12	
Christmas 7×7	k	1/60	-3/20	3/4	1			
	d				0	-3/4	3/20	-1/60

Table 1: Gradient estimation kernels evaluated for GBSE methods.

2.7 Interpolation methods for image resampling

In order to iterate the algorithm, resampling must be done to shift the image, as indicated in the step 6 of Algorithm 1. To this end, five different interpolation methods were evaluated, namely bilinear, bicubic [47] and cubic spline interpolation [23], together with resampling using the Fourier shift theorem [16], which is evaluated with and without image periodization. Image periodization avoids the generation of ringing artifacts due to the discontinuities on the image borders by generating an augmented version mirroring the image. This resulting image has no discontinuities when periodization is assumed, as depicted in Fig. 2.

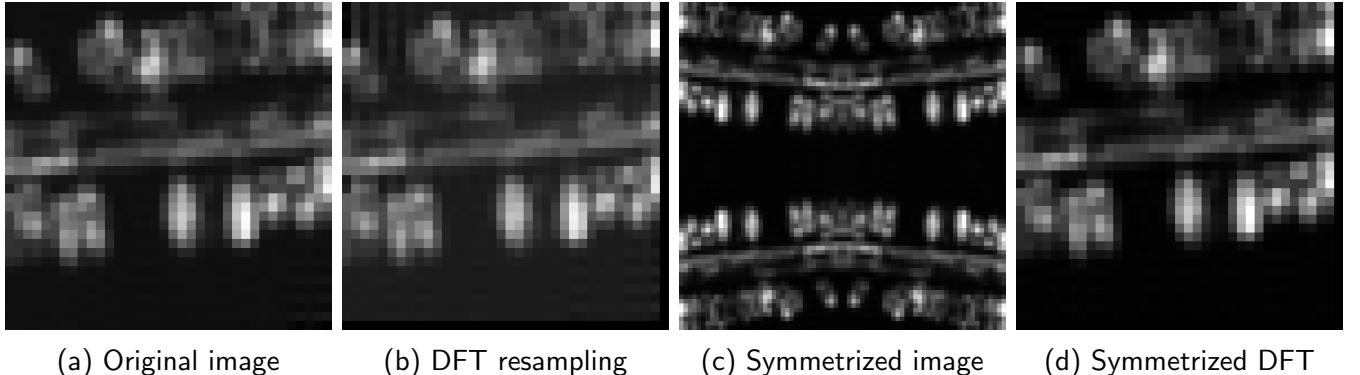


Figure 2: Example of FFT resampling with and without image symmetrization. Direct resampling with DFT produces ringing due to the discontinuities at the periodized boundaries. No visible ringing is observed after resampling with symmetrization.

The impact on the results by selecting a correct interpolation method could be important, as it will be shown in the results. The more precise the interpolation, the more accurate the results, however it comes with an increase in the processing cost. A report of interpolation methods for fast image resampling due to a global displacement will be presented in Appendix 1.

2.8 Shift estimation by centroid of interpolating kernel

In a recent work by Gilman and Leist [35], the authors propose an illumination-invariant method for fast image registration, for the case of additive contrast changes. Their approach tries to overcome the negative aspects of iterative gradient-based methods by proposing a fast non-iterative solution that does not require performing gradient estimation. Conventionally, iterative GBSE methods perform the following minimization

$$\operatorname{argmin}_{v_x, v_y} \sum_{x, y} (I_2(x, y) - I_1(x + v_x, y + v_y))^2 \quad (61)$$

by approximating $I_1(x + v_x, y + v_y)$ by its first order Taylor development. This is followed by an iterative scheme in which the second image is resampled using the result from the shift estimation. By doing this, an interpolation method such as Eq. (62) is required. The interpolation kernel h is based on some interpolation basis function with rectangular support between (M^-, N^-) and (M^+, N^+) and its coefficients are samples of the basis function offset by the estimated shift (v_x, v_y) . The interpolation is given by the following expression

$$I_1(x + v_x, y + v_y) = \sum_{m=M^-}^{M^+} \sum_{n=N^-}^{N^+} I_1(x + m, y + n) h_{\hat{v}_x, \hat{v}_y}(m, n). \quad (62)$$

By combining (61) with (62), the authors propose to perform the minimization with respect to the interpolating kernel

$$\operatorname{argmin}_{\mathbf{h}} \sum_{x,y} \left(I_2(x,y) - \sum_{m=M^-}^{M^+} \sum_{n=N^-}^{N^+} I_1(x+m, y+n) h(m, n) \right)^2 \quad (63)$$

yielding a linear system $\mathbf{Av} = \mathbf{b}$ where, provided the interpolation kernel is of size $W_k \times H_k$, its main computational cost involves inverting a $W_k^2 \times H_k^2$ matrix. Its implementation is accelerated by simply pre-computing the shifted versions of I_1 and then performing the Hadamard product.

Finally the shift estimates can be obtained by computing

$$\hat{u}_x = \sum_m \sum_n m h(m, n), \quad \hat{u}_y = \sum_m \sum_n n h(m, n). \quad (64)$$

However, the method could easily be made robust against illumination changes. Let A and S be the gain and the offset of a global illumination change given by $I_2(x, y) = AI_1(x, y) + S$, then by normalizing its computation

$$\hat{u}_x = \frac{\sum_m \sum_n m h(m, n)}{\sum_m \sum_n h(m, n)}, \quad \hat{u}_y = \frac{\sum_m \sum_n n h(m, n)}{\sum_m \sum_n h(m, n)}, \quad (65)$$

the method becomes robust against gain A and therefore contrast invariant. The authors also suggest adding the offset variable S into the optimization in Eq. (63) to add illumination invariance as well.

3 Phase-correlation methods

Introduced in 1975 by Kuglin and Hines [50], the phase correlation method aligns images based on their normalized cross-power spectrum, which is equivalent to computing the normalized circular cross-correlation in the spatial domain. Let I_1 and I_2 be $M \times N$ images such that

$$I_2(x, y) = I_1((x - v_x) \bmod M, (y - v_y) \bmod N) + n(x, y), \quad (66)$$

where \bmod is the modulo operator implying a circular shift and $n(x, y)$ denotes the effect of interference terms such as noise, non-overlapping regions, etc. Let F_1 and F_2 be the discrete 2D Fourier transforms of I_1 and I_2 respectively, then by ignoring this last term the Fourier Shift Theorem indicates that

$$F_2(\omega_x, \omega_y) = F_1(\omega_x, \omega_y) \exp \left(-i2\pi \left(\frac{\omega_x v_x}{M} + \frac{\omega_y v_y}{N} \right) \right) \quad (67)$$

and by the cross-correlation theorem we have that

$$C = I_1 \star I_2 = I_1^*(-t) \otimes I_2(t) = \mathcal{F}^{-1}\{F_1^* F_2\}, \quad (68)$$

where \star and \otimes denote the cross-correlation and the convolution operator respectively, $*$ denotes the complex conjugate and \mathcal{F}^{-1} stands for the discrete inverse Fourier transform. Then by normalizing in Fourier the cross-correlation and based on the Fourier Shift Theorem, the phase correlation matrix, defined by the normalized cross-power spectrum C , is given by

$$\begin{aligned} C(\omega_x, \omega_y) &= \frac{F_2(\omega_x, \omega_y) F_1^*(\omega_x, \omega_y)}{F_1(\omega_x, \omega_y) F_1^*(\omega_x, \omega_y)} \\ &= \exp \left(-i2\pi \left(\frac{\omega_x v_x}{M} + \frac{\omega_y v_y}{N} \right) \right). \end{aligned} \quad (69)$$

If both F_1 and F_2 were continuous, then by applying the inverse Fourier transform, we obtain the phase-only correlation (POC) [87] or phase correlation surface (PCS) [73] given by

$$c(x, y) = \mathcal{F}^{-1} \left\{ C(\omega_x, \omega_y) \right\} (x, y) = \delta(x - v_x, y - v_y), \quad (70)$$

where $\delta(x - v_x, y - v_y)$ is a Dirac function centered at (v_x, v_y) . Note that by performing normalization, this method becomes robust to affine intensity changes, i.e.

$$I_1(x - v_x, y - v_y) = a \cdot I_2(x, y) + b, \quad a, b \in \mathbf{R}. \quad (71)$$

Also note that this method is based on the Fourier shift theorem, which holds when the shift between both images is circular, that is, the part of the image that disappears on one side, appears on its opposite side. Finally, the peak of c is searched to obtain the translation between both images:

$$(\hat{v}_x, \hat{v}_y) = \arg \max_{(x, y)} c(x, y). \quad (72)$$

Unless the shift can be exactly described as in Eq. (66), due to reasons such as sub-pixel displacements, aliasing, image noise or non-overlapped regions, $c(x, y)$ is not an exact Dirac function, although the location of the peak still permits to accurately compute the displacement. Indeed, the peak value is often lower than one pixel and the surface is frequently noisy, as seen from figures 3 and 4. Sub-pixel shifts, for example, imply that the energy is distributed between the peak and its adjacent neighbors. Commonly, function fitting methods are used to estimate the peak location to non-integer values. Some other methods try to estimate the shift directly in the Fourier domain, eliminating the need to compute another FFT and thus, decreasing the complexity of the algorithm.

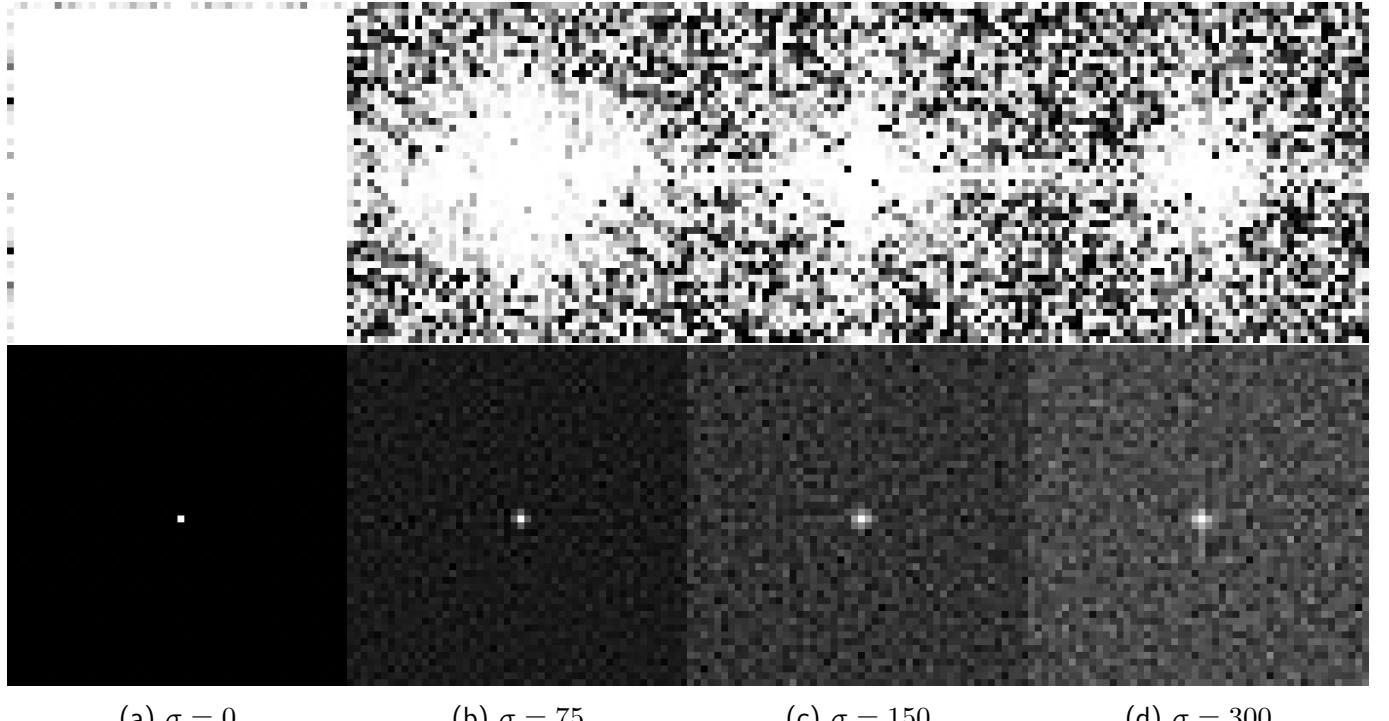


Figure 3: Cross-power spectrum $C(\omega_x, \omega_y)$ and phase correlation surfaces $c(x, y)$ of two identical images under different random WGN. **Top:** Cross power spectrum. **Bottom:** phase correlation surfaces. Dynamic ranges extended for visualization purposes.

While in this article we will describe only the most relevant approaches, complementary information could be found on review articles [71, 3]. Extensions of this technique to estimate scale and rotation transformations also exist [70], although they will not be covered in this review.

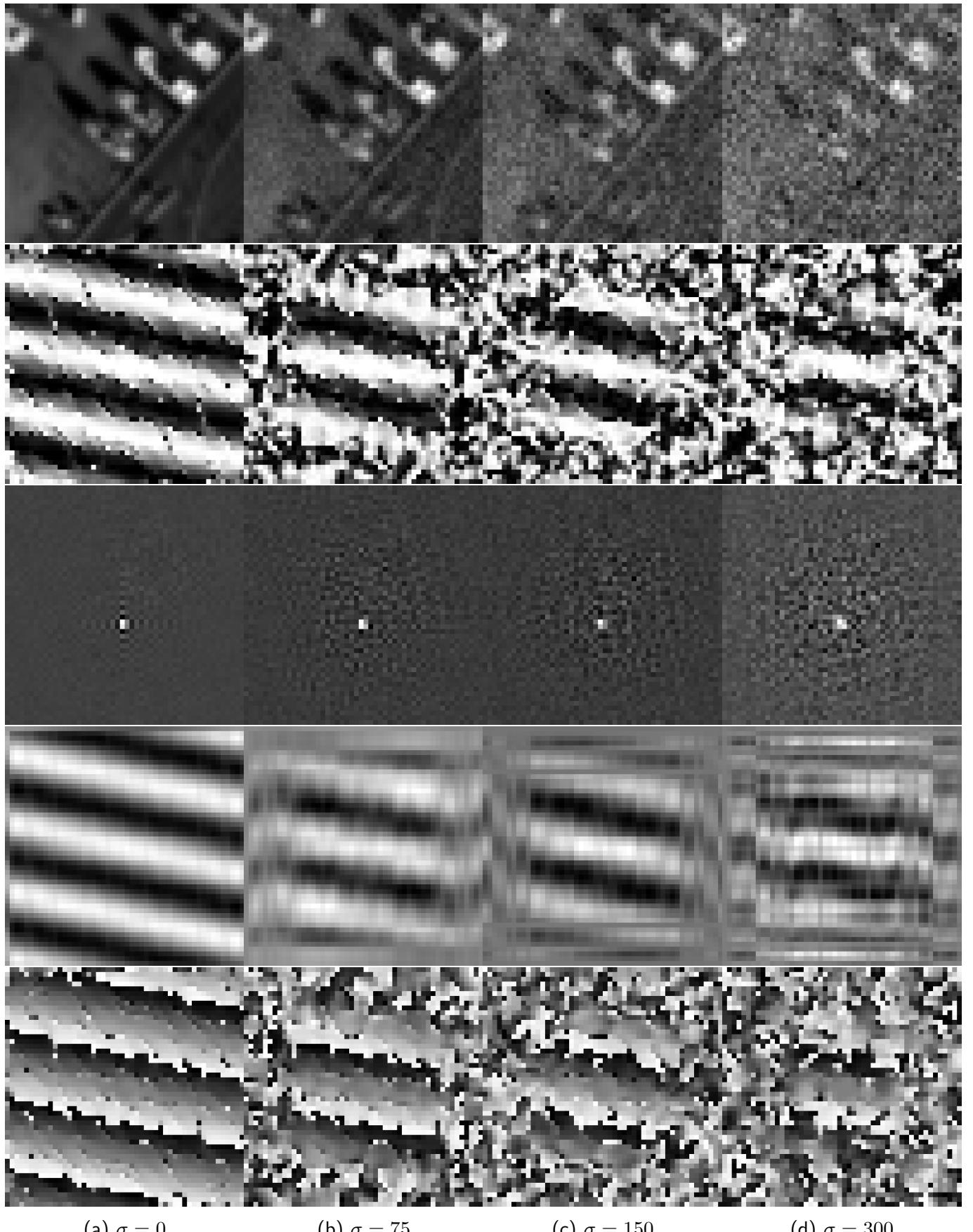


Figure 4: Example of different surfaces used to estimate the displacement of two images shifted by $(-3.5, 0.75)$ pixels under different random WGN ($\sigma = 0, 75, 150$ and 300) assuming 12-bit images. **First row:** First image. **Second row:** Real component of cross-power spectrum $C(\omega_x, \omega_y)$. **Third row:** Phase correlation surfaces $c(x, y)$. **Fourth row:** real component rank-1 approximation of the CPS [43]. **Last row:** Phase difference matrix [86]. Dynamic ranges extended for visualization purposes.

3.1 Local function fitting in the spatial domain

Fitting well-known functions in the vicinity of the integer maximum peak of the phase correlation surface can be used to obtain accurate subpixel precision while being computationally efficient.

Quadratic fitting. Abdou [1], working on registration of video frames, proposed to fit a 1D quadratic (parabolic) function on each separate coordinate on the vicinity of the main peak. Let (x_m, y_m) be the integer position of the peak (the integer solution of Eq. (72)), then the values used for the fitting on the horizontal and vertical directions are

$$\text{Horizontal: } \{c(x_m - 1, y_m), c(x_m, y_m), c(x_m + 1, y_m)\}, \quad (73)$$

$$\text{Vertical: } \{c(x_m, y_m - 1), c(x_m, y_m), c(x_m, y_m + 1)\}. \quad (74)$$

Then fitting a parabolic function yields a closed-form solution

$$\hat{v}_x = \frac{c(x_m + 1, y_m) - c(x_m - 1, y_m)}{2 * c(x_m, y_m) - c(x_m + 1, y_m) - c(x_m - 1, y_m)}. \quad (75)$$

Gaussian fitting. In [1] the author also proposed fitting a Gaussian function to the same data, thus yielding the closed form formula

$$\hat{v}_x = \frac{\log \left| \frac{c(x_m + 1, y_m)}{c(x_m - 1, y_m)} \right|}{\log \left| \frac{c(x_m, y_m)^2}{c(x_m + 1, y_m) \cdot c(x_m - 1, y_m)} \right|}. \quad (76)$$

The vertical shifts are obtained by using the samples from (74) instead of (73).

Sinc fitting. Foroosh et al. [78, 32] propose to understand the subpixel shift of the input images as if it has been obtained by an integer shift on a higher resolution grid followed by subsampling. Based on this assumption, they analytically determine that the shape of the phase correlation surface corresponds to a Dirichlet kernel. Indeed, the cross-power spectrum of the downsampled frames by factors of M and N along both horizontal and vertical axes is given by

$$C(u, v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \frac{F(\frac{u+2\pi m}{M}, \frac{v+2\pi n}{N})}{\sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} F(\frac{u+2\pi m'}{M}, \frac{v+2\pi n'}{N})} \exp \left(-i \left(\frac{u+2\pi m}{M} v_x, \frac{v+2\pi n}{N} v_y \right) \right) \quad (77)$$

and its inverse DFT is

$$\mathcal{F}^{-1} \left\{ C(u, v) \right\} = c(x, y) = \frac{1}{WH} \frac{\sin(\pi(Mx - v_x))}{\sin(\pi(Mx - v_x)/W)} \frac{\sin(\pi(Ny - v_y))}{\sin(\pi(Ny - v_y)/H)}, \quad (78)$$

where W and H are the width and height respectively before downsampling. Based on this study, the authors propose to approximate this Dirichlet kernel by a sinc function

$$\text{sinc}(x) = \frac{\sin \pi x}{\pi x} \quad (79)$$

yielding

$$c(x, y) \approx \frac{\sin(\pi(Mx - v_x))}{\pi(Mx - v_x)} \frac{\sin(\pi(Ny - v_y))}{\pi(Ny - v_y)} + n(x, y), \quad (80)$$

where $n(x, y)$ refers to interference terms such as noise, aliasing, non-overlapped regions, possible compression artifacts, etc.

Therefore, to rapidly recover the shift, based on the fact that for subpixel displacements the signal power in the phase correlation is usually concentrated in a main peak (x_m, y_m) and two side-peaks at $(x_m, y_m \pm 1)$ and $(x_m \pm 1, y_m)$, the authors calculate the shift by linearly weighting the main peak together with one of the side-peaks using only two values to estimate the shift on each dimension. For example, by assuming the side-peaks to be $(x_m + 1, y_m)$ and $(x_m, y_m + 1)$ and neglecting the interference component $n(x, y)$, this yields

$$\hat{v}_x = \frac{c(x_m + 1, y_m)}{c(x_m + 1, y_m) \pm c(x_m, y_m)}, \quad \hat{v}_y = \frac{c(x_m, y_m + 1)}{c(x_m + 1, y_m) \pm c(x_m, y_m)}, \quad (81)$$

where both \hat{v}_x and \hat{v}_y are chosen so that they both lie between $[x_m - 1, x_m + 1]$ and $[y_m - 1, y_m + 1]$ respectively. Another possibility is to do least squares minimization to fit the 3×3 grid centered on $c(x_m, y_m)$ on each dimension with a 1D sinc function:

$$\hat{v}_x = \operatorname{argmin}_{x_i \in \{x_m - 1, x_m, x_m + 1\}} \sum [c(x_i, y_m) - \operatorname{sinc}(x_i - C)]^2. \quad (82)$$

However, due to the non-linearity of the sinc, a non-linear least squares minimization method has to be used. Since this minimization requires only to fit a function over just three values, it does not require much computational time. However, this solution ignores the interference term, thus yielding poor results under low SNR scenarios. Indeed, computing \hat{v}_x of Eq. (81) by considering $n(x, y)$ in Eq. (80) and taking the limit when $v_x \rightarrow 0$ we obtain

$$\lim_{v_x \rightarrow 0} \hat{v}_x = \frac{n(1, 0)}{(\pi v_y)^{-1} \sin(\pi v_y) + n(1, 0) + n(0, 0)} \neq 0, \quad (83)$$

which proves that the estimator is biased under low SNR situations.

eSinc fitting. More recently, Argyriou and Vlachos [7] proposed a modification of the Sinc function by applying exponential weighting, defined as follows

$$\operatorname{esinc}(x) = \exp(-x^2) \frac{\sin \pi x}{\pi x}. \quad (84)$$

They claim this modified sinc function is able to better approximate phase correlation surfaces obtained from video data. The esinc is later parametrized to adapt to different magnitudes, scales and shifts changes, searching the parameters of $A \operatorname{esinc}(B(x - C))$, which leads to the following minimization problem

$$(A, B, C) = \operatorname{argmin}_{x_i \in \{x_m - 1, x_m, x_m + 1\}} \sum [c(x_i, y_m) - A \operatorname{esinc}(B(x_i - C))]^2, \quad (85)$$

that is again solved using a non-linear optimization method. In this case, C stands for the horizontal shift estimation \hat{v}_x ; \hat{v}_y is obtained in a similar way, by sampling over the vertical axis. Note that this minimization scheme could also be applied to the sinc fitting case, so therefore will be included in the evaluation.

Using the difference of both side-peaks. Ren et al. [72] claimed that using the difference between both side-peaks of $c(x_m, y_m)$ instead of a single side-peak as in [32] reduces the bias caused by ignoring the interference term as shown in Eq. (83). Let $D_x = c(x_m + 1, y_m) - c(x_m - 1, y_m)$ and $D_y = c(x_m, y_m + 1) - c(x_m, y_m - 1)$, their method then reduces to calculating

$$\hat{v}_x = \frac{D_x}{c(x_m, y_m) + |D_x|} = \frac{\operatorname{sign}(D_x)}{1 + c(x_m, y_m)/|D_x|}, \quad \hat{v}_y = \frac{D_y}{c(x_m, y_m) + |D_y|} = \frac{\operatorname{sign}(D_y)}{1 + c(x_m, y_m)/|D_y|}. \quad (86)$$

Least squares fitting of modified cross-power spectrum Takita *et al.* [87] drastically improved the accuracy by proposing three important modifications to the method. First, they suggest using least squares fitting involving the frequencies of the neighbourhood around the peak (x_m, y_m) up to 9×9 pixels. Second, a 2D Hanning window should be applied to the images before computing their cross-power spectrum to avoid false edge effects. Finally, the authors propose to filter the cross-power spectrum using low-pass-type filters in order to reduce the influence of corrupted high frequency components. Since this implies modifying the peak shape, they propose a specific fitting model for each low-pass filter. The four filters are

$$H_1(\omega_x, \omega_y) = \begin{cases} 1 & |\omega_x| \leq U_1, |\omega_y| \leq U_2, \\ 0 & \text{otherwise,} \end{cases} \quad (87)$$

$$H_2(\omega_x, \omega_y) = \frac{1}{MN} H_1(\omega_x, \omega_y) \otimes H_1(\omega_x, \omega_y), \quad (88)$$

$$H_3(\omega_x, \omega_y) = \frac{1}{M^2 N^2} H_2(\omega_x, \omega_y) \otimes H_1(\omega_x, \omega_y), \quad (89)$$

$$H_4(\omega_x, \omega_y) = \exp \{2\pi^2 \sigma^2 (\omega_x^2 + \omega_y^2)\}, \quad (90)$$

and the models to fit each of them are given by

$$r_1(x, y) = \frac{\alpha}{MN} \frac{\sin \left(\frac{V_1}{M} \pi (x + v_x) \right)}{\sin \left(\frac{\pi}{M} (x + v_x) \right)} \frac{\sin \left(\frac{V_2}{N} \pi (y + v_y) \right)}{\sin \left(\frac{\pi}{N} (y + v_y) \right)}, \quad (91)$$

$$r_2(x, y) = \left\{ \frac{\alpha}{MN} \frac{\sin \left(\frac{V_1}{M} \pi (x + v_x) \right)}{\sin \left(\frac{\pi}{M} (x + v_x) \right)} \frac{\sin \left(\frac{V_2}{N} \pi (y + v_y) \right)}{\sin \left(\frac{\pi}{N} (y + v_y) \right)} \right\}^2, \quad (92)$$

$$r_3(x, y) = \left\{ \frac{\alpha}{MN} \frac{\sin \left(\frac{V_1}{M} \pi (x + v_x) \right)}{\sin \left(\frac{\pi}{M} (x + v_x) \right)} \frac{\sin \left(\frac{V_2}{N} \pi (y + v_y) \right)}{\sin \left(\frac{\pi}{N} (y + v_y) \right)} \right\}^3, \quad (93)$$

$$r_4(x, y) = \frac{1}{2\pi\sigma^2} \exp \left\{ -\frac{x^2 + y^2}{2\sigma^2} \right\}, \quad (94)$$

where α is a cut-off frequency parameter, $V_1 = 2U_1 + 1$, $V_2 = 2U_2 + 1$ and U_1 , U_2 and σ controls the pass-band width.

Increasing accuracy by zero padding A simple strategy to increase phase correlation accuracy is to upsample the cross-correlation grid by zero-padding the cross-power spectrum $C(\omega_x, \omega_y)$. Obviously, this implies increasing the algorithm cost since the inverse DFT has to be computed for a much larger image. However, once an initial estimate of the correlation peak is found, Guizar-Sicairos [39] proposed to refine this value by only computing the DFT values on a small neighborhood around the peak, achieving more precision without using too many computational resources. In fact, the authors propose the use a matrix multiplication implementation of the 2D DFT [84], which performs the upsampling of a 1.5×1.5 pixel neighborhood around the initial estimate without the need to zero-pad the cross-correlation.

Local center of mass on the phase correlation surface Another fast strategy to achieve subpixel accuracy is to compute the center of mass of the phase correlation surface [3, 51]. The center of mass could be computed in 1D using

$$\hat{v}_x = \frac{\sum_{x=x_m-w}^{x_m+w} x \cdot c(x, y_m)}{\sum_{x=x_m-w}^{x_m+w} c(x, y_m)}, \quad \hat{v}_y = \frac{\sum_{y=y_m-w}^{y_m+w} y \cdot c(x_m, y)}{\sum_{y=y_m-w}^{y_m+w} c(x_m, y)}, \quad (95)$$

where $w \in [1, 2, 3]$ is the support size, or using a 2D approach

$$\hat{v}_x = \frac{\sum_{y=y_m-w}^{y_m+w} \sum_{x=x_m-w}^{x_m+w} x \cdot c(x, y)}{\sum_{y=y_m-w}^{y_m+w} \sum_{x=x_m-w}^{x_m+w} c(x, y)}, \quad \hat{v}_y = \frac{\sum_{y=y_m-w}^{y_m+w} y \cdot c(x, y)}{\sum_{y=y_m-w}^{y_m+w} \sum_{x=x_m-w}^{x_m+w} c(x, y)}. \quad (96)$$

3.2 Analyzing the phase difference matrix in the frequency domain

Instead of computing the inverse Fourier transform in Eq. (70), several methods try to compute the shift directly in the Fourier domain [86, 12]. These methods work by computing the phase of the cross power spectrum

$$\phi(\omega_x, \omega_y) = \arg(C(\omega_x, \omega_y)) = \frac{\omega_x v_x}{M} + \frac{\omega_y v_y}{N}, \quad (97)$$

where the shift (v_x, v_y) can be easily recovered by fitting a plane passing through the origin. The matrix $\phi(\omega_x, \omega_y)$ is usually referred to as the phase difference matrix [12], displayed in the last row of Fig. 4.

Robust plane fitting Once ϕ is computed, several approaches recover the shift by a linear least squares plane regression method [86], [49] and [81]. However, due to noise and aliasing on the input images, these methods discard corrupted frequencies in the minimization, yielding the following least squares minimization

$$\hat{v}_x, \hat{v}_y = \underset{v_x, v_y}{\operatorname{argmin}} \sum_{\omega_x, \omega_y} M(\omega_x, \omega_y) \left[2\pi \left(\frac{\omega_x v_x}{M} + \frac{\omega_y v_y}{N} \right) - \phi(\omega_x, \omega_y) \right]^2, \quad (98)$$

where $M(\omega_x, \omega_y)$ is the binary mask used to exclude contaminated spectral components from the minimization.

Knutsson *et al.* [49] emphasizes on speed, so its method uses only two or four frequencies of the whole spectra, yielding an extremely fast method that avoids the computation of the DFTs:

$$M(\omega_x, \omega_y) = \begin{cases} 1 & \text{if } (\omega_x, \omega_y) = (1, 0) \vee (\omega_x, \omega_y) = (0, 1), \\ 0 & \text{otherwise.} \end{cases} \quad (99)$$

Stone *et al.* [86] analytically studied the influence of aliasing on the phase difference matrix and proposed an algorithm that focuses on reducing its influence on the final accuracy. To this end, their approach first applies a window in the spatial domain in order to eliminate image-boundary effects in the frequency domain. In particular, the authors suggest using either Blackman or Blackman-Harris windows [83]. Finally, the minimization in (98) is performed by masking spectral components mostly contaminated by aliasing, namely:

$$M(\omega_x, \omega_y) = \begin{cases} 0 & \text{if } \sqrt{\omega_x^2 + \omega_y^2} > 0.3 \cdot N \vee |F_1(\omega_x, \omega_y)| < \alpha \vee |F_2(\omega_x, \omega_y)| < \alpha, \\ 1 & \text{otherwise.} \end{cases} \quad (100)$$

Indeed, they exclude frequencies further away than $0.3 \cdot N$ from the center, where N is the minimum of the number of samples in both dimensions, and frequencies where the energy is below a specified threshold α in any of both images. In the original article, α was chosen by sorting the frequencies by magnitude and retaining the K largest ones. Furthermore, K was set by performing several shift estimations over a range of values and keeping the one where the shift remained stable. In addition to the fact that this procedure is computationally expensive, the range where the shift remained stable was not always straightforward to detect in our empirical testing. A greedy technique that improved over the original method was to discard the frequencies where the magnitudes of both

images were lower than a specified percentile. In particular, we observed that taking the percentile 60 systematically improved over the original approach.

Sidick *et al.* [81] use this same approach in the adaptive cross-correlation (ACC) method to perform shift estimation. However, they perform it in an iterative scheme where they first estimate the shift between centered windows half the size of the images followed by resampling the second image in the frequency domain accumulating the estimated shift. The same authors later proposed adaptive periodic correlation (APC) [79] where they changed this shift estimation method for a more robust periodic correlation approach [67].

Sawtooth cycle count Robust plane fitting approaches estimate the slope of the plane passing through the origin of the phase difference matrix. Not only these methods decay considerably with the noise, but they also suffer from the phase wrapping problem [34]. For this reason, both approaches are limited to estimating displacements up to one pixel, for which the phase will not wrap.

Balci and Fooresh [12] made the observation that the phase difference matrix $\phi(\omega_x, \omega_y)$ defined in Eq. (97) is a 2D sawtooth signal (see last row of Fig. 4) and that the subpixel displacement between both images can be determined by counting the number of cycles along each axis. Indeed, if no phase unwrapping is performed, these cycles are attributed to the wrapping behaviour of the phase. To compute the number of cycles, they propose to count the peaks in the Hough-transform domain [44]. Each of these peaks provides one linear constraint on both v_x and v_y , yielding an over-determined system which is solved by imposing a regularity constraint where the regularization parameter is obtained using the Generalized Cross Validation (GCV) method.

3.3 Subspace phase correlation methods

Projection-based subspace methods Alliney and Morandi [4] showed that the complexity of 2D phase correlation could be significantly reduced by performing two 1D phase correlation estimations on the image projections in each dimension. Indeed, let $F(\omega_x, \omega_y)$ be the Fourier transform of $I(x, y)$, then by defining

$$I_x(x) = \sum_y I(x, y), \quad I_y(y) = \sum_x I(x, y) \quad (101)$$

and computing the Fourier transform $F_x(\omega_x)$ of $I_x(x)$ we have that

$$F_x(\omega_x) = \mathcal{F} \left\{ \sum_y I(x, y) \right\} = \sum_x \sum_y I(x, y) e^{-i2\pi\omega_x x/M}, \quad (102)$$

$$= \sum_x \sum_y I(x, y) e^{-i2\pi(\omega_x x/M + 0 \cdot y/N)} = F(\omega_x, 0) \quad (103)$$

and similarly for $I_y(y)$

$$F_y(\omega_y) = F(0, \omega_y). \quad (104)$$

Based on Eq. (67) we know that

$$F_2(\omega_x, 0) = F_1(\omega_x, 0) \exp(-i2\pi\omega_x v_x/M), \quad (105)$$

which yields

$$F_{2x}(\omega_x) = F_{1x}(\omega_x) \exp(-i2\pi\omega_x v_x/M), \quad (106)$$

where F_{1x} and F_{2x} are the Fourier transforms of the x projections for both images I_1 and I_2 respectively. Then the phase correlation method could be applied to each 1D projection to estimate the

displacement. In this case, the 1D cross-power spectrum C_x and the 1D PCS c_x for the x direction becomes

$$C_x(\omega_x) = \frac{F_{1x}(\omega_x)F_{2x}^*(\omega_x)}{|F_{1x}(\omega_x)F_{2x}^*(\omega_x)|}, \quad c_x(x) = \mathcal{F}^{-1}\left\{C_x(\omega_x)\right\}(x). \quad (107)$$

Equivalently for y , given F_{1y} and F_{2y} the Fourier transforms of y projections for images I_1 and I_2 respectively, both vectors are defined as

$$C_y(\omega_y) = \frac{F_{1y}(\omega_y)F_{2y}^*(\omega_y)}{|F_{1y}(\omega_y)F_{2y}^*(\omega_y)|}, \quad c_y(y) = \mathcal{F}^{-1}\left\{C_y(\omega_y)\right\}(y). \quad (108)$$

The authors of [4] suggest to compute both shift estimates using 1D versions of the phase correlation methods presented so far, by windowing the input images. Nevertheless, they do so for integer displacements. In Fig. 5 one can see two images shifted by $(-3.5, 0.75)$ pixels, under different noise intensities. One observes that the noise does not affect so much the projections, however differences between the projections appear due to objects entering and leaving the scene. This indeed affects the performance of projection-based methods.

Robinson and Milanfar [74] generalized the notion of projection by using the Radon transform at an angle θ defined on the continuous image $f(x, y)$ as,

$$\mathcal{R}(p, \theta)[f(x, y)] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \delta(p - x \cos(\theta) - y \sin(\theta)) \, dx \, dy, \quad (109)$$

where p is the perpendicular distance from a line to the origin and θ is the angle formed by the distance vector. Then by using two perpendicular angles $\theta_i, i = 1, 2$, they estimate the 2D shift by first calculating both 1D shifts u_{θ_1} and u_{θ_2} to finally compute

$$\begin{bmatrix} \hat{v}_x \\ \hat{v}_y \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \end{bmatrix}^{-1} \begin{bmatrix} u_{\theta_1} \\ u_{\theta_2} \end{bmatrix}. \quad (110)$$

Each 1D shift u_{θ_i} is computed using a line fitting approach on the unwrapped phases of the 1D normalized cross-power spectrums $C_{\theta}(\omega_x)$, using a masking procedure resembling the approach of Stone [86], as presented in section 3.2. The shift in the θ direction is therefore estimated by computing

$$\underset{u_{\theta}}{\operatorname{argmin}} \sum_{\omega} M_{\theta}(\omega) [2\pi\omega u_{\theta}/M - \phi_{\theta}(\omega)]^2, \quad (111)$$

where $\phi_{\theta}(\omega) = \arg(C_{\theta}(\omega))$ and $M_{\theta}(\omega)$ is the 1D equivalent of the binary weighting mask used to reduce contaminated frequencies. The shift in the y directions is calculated using the same strategy. To keep the computational complexity low, they again reduce to the setup presented above by choosing $\theta_1 = 0$ and $\theta_2 = \pi/2$, so both C_{θ_1} and C_{θ_2} are defined by Eqs. (107) and (108) respectively. Finally, they conclude that using a projection-based approach achieves dramatic savings in computation with essentially no degradation in final accuracy. In Fig. 6 we see the incidence of noise on the unwrapped phases of the cross-power spectrums of two images shifted by $(-3.5, 0.75)$. When the noise is low, a line fitting algorithms seems suitable, however under lower SNR scenarios, it seems the method accuracy will suffer.

Ren *et al.* [73] use the gradients of the projected components to improve even further the accuracy and robustness of the method to the ignored interference terms in Eq. (66). Indeed, let $I'_{2x}(x) =$

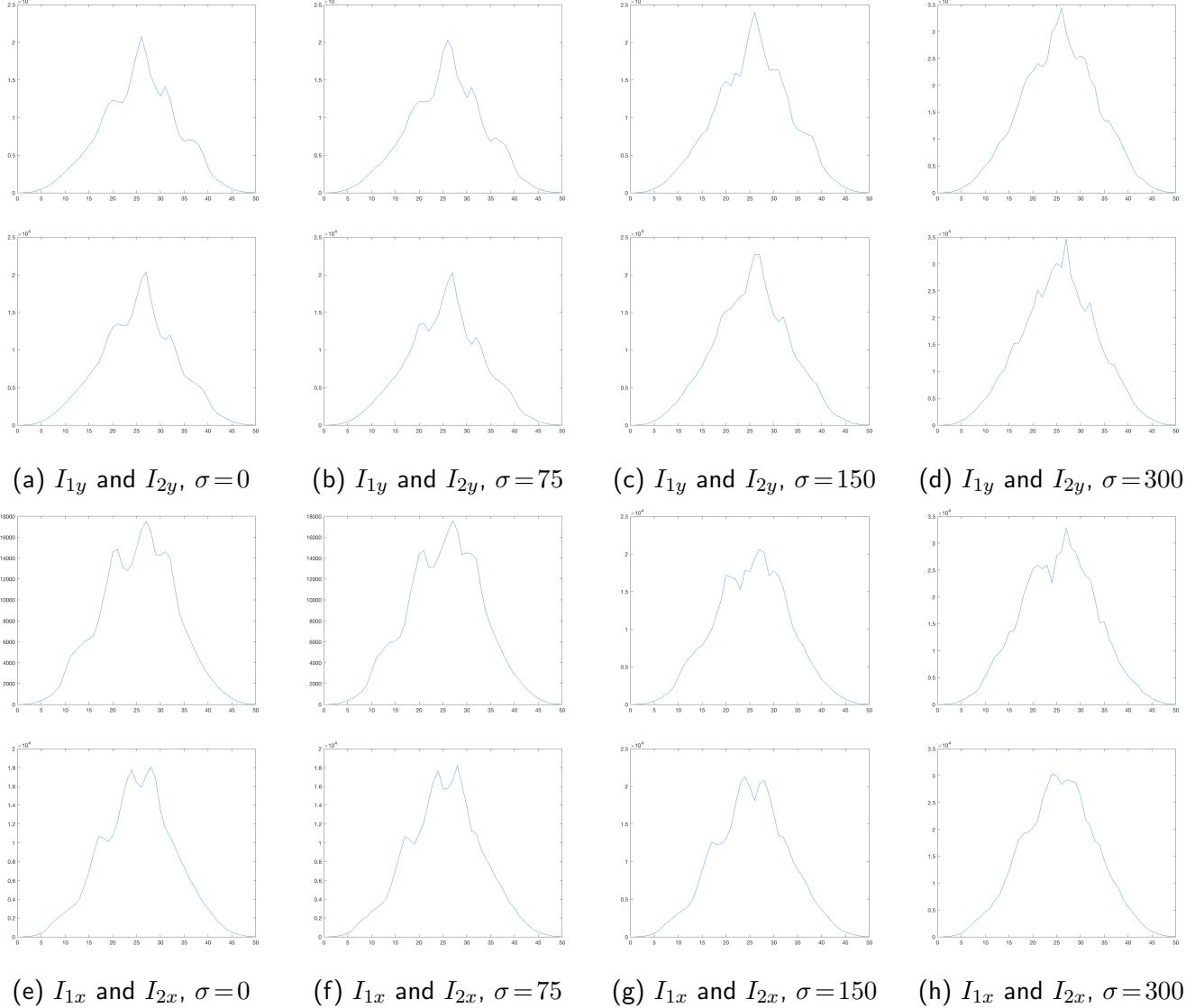


Figure 5: Projections of two images shifted by $(-3.5, 0.75)$ pixels under different random WGN ($\sigma = 0, 75, 150$ and 300) assuming 12-bit images. **Top two rows:** Vertical direction of both images (shift 0.75). **Last two rows:** Horizontal direction of both images (shift -3.5).

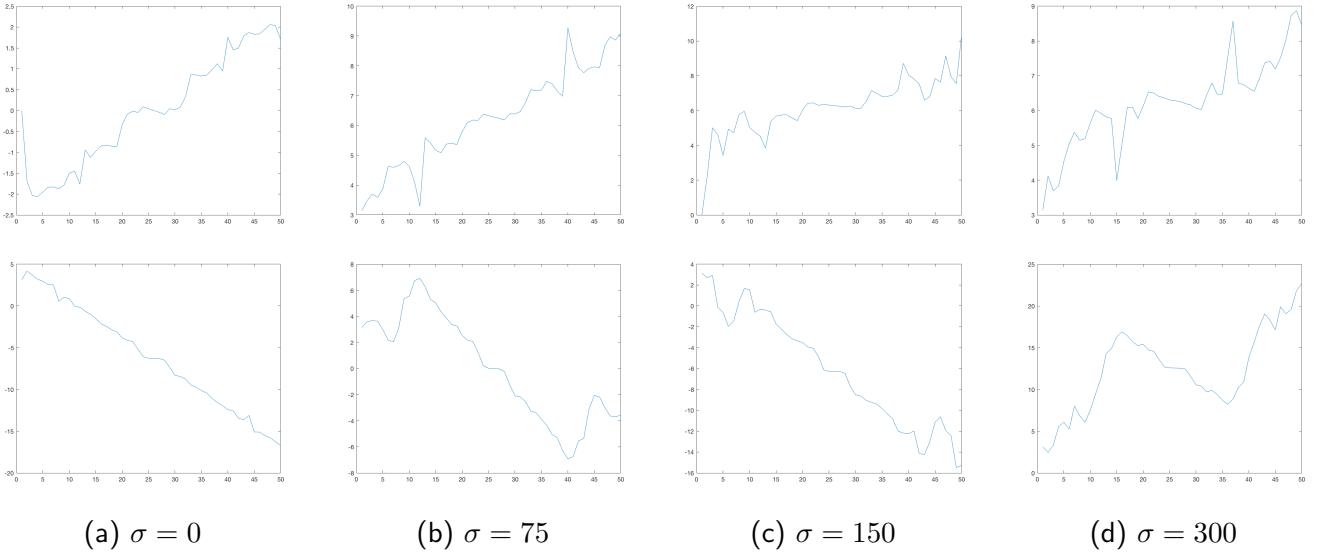


Figure 6: Unwrapped phases obtained from the normalized cross-power spectrum of two images shifted by $(-3.5, 0.75)$ pixels under different random WGN ($\sigma = 0, 75, 150$ and 300) assuming 12-bit images. Both images were pre-filtered using a Blackman window. **Top:** Vertical direction (shift 0.75). **Bottom:** Horizontal direction (shift -3.5).

$I_{2x}(x+1) - I_{2x}(x)$ then

$$I'_{2x}(x) = I_{2x}(x+1) - I_{2x}(x) \quad (112)$$

$$= \sum_y I_2((x+1) \bmod M, y) - \sum_y I_2(x, y) \quad (113)$$

$$= \sum_y [I_1((x+1 - v_x) \bmod M, (y - v_y) \bmod N) + n((x+1) \bmod M, y)] - \sum_y [I_1((x - v_x) \bmod M, (y - v_y) \bmod N) + n(x, y)] \quad (114)$$

$$= I'_{1x}(x - v_x) + n'_x(x), \quad (115)$$

where $n'_x(x) = n_x(x+1) - n_x(x)$ and $n_x(x) = \sum_y n(x, y)$. When N is large enough, the authors point out that $n'_x(x) = 0$, thus yielding $I'_{2x}(x) = I'_{1x}(x - v_x)$. This means that the gradients of the projections are not affected by the interference terms, therefore the method becomes more robust to them, even under non-zero-mean noise.

Then the PCS between both gradients is computed and the subpixel shifts are estimated by fitting a Gaussian around the peak values. To increase robustness while also accelerating the computations, the peak value is only searched around the interval $[\frac{7M}{16}, \frac{9M}{16}]$, using one-eighth of the samples.

The rank-one approach A straightforward but usually overlooked aspect about the phase correlation matrix of Eq. (69) is that, in a noiseless context, each element could be written as the product of two complex exponentials,

$$C(\omega_x, \omega_y) = \exp \left(i2\pi\omega_x v_x / M \right) \cdot \exp \left(i2\pi\omega_y v_y / N \right) = q_x(\omega_x) q_y(\omega_y). \quad (116)$$

This means that matrix $C(\omega_x, \omega_y)$ can be written as the outer product of non-zero vectors, implying it has rank one. By making this observation, Hoge [43] proposed to compute the singular value

decomposition (SVD) of the phase correlation matrix C to obtain a rank-1 approximation, followed by applying linear least squares on unwrapped versions of both $\arg(q_x)$ and $\arg(q_y)$, the left and right dominant singular vectors. Since both are 1D signals, the unwrapping is straightforward and the computational cost of the minimization is significantly lower. To gain robustness against aliasing and edge effects, the unwrapped 1D phases are masked by only including frequencies lying on a specific range, namely $|\omega| \in [2, 0.6(s/2)]$ where s is the length of the array. Fig. 4 puts in evidence the effects of computing the rank-1 approximation of the matrix $C(\omega_x, \omega_y)$. While the original cross-power spectrum is noisy, its rank-1 approximation yields a much cleaner version. Its effects under low SNR are however not stable, affecting the accuracy. Interestingly, the phase difference matrix gets considerably denoised using the rank-1 approximation, as seen in Fig. 7. The unwrapped versions of the 1D phase components for both left and right singular vectors are also displayed in this figure. While on the vertical direction, the line fitting seems straightforward in most cases, the same does not occur on the other direction due to the high amount of outliers. A robust line fitting method such as RANSAC could be used in this task. Indeed, this is the main idea behind the recent article of Tong *et al.* [90].

3.4 Gradient correlation methods

The phase correlation (PC) methods discussed above are able to estimate image displacements because the phase component of images holds structural information. Therefore by correlating the phase components between two images, it is possible to estimate the displacement between them. Indeed, phase information is invariant with respect to uniform variations of illumination, has a strong response to edges and usually yields high peak localization accuracy, which makes phase-correlation approaches suitable for several shift estimation tasks. Nevertheless, other shift estimation approaches, which resemble PC methods, are based on correlating the image gradients [6, 93, 92].

For each input image, the gradient correlation (GC) approach [6] combines both horizontal and vertical gradients into a new image using complex numbers. Then, it computes the cross-correlation of these two complex images in the frequency domain to finally estimate the peak. Formally, images g_1 and g_2 are given by

$$g_i(\mathbf{x}) = \frac{\partial I_i}{\partial x}(\mathbf{x}) + j \frac{\partial I_i}{\partial y}(\mathbf{x}), \quad i = 1, 2. \quad (117)$$

If $G_1 = \mathcal{F}\{g_1(\mathbf{x})\}$ and $G_2 = \mathcal{F}\{g_2(\mathbf{x})\}$ are their respective Fourier representations, $|G_1| = \mathcal{F}\{|g_1(\mathbf{x})|\}$ and $|G_2| = \mathcal{F}\{|g_2(\mathbf{x})|\}$ the Fourier representations of their magnitudes, then the gradient correlation (GC) and the normalized gradient correlation NGC are as follows

$$GC(\mathbf{x}) = g_1(\mathbf{x}) \star g_2(\mathbf{x}) = \mathcal{F}^{-1}\{G_1^* G_2\}(\mathbf{x}) \quad (118)$$

$$NGC(\mathbf{x}) = \frac{\mathcal{F}^{-1}\{G_1^* G_2\}}{\mathcal{F}^{-1}\{|G_1|^* |G_2|\}}. \quad (119)$$

The images are previously zero padded to avoid undesired edge effects in the correlation. Finally, to achieve sub-pixel accuracy, the same quadratic fitting proposed in Abdou [1] is used around the peak value. An example of a gradient correlation surface is shown on the first row of Fig. 8.

In [93] the method is improved by fitting a 1D kernel, based on the mexican hat wavelet, to the left and right singular vectors of the SVD of the (normalized) gradient correlation grid. The fitting is performed through least squares minimization and resolved using the Levenberg-Marquardt [53] algorithm. For the minimization to converge rapidly, the method is initialized with the preliminary result of applying quadratic fitting to the rank-1 approximation of the (normalized) gradient correlation (Fig. 8, middle row). Therefore, given either $GC(\mathbf{x})$ or $NGC(\mathbf{x})$, the method first computes

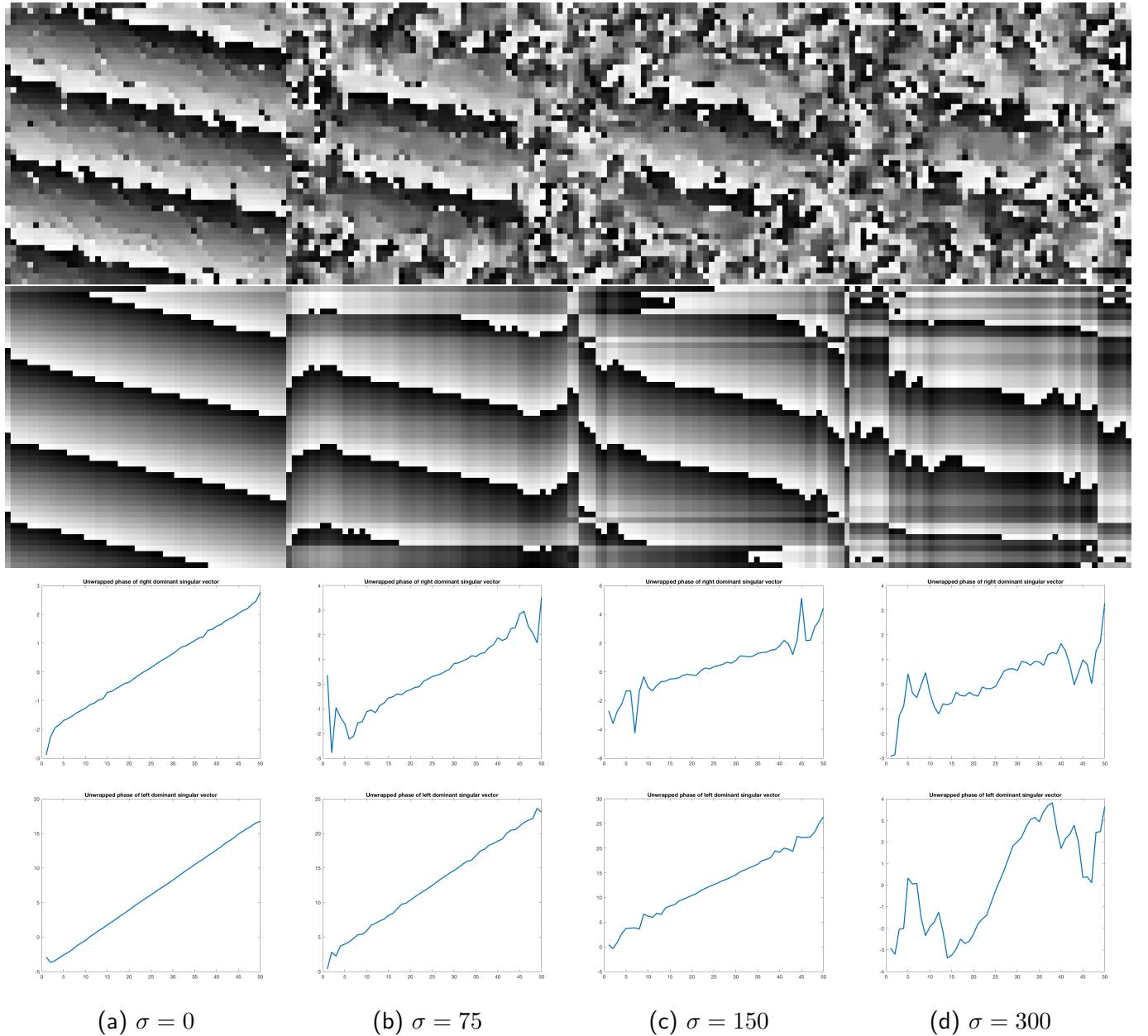


Figure 7: Phase difference matrix computed directly (first row) and by using a rank-1 approximation (second row) of the cross-power spectrum of two images shifted by $(-3.5, 0.75)$ pixels under different random WGN ($\sigma = 0, 75, 150$ and 300) assuming 12-bit images. Third and fourth rows display the phase unwrapped left and right dominant singular vectors of the cross-power spectrum respectively. Both images were pre-filtered using a Blackman window. Dynamic ranges extended for visualization purposes.

its rank-1 approximation $GC_1 = \lambda_1 U_1 V_1^T$ using its SVD, and uses GC_1 to compute an initial shift v_x by linearly fitting Eq. (75) to its peak value. To refine the results, it fits the following kernel using $2R + 1$ samples around the peak on each U and V

$$K_{1D}(x) = p_1 \left\{ 1 - (p_2(x - v_x))^2 \right\} \frac{1}{\sqrt{2\pi p_3}} \exp \left\{ -\frac{(x - v_x)^2}{2p_3^2} \right\} \quad (120)$$

where $[v_x, p_1, p_2, p_3]$ are the kernel parameters being optimized. As initial values for the optimization, they propose to use $p_1 = p_2 = p_3 = 1$ and $R = 10$. It should be noted that the authors evaluate two possible normalized gradient correlation rank-1 approximations. Either they compute directly the SVD of Eq. (119) as already explained, either they divide the rank-1 approximations of both numerator and denominator of Eq. (119), followed by taking again the rank-1 of the quotient, as in

$$NGC(\mathbf{x}) = \text{rank1} \left\{ \frac{\text{rank1}(\mathcal{F}^{-1} \{G_1^* G_2\})}{\text{rank1}(\mathcal{F}^{-1} \{|G_1|^* |G_2|\})} \right\}. \quad (121)$$

This last approach, whose NGC grid is shown in the last row of Fig. 8, seems to offer slightly better results, although the difference is not significative.

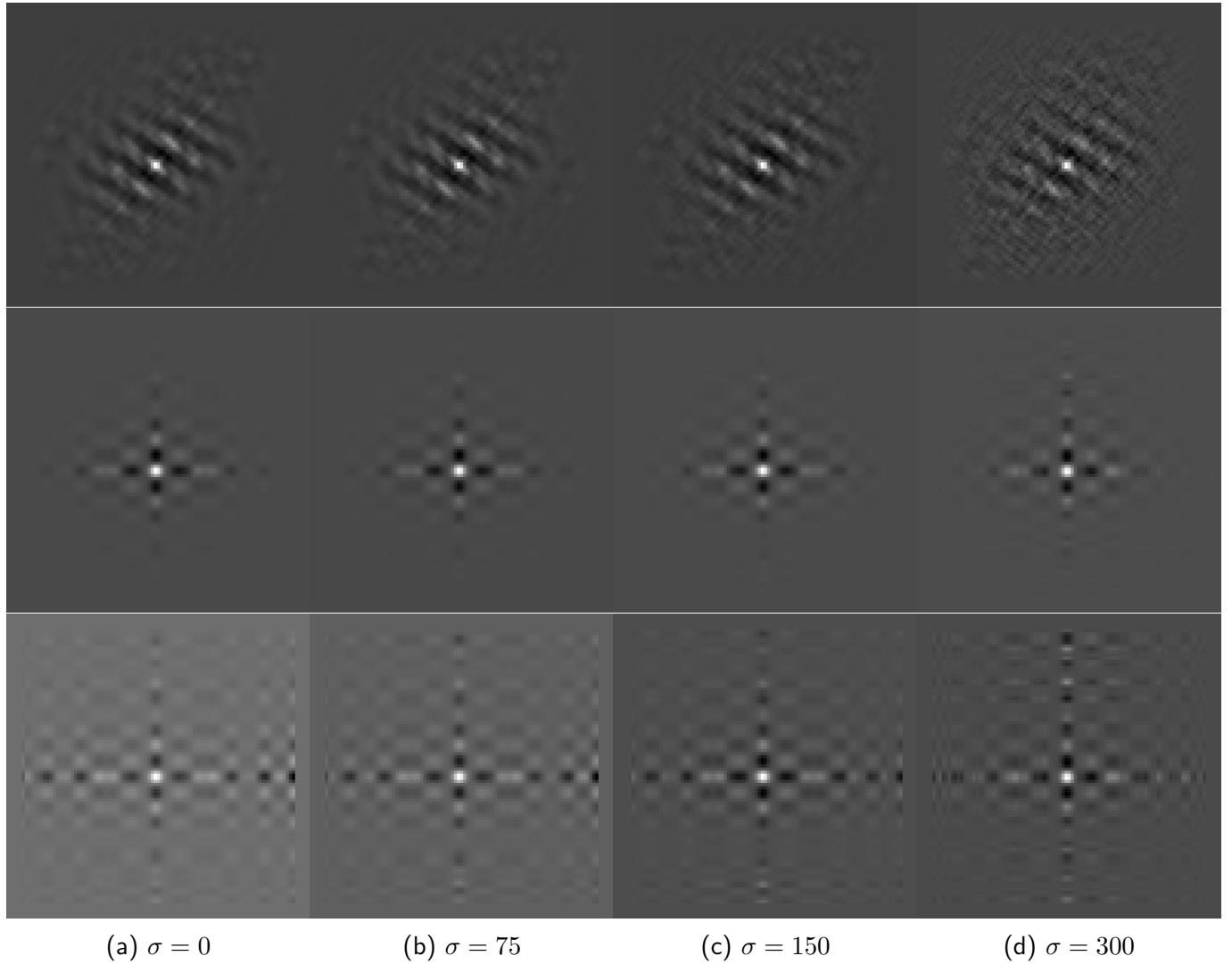


Figure 8: Gradient correlation without (top) and by using a rank-1 approximation (center) and the normalized gradient correlation after forcing rank-1 (bottom) of two images shifted by $(-3.5, 0.75)$ pixels under different random WGN ($\sigma = 0, 75, 150$ and 300) assuming 12-bit images. Both images were zero-padded up to $2N - 1$ pixels [92]. Dynamic ranges extended for visualization purposes.

4 Method evaluation

In this section we perform a full evaluation of the presented methods, under different parametrizations, on several shift-estimation tasks. For experimentation, all experiments were done using a high-resolution satellite image of the city of Cannes 9. In a simulated environment, I_1 was generated by taking a 50×50 subimage from a random location, while I_2 was obtained by shifting the high-resolution image in the Fourier domain using Eq. (67) followed by taking a subimage with the same size and from the same location. All shift estimation errors were computed as the root mean squared error (RMSE) and are in pixels. This implies that if \mathbf{v} and $\hat{\mathbf{v}}$ are the real shift and the estimated shift, then the error obtained is

$$E(\hat{\mathbf{v}}) = \sqrt{\frac{(v_x - \hat{v}_x)^2 + (v_y - \hat{v}_y)^2}{2}}. \quad (122)$$

Results presented in this evaluation are, in every case, obtained by averaging 100 realizations.



Figure 9: Input image used for simulation.

Apart from the most important shift-estimation methods introduced in the previous section, five more approaches coming from the remote sensing community were also evaluated. Three of these methods (SDF, ADF and ADF2 [54]) are loosely based on correlating both images followed by fitting a conic section in the 2D neighbourhood around the peak, while ACC [80] and APC [79] are two iterative methods, the former being based on the method of Stone [86] while the latter is based on a periodic correlation technique.

Due to the extensive amount of evaluated methods and variants, we defined a nomenclature to refer to each of them. In Table 2 we observe all evaluated methods together with their parameters. For every method, parameter \mathbf{i} refers to the amount of iterations ($\mathbf{i} \in [1, 2, 3, 4]$ in our experiments), \mathbf{gr} to a gradient estimation method presented in section 2.6 (Table 3), \mathbf{int} to an interpolation method presented in section 2.7 (Table 5), \mathbf{sp} to the support size, \mathbf{up} to the upsampling factor, \mathbf{win} to the apodization window used (Table 4) and \mathbf{d} to the dimensionality of the solution ($\mathbf{d} \in [1, 2]$).

For the multiscale GBSE method (MS), we first put the amount scales followed by a coma. Then we specify its configuration by putting for each scale the amount of iterations \mathbf{i} followed by the

interpolation method *int* used in each scale from the finest to the coarsest scale. For example, MS-3,321-IdssGh represents three scales, with a single iteration and spline interpolation on the coarsest scale, followed by two iterations and spline interpolation in the intermediate scale, and finally doing three iterations and DFT interpolation with symmetrization in the original scale, and where the image gradients are always computed using the hypomode.

Code	Parameters	Method	Reference
LS	LS- <i>i</i> - <i>Iint</i> - <i>Ggr</i>	Least Squares GBSE	Alg. 1
TLS	TLS- <i>i</i> - <i>Iint</i> - <i>Ggr</i>	Total Least Squares GBSE	Eq. (10)
CLS	CLS- <i>i</i> - <i>Iint</i> - <i>Ggr</i>	Corrected Least Squares GBSE	Eq. (22)
ULS	ULS- <i>Ggr</i>	Bidirectional bias correction GBSE	[63]
MS	MS- <i>s</i> , <i>i</i> - <i>Iint</i> <i>Ggr</i>	Multiscale GBSE	Alg. 2
INT	INT- <i>sp</i>	Interpolation method	Sec. 2.8
PC-GUIZAR	PC-GUIZAR- <i>up</i>	Zero-fitting the cross-power spectrum	[39]
PCSTONE	PCSTONE-W <i>win</i>	Robust plane fitting on the phase difference matrix	[86]
PC-QUADFIT	PC-QUADFIT-W <i>win</i>	Quadratic fitting on the phase correlation surface	[1]
PC-GAUSSFIT	PC-GAUSSFIT-W <i>win</i>	Gaussian fitting on the phase correlation surface	[1]
PCFOO	PCFOO-W <i>win</i>	Max. of sinc approx. to phase correlation surface	[32]
PC-SINC	PC-SINC-W <i>win</i>	Sinc fitting on the phase correlation surface	[7]
PC-ESINC	PC-ESINC-W <i>win</i>	E-Sinc fitting on the phase correlation surface	[7]
PC-LCM	PC-LCM- <i>dDsp</i>	<i>d</i> -dimensional local center of mass of the PCS	[3]
PC-REN2010	PC-REN2010-W <i>win</i>	Difference between both side-peaks of the PCS	[72]
SS-HOGE	SS-HOGE-W <i>win</i>	Subspace method by Rank-1 approximation	[43]
SS-ROBINSON	SS-ROBINSON-W <i>win</i>	Projection-based subspace phase correlation	[74]
SS-REN2014	SS-REN2014-W <i>win</i>	Projection-based subspace gradient correlation	[73]
GC04	GC04- <i>Ggr</i>	Gradient Correlation	[6]
GC11	GC11- <i>Ggr</i>	Kernelized Gradient Correlation	[93]
NGC04	NGC04- <i>Ggr</i>	Normalized Gradient Correlation	[6]
NGC11	NGC11- <i>Ggr</i>	Kernelized Normalized Gradient Correlation	[93]
SDF	SDF-2QI	Sum of squared differences	[54]
ADF	ADF-2QI	Sum of absolute differences differences	[54]
ADF2	ADF2-2QI	Sum of squared absolute differences differences	[54]
ACC	ACC- <i>iIt</i>	Adaptive Cross-Correlation	[80]
APC	APC- <i>iIt</i>	Adaptive Periodic-Correlation	[79]

Table 2: Method references for this evaluation section.

As mentioned in section 1, when evaluating the methods in the presence of noise, we assumed white Gaussian noise uncorrelated with the signal. Simulating Ren *et al.* [72], we discretized the noise into five distinct levels (five different values for the noise standard deviation σ_N), shown in Table 6, and used this categorization to evaluate the methods.

In the same manner, all evaluated displacements were grouped into four categories. The reason behind this distinction is that some methods are more suited than others for each designated category. The ranges for each shift magnitude level are as shown in Table 7. In the first category, extremely small sub-pixel shifts were considered, as they could exist in several applications. We also include as a fourth category, magnitudes larger than 1.1 pixels. By design, all non-multiscale GBSE methods as well as the phase correlation method from Stone [86] fail under this category. Due to this reason, although being shown for completeness, this shift magnitude is excluded from the general averages on every table.

We begin our evaluation by studying the impact of gradient estimation methods on GBSE approaches.

Code	Name
h	Hypomode
g0.3	3×3 Gaussian derivative $\sigma = 0.3$
g0.6	5×5 Gaussian derivative $\sigma = 0.6$
g1	7×7 Gaussian derivative $\sigma = 1$
sim3	3×3 Simoncelli derivative
sim5	5×5 Simoncelli derivative
sim7	7×7 Simoncelli derivative
fa3	3×3 Farid derivative
fa5	5×5 Farid derivative
fa7	7×7 Farid derivative
ch1	1st order Christmas derivative
ch2	2nd order Christmas derivative
ch3	3rd order Christmas derivative

Table 3: Evaluated gradient estimation method codes **gr**.

Code	Name
nw	No Window
ex	Image zero-padding
bm	Blackman
bh	Blackman-Harris
bl	Bartlett
bw	barthann
cw	Chebyshev
gw	Gaussian
hw	Hamming
tw	Tukey
ft	Flat-top

Table 5: Evaluated interpolation method codes **int**.

Table 4: Window codes **win** used for apodization.

Level	1	2	3	4	5
σ_N	0.000	0.005	0.015	0.025	0.055

Table 6: Std. dev. of white Gaussian noise for each noise level injected to the simulated images.

4.1 Influence of gradient estimation on GBSE methods

Since GBSE methods are gradient-based, their performance obviously depends on the method used to estimate the image gradient. To this end, we evaluated several gradient estimation methods using the original least-squares approach as explained in section 2.6. Emulating Ren *et al.* [72], we used all possible shifts obtained by taking two values from $[-0.875, -0.75, -0.5, -0.25, -0.125, -0.07, -0.02, 0, 0.03, 0.125]$. In total, this represented 196 shifts, from which for each experiment, a random sub image was obtained and shifted in the frequency domain using the Fourier shift theorem. The average error obtained for all 196 shifts is shown in Fig. 10 as the noise increases (left) or by varying shift magnitude (right), using the five noise levels shown in Table 6 and the four shift magnitudes of Table 7.

Several conclusions are drawn following the preceding experiment.

- First, the most accurate gradient estimation method for GBSE proved to be the method of Farid [29], followed by the approach of Simoncelli [82].
- Second, the kernel support size should be set larger (7×7) under higher noise scenarios, and shorter (3×3) when the SNR is high.
- Third, using $\sigma = 0.6$ for the Gaussian kernel always obtained the best results, however the performance using $\sigma = 1$ seems less affected by higher noise situations, and should be considered in those cases. A more in-depth study on Gaussian gradient estimation performance for GBSE methods is given in the following section.
- Finally, even though computationally cheap, the hypomode gradient estimation should be avoided when shift estimation is done on all possible shift magnitudes between the $[-1,1]$ interval. Nevertheless, under low shift magnitudes and high noise, the hypomode becomes the most accurate method, as seen in Table 8, where the top 10 gradient estimation methods are shown depending on both the shift magnitude and the noise.

Level	1	2	3	4
Range	$\ \mathbf{v}\ \leq 0.1$	$0.1 < \ \mathbf{v}\ \leq 0.5$	$0.5 < \ \mathbf{v}\ \leq 1.1$	$\ \mathbf{v}\ > 1.1$

Table 7: Shift magnitude ($\|\mathbf{v}\|$) ranges for each evaluated category.

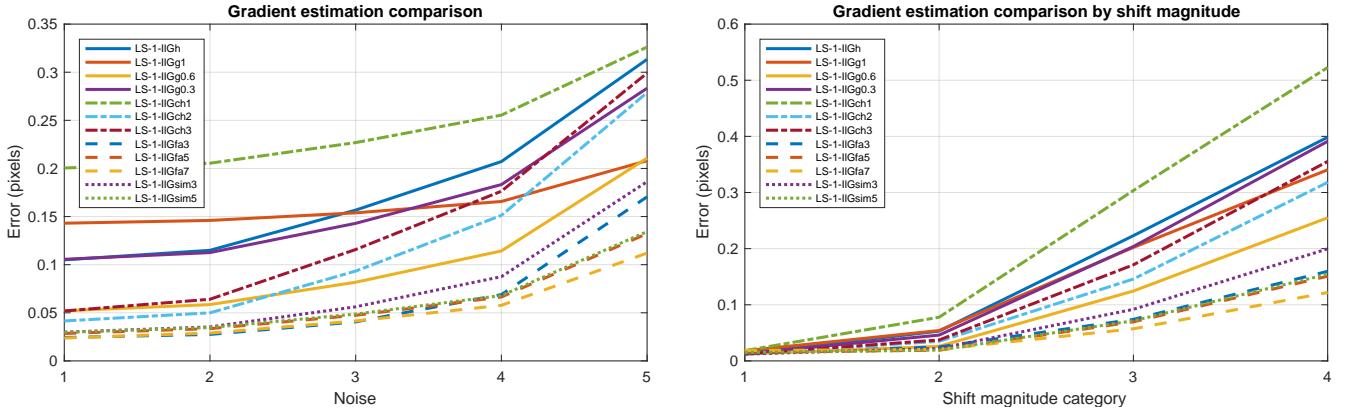


Figure 10: Comparison of accuracy obtained by using different gradient estimation methods on the GBSE algorithm. **Left:** By noise level. **Right:** By shift magnitude category.

4.1.1 A study on Gaussian derivatives

An experiment was performed in order to study the influence of prefiltering using a Gaussian blur and computing the Gaussian derivatives to estimate the gradient. The objective was to search for the standard deviation $\hat{\sigma}$ of the Gaussian kernel h that minimized the registration error

$$\hat{\sigma} = \operatorname{argmin}_{\sigma} \|(A^T A)^{-1} A^T b - \mathbf{v}\|, \quad (123)$$

where

$$\mathbf{A} = \begin{pmatrix} (h_{\sigma}^x * I_1)(p_1) & (h_{\sigma}^y * I_1)(p_1) \\ \vdots & \vdots \\ (h_{\sigma}^x * I_1)(p_n) & (h_{\sigma}^y * I_1)(p_n) \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}, \quad \mathbf{b} = - \begin{pmatrix} (h_{\sigma} * I_t)(p_1) \\ \vdots \\ (h_{\sigma} * I_t)(p_n) \end{pmatrix}. \quad (124)$$

In the previous expression, $*$ stands for convolution and $h_{\sigma}^x, h_{\sigma}^y$ are the derivatives of the Gaussian kernel on both dimensions, that depend on the σ parameter. By performing simulation, two images shifted by \mathbf{v} were generated and WGN was added. The standard GBSE algorithm was used by computing the Gaussian derivative from the original I_1 image, and by blurring I_t with a Gaussian with the same standard deviation, as explained in section 2.6. The minimization was performed using the Levenberg-Marquardt non-linear least squares optimization method. In Fig. 11, the evolution of $\hat{\sigma}$ is shown for each shift, as the noise of the input image increases. While for extremely small displacements (or no displacement), the $\hat{\sigma}$ varied in a random manner, for displacements already as small as $(0.0743, 0)$ and larger, the obtained values tended to be along 0.6 up to 0.7 as the noise increased.

To better understand the importance of choosing the correct $\hat{\sigma}$, in Fig. 12, we performed two other experiments. In Fig. 12(a) the average standard deviation of the top 10 best RMSEs obtained by varying the σ parameter is displayed, for each displacement. This implies to test each one of the values for the σ parameter between 0 and 1 with step size of 0.05, and obtain for each the resulting RMSE to finally compute the standard deviation of the lowest 10. For consistency, this experiment was averaged several times. As seen from the figure, the gain gets higher with the displacement magnitude. However, when the shift magnitude is under 0.1 pixels, this gain is negligible, suggesting in this case, a more relaxed selection of the σ value.

Finally, since $\sigma = 0.6$ usually achieved good results, a comparison between the best $\hat{\sigma}$ and $\sigma = 0.6$ was made in Fig. 12(b). Interestingly, by fixing $\sigma = 0.6$, higher accuracy gains could be obtained when

	$\ \mathbf{v}\ \leq 0.1$	$0.1 < \ \mathbf{v}\ \leq 0.5$	$0.5 < \ \mathbf{v}\ \leq 1.1$	$\ \mathbf{v}\ > 1.1$	Avg.
σ	0.0000 LS-1-IIGfa7	0.0010 LS-1-IIGfa5	0.0245 LS-1-IIGfa3	0.0733 LS-1-IIGfa3	0.0286 LS-1-IIGfa7
	0.0001 LS-1-IIGsim5	0.0010 LS-1-IIGfa7	0.0290 LS-1-IIGfa7	0.0843 LS-1-IIGfa7	0.0291 LS-1-IIGfa3
	0.0001 LS-1-IIGfa5	0.0012 LS-1-IIGsim5	0.0333 LS-1-IIGsim3	0.1025 LS-1-IIGfa5	0.0346 LS-1-IIGfa5
	0.0007 LS-1-IIGg0.6	0.0046 LS-1-IIGg0.6	0.0347 LS-1-IIGfa5	0.1054 LS-1-IIGsim5	0.0357 LS-1-IIGsim5
	0.0015 LS-1-IIGsim3	0.0052 LS-1-IIGsim3	0.0362 LS-1-IIGsim5	0.1163 LS-1-IIGsim3	0.0391 LS-1-IIGsim3
	0.0020 LS-1-IIGch3	0.0067 LS-1-IIGch3	0.0420 LS-1-IIGch2	0.1717 LS-1-IIGg0.6	0.0592 LS-1-IIGch2
	0.0033 LS-1-IIGch2	0.0128 LS-1-IIGch2	0.0583 LS-1-IIGch3	0.1785 LS-1-IIGch2	0.0603 LS-1-IIGg0.6
	0.0037 LS-1-IIGfa3	0.0150 LS-1-IIGfa3	0.0641 LS-1-IIGg0.6	0.2094 LS-1-IIGch3	0.0691 LS-1-IIGch3
	0.0045 LS-1-IIGg0.3	0.0216 LS-1-IIGg0.3	0.1305 LS-1-IIGg0.3	0.2725 LS-1-IIGh	0.1079 LS-1-IIGh
	0.0048 LS-1-IIGh	0.0231 LS-1-IIGh	0.1315 LS-1-IIGh	0.2952 LS-1-IIGg0.3	0.1130 LS-1-IIGg0.3
σ	0.0039 LS-1-IIGg0.6	0.0049 LS-1-IIGfa5	0.0260 LS-1-IIGfa3	0.0783 LS-1-IIGfa3	0.0311 LS-1-IIGfa3
	0.0041 LS-1-IIGsim3	0.0050 LS-1-IIGsim5	0.0314 LS-1-IIGfa7	0.0867 LS-1-IIGfa7	0.0323 LS-1-IIGfa7
	0.0043 LS-1-IIGch3	0.0057 LS-1-IIGfa7	0.0374 LS-1-IIGfa5	0.1051 LS-1-IIGfa5	0.0380 LS-1-IIGfa5
	0.0045 LS-1-IIGsim5	0.0061 LS-1-IIGsim3	0.0376 LS-1-IIGsim3	0.1080 LS-1-IIGsim5	0.0391 LS-1-IIGsim5
	0.0045 LS-1-IIGfa5	0.0068 LS-1-IIGch3	0.0390 LS-1-IIGsim5	0.1216 LS-1-IIGsim3	0.0423 LS-1-IIGsim3
	0.0049 LS-1-IIGch2	0.0072 LS-1-IIGg0.6	0.0508 LS-1-IIGch2	0.1774 LS-1-IIGg0.6	0.0644 LS-1-IIGg0.6
	0.0054 LS-1-IIGfa7	0.0111 LS-1-IIGch2	0.0691 LS-1-IIGg0.6	0.1923 LS-1-IIGch2	0.0648 LS-1-IIGch2
	0.0055 LS-1-IIGfa3	0.0145 LS-1-IIGfa3	0.0716 LS-1-IIGch3	0.2267 LS-1-IIGch3	0.0774 LS-1-IIGch3
	0.0059 LS-1-IIGg0.3	0.0238 LS-1-IIGg0.3	0.1372 LS-1-IIGg0.3	0.2863 LS-1-IIGh	0.1151 LS-1-IIGh
	0.0063 LS-1-IIGh	0.0264 LS-1-IIGh	0.1415 LS-1-IIGh	0.3025 LS-1-IIGg0.3	0.1173 LS-1-IIGg0.3
σ	0.0105 LS-1-IIGch2	0.0131 LS-1-IIGsim3	0.0400 LS-1-IIGfa3	0.0993 LS-1-IIGfa7	0.0435 LS-1-IIGfa7
	0.0108 LS-1-IIGch3	0.0142 LS-1-IIGfa5	0.0428 LS-1-IIGfa7	0.1134 LS-1-IIGfa3	0.0452 LS-1-IIGfa3
	0.0109 LS-1-IIGg0.6	0.0143 LS-1-IIGsim5	0.0510 LS-1-IIGfa5	0.1224 LS-1-IIGfa5	0.0502 LS-1-IIGfa5
	0.0110 LS-1-IIGsim3	0.0156 LS-1-IIGfa3	0.0527 LS-1-IIGsim5	0.1256 LS-1-IIGsim5	0.0515 LS-1-IIGsim5
	0.0110 LS-1-IIGg0.3	0.0161 LS-1-IIGfa7	0.0622 LS-1-IIGsim3	0.1563 LS-1-IIGsim3	0.0606 LS-1-IIGsim3
	0.0118 LS-1-IIGh	0.0176 LS-1-IIGg0.6	0.0960 LS-1-IIGg0.6	0.2137 LS-1-IIGg0.6	0.0845 LS-1-IIGg0.6
	0.0119 LS-1-IIGfa3	0.0178 LS-1-IIGch2	0.1080 LS-1-IIGch2	0.2705 LS-1-IIGch2	0.1017 LS-1-IIGch2
	0.0132 LS-1-IIGsim5	0.0232 LS-1-IIGch3	0.1382 LS-1-IIGch3	0.3113 LS-1-IIGch3	0.1209 LS-1-IIGch3
	0.0134 LS-1-IIGfa5	0.0362 LS-1-IIGg0.3	0.1741 LS-1-IIGg0.3	0.3223 LS-1-IIGg1	0.1434 LS-1-IIGg0.3
	0.0154 LS-1-IIGg1	0.0432 LS-1-IIGh	0.1905 LS-1-IIGg1	0.3524 LS-1-IIGg0.3	0.1445 LS-1-IIGg1
σ	0.0161 LS-1-IIGch2	0.0226 LS-1-IIGfa3	0.0592 LS-1-IIGfa7	0.1250 LS-1-IIGfa7	0.0589 LS-1-IIGfa7
	0.0162 LS-1-IIGg0.3	0.0241 LS-1-IIGfa5	0.0718 LS-1-IIGfa5	0.1552 LS-1-IIGfa5	0.0679 LS-1-IIGfa5
	0.0165 LS-1-IIGg0.6	0.0242 LS-1-IIGsim5	0.0738 LS-1-IIGsim5	0.1589 LS-1-IIGsim5	0.0694 LS-1-IIGsim5
	0.0166 LS-1-IIGsim3	0.0246 LS-1-IIGsim3	0.0744 LS-1-IIGfa3	0.1735 LS-1-IIGfa3	0.0720 LS-1-IIGfa3
	0.0168 LS-1-IIGch3	0.0267 LS-1-IIGfa7	0.1001 LS-1-IIGsim3	0.2139 LS-1-IIGsim3	0.0888 LS-1-IIGsim3
	0.0170 LS-1-IIGh	0.0313 LS-1-IIGg0.6	0.1351 LS-1-IIGg0.6	0.2720 LS-1-IIGg0.6	0.1137 LS-1-IIGg0.6
	0.0175 LS-1-IIGfa3	0.0398 LS-1-IIGch2	0.1816 LS-1-IIGch2	0.3436 LS-1-IIGg1	0.1516 LS-1-IIGch2
	0.0201 LS-1-IIGch1	0.0488 LS-1-IIGch3	0.2032 LS-1-IIGg1	0.3689 LS-1-IIGch2	0.1556 LS-1-IIGg1
	0.0205 LS-1-IIGg1	0.0533 LS-1-IIGg0.3	0.2146 LS-1-IIGch3	0.4136 LS-1-IIGch3	0.1735 LS-1-IIGch3
	0.0205 LS-1-IIGsim5	0.0552 LS-1-IIGg1	0.2242 LS-1-IIGg0.3	0.4194 LS-1-IIGg0.3	0.1783 LS-1-IIGg0.3
σ	0.0261 LS-1-IIGh	0.0512 LS-1-IIGfa5	0.1235 LS-1-IIGfa7	0.2137 LS-1-IIGfa7	0.1092 LS-1-IIGfa7
	0.0265 LS-1-IIGg0.3	0.0515 LS-1-IIGsim5	0.1530 LS-1-IIGfa5	0.2693 LS-1-IIGfa5	0.1277 LS-1-IIGfa5
	0.0274 LS-1-IIGch2	0.0535 LS-1-IIGfa7	0.1560 LS-1-IIGsim5	0.2742 LS-1-IIGsim5	0.1296 LS-1-IIGsim5
	0.0274 LS-1-IIGch1	0.0569 LS-1-IIGfa3	0.2051 LS-1-IIGfa3	0.3593 LS-1-IIGfa3	0.1626 LS-1-IIGfa3
	0.0276 LS-1-IIGg0.6	0.0616 LS-1-IIGsim3	0.2262 LS-1-IIGsim3	0.3900 LS-1-IIGsim3	0.1765 LS-1-IIGsim3
	0.0281 LS-1-IIGsim3	0.0696 LS-1-IIGg0.6	0.2548 LS-1-IIGg1	0.4156 LS-1-IIGg1	0.1942 LS-1-IIGg1
	0.0281 LS-1-IIGch3	0.0735 LS-1-IIGg1	0.2582 LS-1-IIGg0.6	0.4400 LS-1-IIGg0.6	0.1989 LS-1-IIGg0.6
	0.0292 LS-1-IIGfa3	0.0927 LS-1-IIGch2	0.3468 LS-1-IIGch2	0.5801 LS-1-IIGch2	0.2618 LS-1-IIGch2
	0.0329 LS-1-IIGg1	0.0951 LS-1-IIGg0.3	0.3530 LS-1-IIGg0.3	0.5861 LS-1-IIGg0.3	0.2652 LS-1-IIGg0.3
	0.0369 LS-1-IIGsim5	0.1008 LS-1-IIGch3	0.3733 LS-1-IIGch3	0.6156 LS-1-IIGch3	0.2794 LS-1-IIGch3
Avg.	0.0119 LS-1-IIGg0.6	0.0191 LS-1-IIGfa5	0.0572 LS-1-IIGfa7	0.1218 LS-1-IIGfa7	0.0545 LS-1-IIGfa7
	0.0122 LS-1-IIGsim3	0.0192 LS-1-IIGsim5	0.0696 LS-1-IIGfa5	0.1509 LS-1-IIGfa5	0.0637 LS-1-IIGfa5
	0.0124 LS-1-IIGch3	0.0206 LS-1-IIGfa7	0.0715 LS-1-IIGsim5	0.1544 LS-1-IIGsim5	0.0651 LS-1-IIGsim5
	0.0125 LS-1-IIGch2	0.0221 LS-1-IIGsim3	0.0740 LS-1-IIGfa3	0.1596 LS-1-IIGfa3	0.0680 LS-1-IIGfa3
	0.0128 LS-1-IIGg0.3	0.0249 LS-1-IIGfa3	0.0919 LS-1-IIGsim3	0.1996 LS-1-IIGsim3	0.0815 LS-1-IIGsim3
	0.0132 LS-1-IIGh	0.0261 LS-1-IIGg0.6	0.1245 LS-1-IIGg0.6	0.2549 LS-1-IIGg0.6	0.1044 LS-1-IIGg0.6
	0.0135 LS-1-IIGfa3	0.0348 LS-1-IIGch2	0.1459 LS-1-IIGch2	0.3181 LS-1-IIGch2	0.1278 LS-1-IIGch2
	0.0150 LS-1-IIGsim5	0.0373 LS-1-IIGch3	0.1712 LS-1-IIGch3	0.3405 LS-1-IIGg1	0.1441 LS-1-IIGch3
	0.0152 LS-1-IIGfa5	0.0460 LS-1-IIGg0.3	0.2022 LS-1-IIGg1	0.3553 LS-1-IIGch3	0.1537 LS-1-IIGg1
	0.0180 LS-1-IIGg1	0.0530 LS-1-IIGh	0.2038 LS-1-IIGg0.3	0.3911 LS-1-IIGg0.3	0.1634 LS-1-IIGg0.3

Table 8: Average error per shift and magnitude of top 10 evaluated Least Squares GBSE methods using all gradient estimation approaches. **Rows:** five noise levels. **Columns:** four shift magnitudes. Note that the last column displays the top ten methods of the average for each noise level excluding the last shift magnitude level.

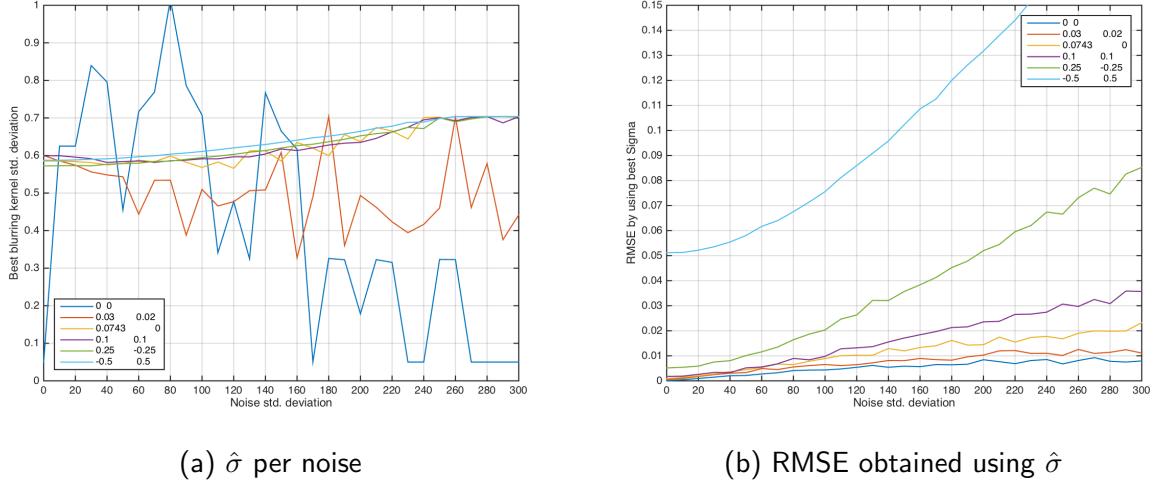


Figure 11: $\hat{\sigma}$ values obtained based on noise and shift. Standard deviation of noise is considering 12-bit images.

the displacements are of small magnitude (lower than 0.1 pixels). However, when the displacement is high enough (in our case half a pixel in magnitude), under low SNR values (noise std. dev. bigger than 150), the accuracy gain by selecting the correct $\hat{\sigma}$ could be as high as 0.016 pixels. This implies that for low displacement magnitudes, lower σ values should be used (around $\sigma = 0.45$); for medium shift magnitudes (between 0.1 and 0.3); $\sigma = 0.6$ should be used, and for high displacements, the decision to increase the σ relies on the noise level.

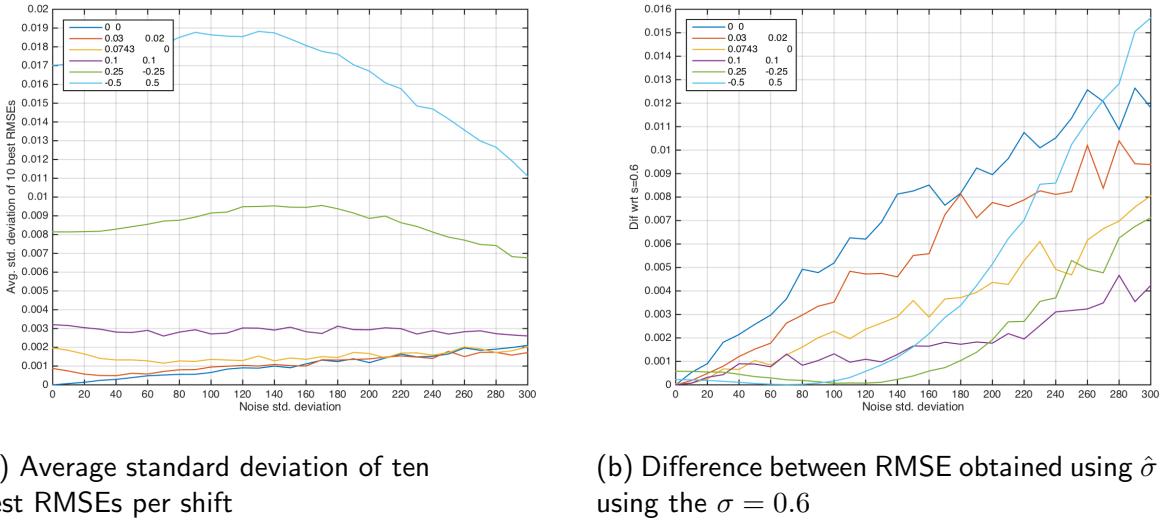


Figure 12: Influence of using the correct σ for the Gaussian kernel on the results

Another experiment was performed to confirm the $\hat{\sigma}$ values obtained under different image conditions. Three image types were selected, namely a textured image, a low SNR image, and an image having the well-known aperture problem (the gradient direction is mostly uniform across the image). Also, larger shifts up to $\|v\| = 1$ were included for this test. Results averaging all evaluated shifts are displayed in Fig. 13. It seems that for well-textured images, values of σ between 0.6 and 0.8 achieved the best performance, however for more complicated images, higher values achieved better results. This makes sense, since in a case where high precision is not possible, considerably blurring the image decreases the maximum shift returned and therefore the maximum error. This can be seen in Fig. 13(b) where the average errors reach almost half the magnitude and the largest evaluated

shift. It has to be noted that for each noise amount, each problematic case and each simulated shift,

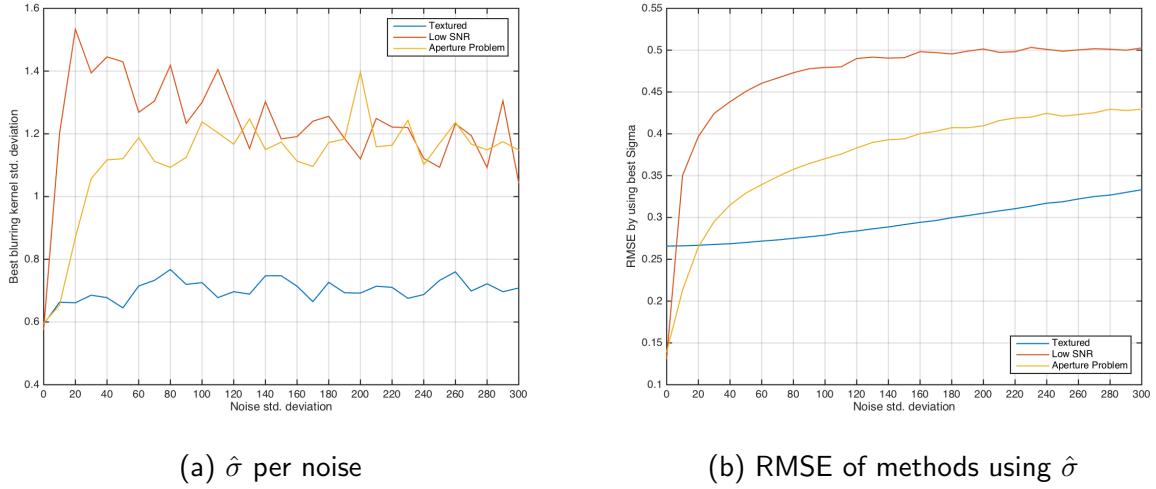


Figure 13: $\hat{\sigma}$ values obtained based on problematic case and noise.

a total of 1000 experiments were performed. The $\hat{\sigma}$ displayed on the figures is the median while the RMSE shown is the average RMSE of these experiments.

Finally, to give more insight about the progression of the error based on the blurring kernel standard deviation used and on the noise, Fig. 14 shows the RMSE, in pixels, depending on the underlying shift to be estimated. As it can be seen from the graphs, as the noise increases, also does the error, as expected, and the $\hat{\sigma}$ that minimizes this error seems to be around 0.6, growing as high as 0.7 when the noise becomes high. Another interesting result is that as the displacement grows, the fact that values of $\hat{\sigma}$ around 0.6 are the best becomes more apparent. However, when the shift is small enough (lower than 0.1 pixel), other $\hat{\sigma}$ values should be used, as remarked in the previous test. Furthermore, for the case of high shift magnitude and high noise, the estimation error becomes higher, so accurately setting σ in these cases could be justified by the accuracy gain.

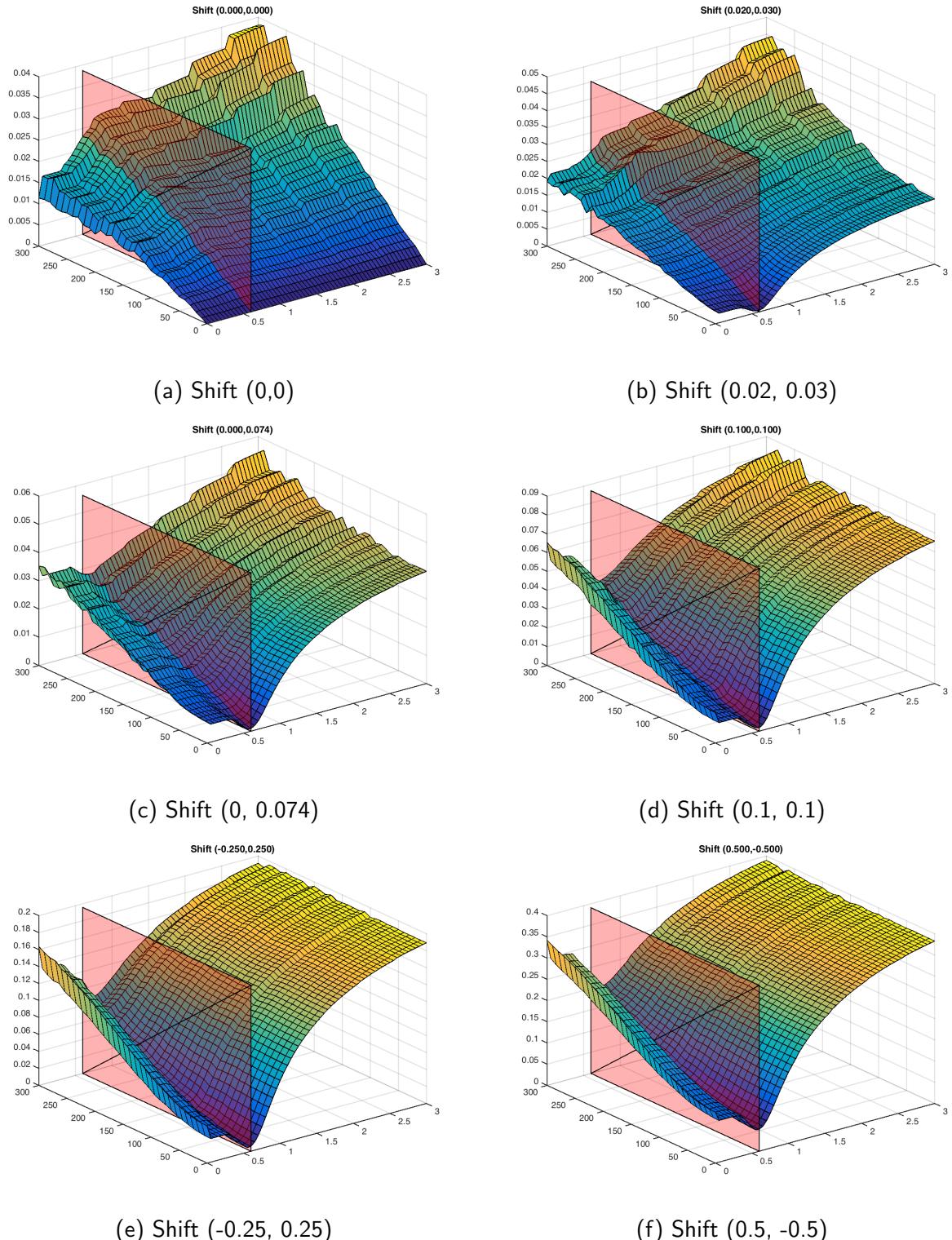


Figure 14: Evolution of RMSEs obtained based on $\hat{\sigma}$ values (from 0 to 3) and noise (from 0 to 300) for each shift. The red rectangle shows, as a reference, $\sigma = 0.6$.

4.2 Robustness to noise

One of the most important difficulties in shift estimation is related to the presence of noise on the input images. In this section, we evaluate the effects of noise on each method in a simulated environment. For each error value shown, an average of 100 experiments was performed, each time generating a new noise realization and using a different random subimage. Due to the randomness of the experiment, potential content-less images or aperture problem cases could exist. However, by using validation methods such as the Cramer-Rao lower bound, these cases were excluded from the results and the computed averages were based on valid cases only. Note also that by selecting random subimages, two experiments with injected noise sharing the same standard deviation could still have different SNRs. This was on purpose, to evaluate the versatility of each method under potentially plausible situations.

As proved by both Robinson [75] and Pham [64], the negative effect of noise is stressed as the displacement to estimate gets larger. For this reason, we evaluated the performance of every approach by varying noise and displacement magnitude conditions, based on the noise levels shown in Table 6. Figure 15 shows an example of two landscapes, the first with high photon count and the second with average photon count, under each simulated noise level. While in the first landscape, the effects of the noise start to be noted from the third noise level, for the second case it is already noticeable on the second level.

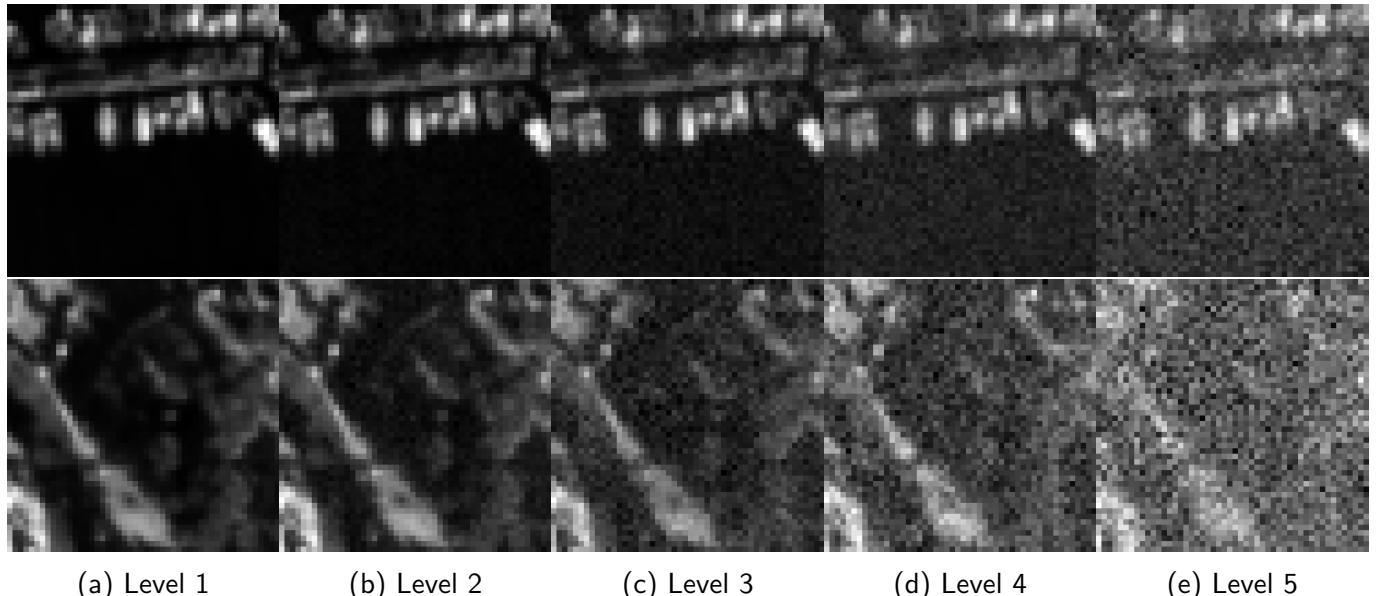


Figure 15: Sample images for the five analyzed noise levels. **Top:** high contrasted image (high SNR). **Bottom:** intermediate contrasted image (average SNR). Dynamic ranges extended for visualization purposes.

To summarize results, we display in Table 9 the top 10 methods together with their average errors for each evaluated shift magnitude category and every noise level analyzed. Also, the averages by noise level and by shift magnitude were included in the results. Since in general each top 10 is dominated by the different variants from a single method, this table does not seem too informative. However, some interesting conclusions could be inferred. First, the best methods are always gradient-based, and the amount of iterations and/or scales depends on the shift magnitude. For the lowest shift magnitude studied, using a single or two iterations of the original LS or the TLS method are optimal. For the second analyzed magnitude category, the best results were obtained without requiring a multiscale approach, however several iterations should be performed (three or four). Finally, for the remaining shift magnitudes, a multiscale approach is mandatory and the configuration that seems

to achieve the best results is to start with a single iteration on the broadest scale and increasing the amount of iterations by one for each subsequent scale progressively. Also, for the finest scale, the best interpolation scheme should be used (FFT with or without symmetrization or spline). Finally, for short shift-magnitudes, the gradients with the smallest supports are recommended. In particular, the hypomode stands out from the rest. As the shift magnitude increases, both 3×3 Farid or Simoncelli gradient estimation methods attain better scores under lower SNR scenarios. As a special case, under infinite SNR situations, larger gradient estimation kernels seem to achieve the best accuracies.

To give a better comparison between the studied methods, we took the best performing variant for each of them and compared them in the same manner. Therefore, the top 10 for the best variant of each method for each shift magnitude/noise case is displayed in Table 10. From this table several conclusions arise. First and most importantly, GBSE methods systematically improve over other approaches, and the difference in performance could be up to factor of five, and is more remarked on lower SNR scenarios. Secondly, for shifts magnitudes between 0.1 and 1.1, the bidirectional bias corrected method of Pham *et al.* makes the top 10 by using a single iteration and no resampling, yielding itself as the best candidate under limited computational time constraints. For shifts lower than 0.1, the total least squares method with one iteration is recommended for fast processing times. Third, gradient correlation approaches improve over phase-correlation-based methods, particularly under lower SNR where in general phase-correlation methods tend to fail. Finally, under low noise, apodization methods improved phase correlation approaches systematically and Hamming, Blackman or Tukey windows achieved the best results. However, as the noise gets higher, some methods achieved higher accuracies without apodization. This is because noise affects more the final accuracy than the fake edges generated by the false periodicity assumption. Therefore, having more pixels from which to estimate the shift, due to avoiding windowing the images in the spatial domain, helps reduce the noise influence, yielding improved results.

4.2.1 Predefined set of methods

While for practical reasons, it is interesting to know the top 10 methods for each case, we get no knowledge of the performance for each method for all situations. To this end, we preselected 13 methods, and evaluated their performance for every noise/shift magnitude case. Three fast single iteration methods were selected, namely the traditional least squares approach and the total least squares both using one iteration and the hypomode gradient estimation, together with the bidirectional bias correction method of Pham [63] using the 3×3 Farid gradient estimation. Also, both the least squares and the total least squares methods using four iterations with the same gradient estimation method were included in the comparison. As for GBSE multiscale approaches we included two methods. One using two scales, one iteration on the coarse scale and three on the original scale (3,1 iteration pattern), using spline interpolation on the last scale, and another method using three scales and a (3,2,1) iteration pattern for which the employed interpolation method in the final scale was FFT with symmetrization. Both methods used the 3×3 Farid gradient estimation, which proved before to be the most robust against noise. For phase-correlation methods, we included in the evaluation the approach proposed by Guizar-Sicairos [39] using an upsampling value of 2000, the method of Stone [86] and the Sinc fitting approach, both by applying a hamming window for apodization purposes, and the method of Hoge [43] with a Tukey window. Finally, two gradient correlation approaches were included, both using Gaussian derivatives with $\sigma = 0.6$. They are the original method [6] and the more recent approach by Tzimiropoulos [93].

In Table 11 we show the results for each predefined method on each condition sorted by accuracy. From this, we can observe the versatility of multiscale gradient-based approaches. Also, the bidirectional bias correction method (ULS) proved to work well in most cases, and it usually improves over other more expensive methods as the noise increases. Another thing to point out is the instability

	$\ \mathbf{v}\ \leq 0.1$	$0.1 < \ \mathbf{v}\ \leq 0.5$	$0.5 < \ \mathbf{v}\ \leq 1.1$	$\ \mathbf{v}\ > 1.1$	Avg.
σ	0.0000 MS-3,321-IdssGfa7	0.0000 MS-2,31-IdcGfa7	0.0000 MS-3,321-IdssGfa7	0.0000 MS-3,321-IdssGfa7	0.0000 MS-3,321-IdssGfa7
	0.0000 MS-2,21-IdcGfa7	0.0000 MS-2,21-IdcGfa7	0.0000 MS-2,31-IdcGfa7	0.0000 MS-3,321-IfssGfa7	0.0000 MS-2,31-IdcGfa7
	0.0000 MS-2,31-IdcGfa7	0.0000 MS-3,321-IdssGfa7	0.0000 MS-2,21-IdcGfa7	0.0000 MS-3,321-IdssGfa5	0.0000 MS-2,21-IdcGfa7
	0.0000 CLS-3-IdGfa7	0.0000 LS-3-IdGfa7	0.0000 LS-4-IdGfa7	0.0000 MS-3,321-IdssGsim5	0.0000 LS-4-IdGfa7
	0.0000 LS-3-IdGfa7	0.0000 TLS-3-IdGfa7	0.0000 TLS-4-IdGfa7	0.0000 MS-3,321-IfssGfa5	0.0000 TLS-4-IdGfa7
	0.0000 TLS-3-IdGfa7	0.0000 TLS-4-IdGfa7	0.0000 LS-3-IdGfa7	0.0000 MS-3,321-IfssGsim5	0.0000 LS-3-IdGfa7
	0.0000 TLS-4-IdGfa7	0.0000 LS-4-IdGfa7	0.0000 TLS-3-IdGfa7	0.0001 MS-3,321-IdssGg0.6	0.0000 TLS-3-IdGfa7
	0.0000 LS-4-IdGfa7	0.0000 MS-2,31-IdcGsim5	0.0000 MS-2,31-IfcGfa7	0.0005 MS-3,321-IsssGfa7	0.0000 MS-2,31-IfcGfa7
	0.0000 TLS-4-IdGfa7	0.0000 MS-2,21-IdcGsim5	0.0000 MS-2,21-IfcGfa7	0.0006 MS-3,321-IdssGch3	0.0000 MS-2,21-IfcGfa7
	0.0000 LS-2,21-IdcGfa7	0.0000 MS-3,321-IdcGfa5	0.0000 MS-3,321-IdssGsim5	0.0007 MS-3,321-IfssGg0.6	0.0000 MS-2,31-IdcGsim5
σ	0.0031 LS-2-IsGh	0.0035 CLS-4-IdGh	0.0035 MS-3,321-IdssGh	0.0044 MS-3,321-IfssGfa3	0.0034 TLS-4-IdGh
	0.0032 TLS-2-IsGh	0.0035 TLS-4-IdGh	0.0035 MS-3,321-IdssGch2	0.0045 MS-3,321-IdssGfa3	0.0034 MS-3,321-IdssGh
	0.0032 LS-2-IfGh	0.0035 LS-4-IdGh	0.0035 TLS-4-IdGh	0.0045 MS-3,321-IfssGsim3	0.0035 MS-2,31-IdcGh
	0.0032 CLS-2-IsGh	0.0036 MS-3,321-IdssGh	0.0035 MS-2,31-IdcGch2	0.0045 MS-3,321-IdssGsim3	0.0035 MS-3,321-IdssGch2
	0.0032 LS-2-IdGh	0.0036 CLS-3-IdGh	0.0035 MS-2,31-IdcGg0.3	0.0046 MS-3,321-IdssGg0.3	0.0035 MS-2,31-IdcGch2
	0.0032 TLS-2-IfGh	0.0037 MS-2,31-IdcGh	0.0035 MS-2,21-IdcGch2	0.0046 MS-3,321-IsssGfa3	0.0035 MS-2,21-IdcGch2
	0.0032 CLS-2-IfGh	0.0037 LS-3-IdGch2	0.0035 MS-3,321-IdssGg0.3	0.0046 MS-3,321-IdssGg0.6	0.0036 LS-4-IdGh
	0.0032 CLS-2-IdGh	0.0037 CLS-3-IdGch2	0.0035 MS-2,31-IdcGh	0.0047 MS-3,321-IsssGsim3	0.0036 LS-4-IdGh
	0.0032 TLS-2-IdGh	0.0037 LS-4-IdGch2	0.0035 LS-4-IdGch2	0.0048 MS-3,321-IfssGg0.6	0.0036 LS-4-IdGg0.3
	0.0032 LS-3-IdGh	0.0037 CLS-4-IdGch2	0.0036 LS-4-IdGg0.3	0.0050 MS-3,321-IsssGg0.6	0.0036 MS-3,321-IdssGg0.3
σ	0.0092 LS-2-IsGh	0.0105 TLS-4-IdGh	0.0118 MS-3,321-IdssGh	0.0220 MS-3,321-IfssGfa3	0.0113 TLS-4-IdGh
	0.0092 LS-2-IdGh	0.0105 CLS-4-IdGh	0.0118 MS-3,321-IfssGh	0.0221 MS-3,321-IdssGfa3	0.0113 MS-3,321-IdssGh
	0.0093 LS-2-IfGh	0.0106 TLS-4-IfGh	0.0121 MS-3,321-IsssGh	0.0221 MS-3,321-IfssGsim3	0.0114 TLS-4-IfGh
	0.0095 TLS-2-IdGh	0.0107 CLS-3-IdGh	0.0123 MS-3,321-IdssGch2	0.0221 MS-3,321-IdssGsim3	0.0114 MS-3,321-IfssGh
	0.0095 TLS-2-IfGh	0.0107 CLS-4-IfGh	0.0125 MS-3,321-IdssGg0.3	0.0222 MS-3,321-IsssGfa3	0.0116 MS-3,321-IdssGg0.3
	0.0095 TLS-2-IsGh	0.0107 TLS-3-IsGh	0.0126 MS-3,321-IfssGch2	0.0222 MS-3,321-IsssGsim3	0.0117 MS-3,321-IdssGch1
	0.0095 LS-2-IcGch1	0.0107 CLS-4-IdGg0.3	0.0127 MS-3,321-IdssGch1	0.0246 MS-3,321-IdssGg0.3	0.0117 MS-2,31-IdcGh
	0.0095 LS-3-IdGh	0.0107 TLS-3-IdGh	0.0127 MS-3,321-IsssGch2	0.0247 MS-3,321-IfssGg0.3	0.0117 MS-3,321-IsssGh
	0.0096 LS-2-IIGch1	0.0107 MS-3,321-IdssGg0.3	0.0127 MS-3,321-IfssGg0.3	0.0250 MS-3,321-IsssGg0.3	0.0117 MS-2,31-IdcGg0.3
	0.0096 CLS-2-IcGch1	0.0107 LS-4-IfGch2	0.0128 LS-4-IfGfa3	0.0269 MS-3,321-IdssGch2	0.0117 MS-2,31-IdcGch2
σ	0.0145 TLS-1-IIGh	0.0179 MS-3,321-IfssGh	0.0208 MS-3,321-IfssGh	0.0254 MS-3,321-IsssGsim3	0.0200 MS-3,321-IfssGh
	0.0147 LS-2-IdGh	0.0180 MS-3,321-IdssGh	0.0209 MS-3,321-IdssGh	0.0256 MS-3,321-IfssGsim3	0.0200 MS-3,321-IdssGh
	0.0148 LS-2-IfGh	0.0183 MS-3,321-IsssGh	0.0211 MS-3,321-IsssGh	0.0256 MS-3,321-IdssGsim3	0.0203 MS-3,321-IsssGh
	0.0148 LS-2-IsGh	0.0188 MS-2,31-IscGh	0.0220 MS-2,31-IdcGsim3	0.0258 MS-3,321-IsssGfa3	0.0209 MS-3,321-IdssGch1
	0.0149 LS-1-IIGg0.3	0.0189 MS-2,31-IdcGh	0.0220 MS-2,31-IfcGsim3	0.0261 MS-3,321-IfssGfa3	0.0209 MS-3,321-IfssGch1
	0.0150 LS-2-IsGch1	0.0190 MS-2,31-IfcGh	0.0220 MS-2,31-IfcGfa3	0.0261 MS-3,321-IdssGfa3	0.0211 MS-2,31-IfcGsim3
	0.0150 LS-2-IfGch1	0.0190 TLS-4-IsGh	0.0220 MS-2,31-IdcGfa3	0.0278 MS-3,321-IsssGg0.3	0.0211 MS-2,31-IdcGsim3
	0.0150 LS-2-IdGch1	0.0191 MS-3,321-IdssGch1	0.0220 MS-2,31-IscGsim3	0.0283 MS-3,321-IdssGg0.3	0.0212 MS-2,31-IfcGfa3
	0.0150 LS-2-IcGch1	0.0192 TLS-4-IdGh	0.0220 MS-3,321-IdssGch1	0.0284 MS-3,321-IfssGg0.3	0.0212 MS-2,31-IdcGfa3
	0.0150 LS-2-IIGch1	0.0193 MS-3,321-IfssGch1	0.0221 MS-2,31-IscGfa3	0.0330 MS-3,321-IdssGch2	0.0212 MS-2,31-IscGsim3
σ	0.0219 LS-1-IIGh	0.0476 LS-4-IsGfa3	0.0461 MS-2,31-IsGfa3	0.0687 MS-3,321-IfssGfa3	0.0475 LS-4-IfGfa3
	0.0225 TLS-1-IIGh	0.0476 LS-4-IfGfa3	0.0462 MS-2,31-IfcGfa3	0.0688 MS-3,321-IdssGfa3	0.0475 LS-4-IdGfa3
	0.0230 LS-1-IIGch1	0.0476 LS-4-IdGfa3	0.0462 MS-2,31-IdcGfa3	0.0689 MS-3,321-IsssGfa3	0.0476 MS-2,31-IfcGsim3
	0.0232 LS-2-IfGh	0.0477 LS-4-IIGsim3	0.0463 MS-2,31-IscGsim3	0.0706 MS-3,321-IfssGsim3	0.0476 MS-2,31-IdcGsim3
	0.0232 LS-2-IdGh	0.0477 LS-4-IsGsim3	0.0464 MS-2,31-IfcGsim3	0.0707 MS-3,321-IsssGsim3	0.0476 MS-2,31-IscGsim3
	0.0232 LS-2-IsGh	0.0478 LS-4-IIGfa3	0.0464 MS-2,31-IdcGsim3	0.0707 MS-3,321-IdssGsim3	0.0477 LS-4-IsGfa3
	0.0233 LS-2-IIGh	0.0478 LS-4-IfGsim3	0.0470 MS-3,321-IsssGfa3	0.0889 MS-3,321-IsssGfa5	0.0478 MS-2,31-IfcGfa3
	0.0234 LS-2-IcGh	0.0478 LS-4-IdGsim3	0.0471 MS-3,321-IsssGsim3	0.0891 MS-3,321-IfssGfa5	0.0478 MS-2,31-IdcGfa3
	0.0236 LS-1-IIGg0.3	0.0479 LS-4-IsGsim3	0.0472 MS-3,321-IfssGfa3	0.0891 MS-3,321-IdssGfa5	0.0478 MS-2,31-IscGfa3
	0.0241 TLS-1-IIGch1	0.0480 TLS-4-IIGsim3	0.0472 MS-3,321-IdssGfa3	0.0892 MS-3,321-IsssGsim3	0.0483 LS-4-IfGsim3
σ	0.0101 LS-2-IsGh	0.0167 LS-4-IdGfa3	0.0170 MS-2,31-IdcGfa3	0.0282 MS-3,321-IfssGfa3	0.0170 LS-4-IdGfa3
	0.0102 LS-2-IdGh	0.0168 LS-4-IdGsim3	0.0170 MS-2,31-IdcGsim3	0.0282 MS-3,321-IdssGfa3	0.0170 MS-2,31-IdcGsim3
	0.0102 LS-2-IfGh	0.0168 LS-4-IfGfa3	0.0170 MS-2,31-IfcGfa3	0.0284 MS-3,321-IfssGsim3	0.0170 MS-2,31-IfcGsim3
	0.0110 LS-3-IdGh	0.0168 LS-4-IfGsim3	0.0170 MS-2,31-IfcGsim3	0.0284 MS-3,321-IdssGsim3	0.0170 LS-4-IfGfa3
	0.0110 LS-3-IfGh	0.0171 LS-3-IdGfa3	0.0172 MS-3,321-IdssGh	0.0286 MS-3,321-IsssGfa3	0.0171 MS-2,31-IdcGfa3
	0.0110 LS-2-IIGh	0.0171 MS-2,31-IfcGsim3	0.0173 MS-2,31-IscGsim3	0.0289 MS-3,321-IsssGsim3	0.0171 MS-2,31-IfcGfa3
	0.0111 LS-2-IcGh	0.0171 MS-2,31-IdcGsim3	0.0174 MS-3,321-IdssGsim3	0.0331 MS-3,321-IsssGsim5	0.0171 LS-4-IdGsim3
	0.0111 TLS-1-IIGh	0.0172 MS-2,31-IdcGh	0.0174 MS-3,321-IdssGfa3	0.0331 MS-3,321-IdssGsim5	0.0172 LS-4-IfGsim3
	0.0111 LS-2-IcGch1	0.0172 LS-3-IfGfa3	0.0174 MS-2,31-IscGfa3	0.0332 MS-3,321-IfssGsim5	0.0172 MS-2,21-IdcGsim3
	0.0112 LS-3-IsGh	0.0172 MS-2,31-IdcGg0.3	0.0174 MS-3,321-IfssGsim3	0.0332 MS-3,321-IsssGfa5	0.0173 MS-2,21-IfcGsim3
Avg.					

Table 9: Average error per shift and magnitude of top 10 evaluated methods using all variants. **Rows:** five noise levels. **Columns:** four shift magnitudes. Note that the last column displays the top ten methods of the average for each noise level excluding the last shift magnitude level.

	$\ \mathbf{v}\ \leq 0.1$	$0.1 < \ \mathbf{v}\ \leq 0.5$	$0.5 < \ \mathbf{v}\ \leq 1.1$	$\ \mathbf{v}\ > 1.1$	Avg.
σ 0.000	0.0000 MS-3,321-IdssGfa7	0.0000 MS-2,31-IdcGfa7	0.0000 MS-3,321-IdssGfa7	0.0000 MS-3,321-IdssGfa7	0.0000 MS-3,321-IdssGfa7
	0.0000 CLS-3-IdGfa7	0.0000 LS-3-IdGfa7	0.0000 LS-4-IdGfa7	0.0196 PC-SINC-Whw	0.0000 LS-4-IdGfa7
	0.0000 LS-3-IdGfa7	0.0000 TLS-3-IdGfa7	0.0000 TLS-4-IdGfa7	0.0231 SS-HOGE-Wbl	0.0000 TLS-4-IdGfa7
	0.0000 TLS-3-IdGfa7	0.0000 CLS-4-IdGfa5	0.0011 CLS-4-IdGg0.6	0.0835 GC04-Gch3	0.0004 CLS-4-IdGg0.6
	0.0009 GC11-Gch3	0.0040 SS-HOGE-Wbw	0.0068 SS-HOGE-Wbw	0.1340 GC11-Gch3	0.0044 SS-HOGE-Wbw
	0.0012 PCSTONE-Whw	0.0047 PC-SINC-Wbw	0.0083 PC-SINC-Wbw	0.4176 SDF-2QI	0.0049 PC-SINC-Wbw
	0.0012 SS-ROBINSON-Wnw	0.0119 GC11-Gch3	0.0276 GC11-Gch3	0.4607 ADF-2QI	0.0135 GC11-Gch3
	0.0014 PC-SINC-Wtw	0.0126 PCSTONE-Whw	0.0311 PCSTONE-Whw	0.5105 ADF2-2QI	0.0150 PCSTONE-Whw
	0.0016 NGC11-G5	0.0318 ULS-Gfa7	0.0476 GC04-G6	0.5868 CLS-4-IdGfa7	0.0324 ULS-Gfa7
	0.0038 INT-3	0.0399 INT-3	0.0576 ULS-Gg0.3	0.5933 PCFOO	0.0333 GC04-Gg1
σ 0.005	0.0031 LS-2-IsGh	0.0035 CLS-4-IdGh	0.0035 MS-3,321-IdssGh	0.0044 MS-3,321-IfssGfa3	0.0034 TLS-4-IdGh
	0.0031 MS-2-LS2-IsGh	0.0035 TLS-4-IdGh	0.0035 TLS-4-IdGh	0.0458 SS-HOGE-Whw	0.0034 MS-3,321-IdssGh
	0.0032 TLS-2-IsGh	0.0035 LS-4-IdGh	0.0035 LS-4-IdGch2	0.0477 PC-SINC-Whw	0.0036 LS-4-IdGch2
	0.0032 CLS-2-IsGh	0.0035 MS-2-LS4-IdGh	0.0060 CLS-4-IdGg0.6	0.0895 GC04-Gch3	0.0047 CLS-4-IdGg0.6
	0.0063 PC-GUIZAR-100	0.0159 GC11-Gch3	0.0225 SS-HOGE-Wtw	0.1545 GC11-Gch3	0.0192 GC11-Gch3
	0.0063 GC11-Gch2	0.0189 PCSTONE-Whw	0.0277 PC-SINC-Wtw	0.6072 CLS-4-IdGfa7	0.0210 SS-HOGE-Wtw
	0.0063 INT-3	0.0199 SS-HOGE-Wtw	0.0355 GC11-Gch3	0.6188 PCFOO	0.0227 PCSTONE-Whw
	0.0074 GC04-Gg1	0.0301 ULS-Gfa7	0.0367 PCSTONE-Whw	1.3427 APC-2It	0.0249 PC-SINC-Wtw
	0.0082 NGC04-Gg1	0.0304 PC-SINC-Wtw	0.0558 GC04-G6	1.8574 ACC-3It	0.0327 ULS-Gsim5
	0.0082 PCSTONE-Wnw	0.0372 INT-3	0.0564 ULS-Gg0.3	1.9387 INT-5	0.0376 GC04-Gg1
σ 0.015	0.0092 LS-2-IsGh	0.0105 TLS-4-IdGh	0.0118 MS-3,321-IdssGh	0.0220 MS-3,321-IfssGfa3	0.0113 TLS-4-IdGh
	0.0092 MS-2-LS2-IsGh	0.0105 CLS-4-IdGh	0.0128 LS-4-IfGfa3	0.0886 GC04-Gch2	0.0113 MS-3,321-IdssGh
	0.0095 TLS-2-IdGh	0.0107 MS-3,321-IdssGg0.3	0.0135 TLS-4-IfGh	0.1002 PC-ESINC-Wtw	0.0121 LS-4-IdGfa3
	0.0096 CLS-2-IdGch1	0.0107 LS-4-IfGch2	0.0198 CLS-4-IfGg0.6	0.1300 SS-HOGE-Wtw	0.0146 CLS-4-IdGg0.6
	0.0113 PC-GUIZAR-2000	0.0266 GC11-Gg0.3	0.0546 GC11-Gg0.6	0.1893 GC11-Gch3	0.0337 GC11-Gg0.6
	0.0136 GC04-G6	0.0400 PC-REN2010-Wex	0.0593 ULS-Gch1	0.5603 PCFOO	0.0406 ULS-Gfa3
	0.0160 PCSTONE-Wex	0.0419 PCSTONE-Wtw	0.0599 PCSTONE-Whw	0.6411 CLS-4-IdGfa7	0.0430 GC04-Gg0.6
	0.0171 ULS-Gfa3	0.0419 ULS-Gfa5	0.0634 GC04-Gg0.6	1.8438 ACC-3It	0.0479 PCSTONE-Whw
	0.0172 INT-3	0.0432 INT-3	0.0792 SS-HOGE-Wtw	2.0047 INT-5	0.0624 SS-HOGE-Wnw
	0.0191 GC11-Gg0.3	0.0490 SS-HOGE-Wnw	0.0911 PC-ESINC-Wtw	2.1902 PCSTONE-Wbw	0.0662 INT-3
σ 0.025	0.0145 TLS-1-IlGh	0.0179 MS-3,321-IfssGh	0.0208 MS-3,321-IfssGh	0.0254 MS-3,321-IsrssGsim3	0.0200 MS-3,321-IfssGh
	0.0147 LS-2-IdGh	0.0190 TLS-4-IsGh	0.0237 LS-4-IdGfa3	0.1044 GC04-Gch3	0.0214 LS-4-IdGfa3
	0.0147 MS-2-LS2-IdGh	0.0196 CLS-4-IdGg0.3	0.0273 TLS-4-IfGh	0.1051 PC-ESINC-Wnw	0.0215 TLS-4-IdGh
	0.0153 CLS-2-IfGch1	0.0201 LS-4-IfGsim3	0.0334 CLS-4-IfGg1	0.1694 GC11-Gch2	0.0262 CLS-4-IdGg0.6
	0.0196 PC-GUIZAR-2000	0.0437 ULS-Gfa5	0.0715 ULS-Gsim3	0.2217 SS-HOGE-Wtw	0.0475 ULS-Gfa3
	0.0225 PCSTONE-Wex	0.0509 PC-REN2010-Wex	0.0793 GC11-G6	0.6126 PCFOO	0.0606 GC04-Gg0.6
	0.0268 GC04-Gg1	0.0559 GC11-Gg0.6	0.0806 GC04-Gg1	0.6499 CLS-4-IdGfa7	0.0649 PCSTONE-Wtw
	0.0268 ULS-Gfa3	0.0562 PCSTONE-Wtw	0.0915 PCSTONE-Wtw	2.0982 ACC-3It	0.0678 GC11-Gg0.6
	0.0311 INT-3	0.0598 INT-3	0.1292 PC-ESINC-Wtw	2.1405 INT-5	0.0794 PC-GUIZAR-2000
	0.0316 SS-HOGE-Wex	0.0670 GC04-Gg0.6	0.1303 SS-HOGE-Wnw	2.3039 PCSTONE-Wnw	0.0856 SS-HOGE-Wex
σ 0.055	0.0219 LS-1-IlGh	0.0476 LS-4-IsGfa3	0.0461 MS-2,31-IsGfa3	0.0687 MS-3,321-IfssGfa3	0.0475 LS-4-IfGfa3
	0.0219 MS-2-LS1-IlGh	0.0476 MS-2-LS4-IsGfa3	0.0519 LS-4-IfGsim5	0.2764 GC04-Gg0.6	0.0475 MS-2-LS4-IfGfa3
	0.0225 TLS-1-IlGh	0.0480 TLS-4-IlGsim3	0.0562 CLS-4-IdGfa5	0.3321 PC-GUIZAR-1000	0.0546 CLS-4-IdGfa5
	0.0241 CLS-1-IlGch1	0.0574 CLS-4-IdGfa5	0.0791 ULS-Gh	0.3879 GC11-Gg1	0.0622 TLS-4-IfGh
	0.0329 PCSTONE-Wex	0.0621 ULS-Gsim3	0.0835 GC04-Gg1	0.6946 SS-HOGE-Wnw	0.0662 ULS-Gsim3
	0.0483 INT-3	0.0996 PC-GUIZAR-100	0.0954 TLS-4-IfGh	0.9116 PCFOO	0.0834 GC04-Gg1
	0.0523 PC-SINC-Wex	0.1011 PCSTONE-Wnw	0.0955 GC11-Gg1	1.0141 CLS-4-IdGfa7	0.0984 GC11-Gg1
	0.0556 GC04-Gg1	0.1110 GC04-Gg1	0.1341 PCSTONE-Wtw	2.2279 ACC-3It	0.1016 PC-GUIZAR-1000
	0.0560 ULS-Gfa3	0.1157 GC11-Gg1	0.1495 PC-GUIZAR-1000	2.2333 LS-1-IlGfa7	0.1044 PCSTONE-Wnw
	0.0740 SS-HOGE-Wex	0.1238 INT-3	0.2533 SS-HOGE-Wex	2.4070 PCSTONE-Whw	0.1510 SS-HOGE-Wex
Avg.	0.0101 LS-2-IsGh	0.0167 LS-4-IdGfa3	0.0170 MS-2,31-IdcGfa3	0.0282 MS-3,321-IfssGfa3	0.0170 LS-4-IdGfa3
	0.0101 MS-2-LS2-IsGh	0.0167 MS-2-LS4-IdGfa3	0.0186 LS-4-IdGfa3	0.1467 GC04-Gg0.3	0.0170 MS-2-LS4-IdGfa3
	0.0111 TLS-1-IlGh	0.0180 TLS-4-IdGh	0.0265 CLS-4-IdGg1	0.1614 PC-SINC-Wtw	0.0198 TLS-4-IdGh
	0.0117 CLS-2-IdGch1	0.0190 CLS-4-IdGg0.3	0.0281 TLS-4-IdGh	0.2319 SS-HOGE-Wtw	0.0214 CLS-4-IdGg0.6
	0.0181 PCSTONE-Wex	0.0434 ULS-Gfa5	0.0632 GC11-Gg0.6	0.2373 GC11-Gch1	0.0444 ULS-Gfa3
	0.0198 PC-GUIZAR-100	0.0524 PCSTONE-Wtw	0.0658 ULS-Gsim3	0.6593 PCFOO	0.0520 GC11-Gg1
	0.0214 INT-3	0.0544 GC11-Gg1	0.0685 GC04-Gg1	0.6998 CLS-4-IdGfa7	0.0538 GC04-Gg1
	0.0218 GC04-Gg1	0.0550 PC-REN2010-Wex	0.0773 PCSTONE-Whw	1.9667 ACC-3It	0.0581 PCSTONE-Wtw
	0.0232 ULS-Gfa3	0.0608 INT-3	0.1095 PC-SINC-Wtw	2.0953 INT-5	0.0795 PC-GUIZAR-2000
	0.0283 SS-HOGE-Wex	0.0676 SS-HOGE-Wex	0.1280 SS-HOGE-Wnw	2.2549 PCSTONE-Whw	0.0802 SS-HOGE-Wex

Table 10: Average error per shift and magnitude of top 10 evaluated methods using the best variants for each approach. **Rows:** five noise levels. **Columns:** four shift magnitudes. Note that the last column displays the top ten methods of the average for each noise level excluding the last shift magnitude level.

of the iterative total least squares method, particularly under low SNR conditions. Although this method usually achieves more accurate results than the original least squares approach, its usage must be verified [95, 91] under uncontrolled scenarios. Also, the poor tolerance to noise of phase-correlation-based methods was made evident from these results. Nevertheless, the Guizar-Sicairos approach seems to be less affected by upsampling the phase-correlation surface around its peak. Finally, as expected, every non-multiscale GBSE method failed when shift magnitudes are larger than 1.1.

To get a better understanding of the robustness of the methods against noise or shift magnitude, Fig. 16 shows how the error evolves by varying both noise or shift magnitude categories. To be fair, shifts with magnitude larger than 1.1 were excluded from both figures. Once again, the improved accuracy of the multiscale GBSE approach is evidenced in both tests, and improves over using a non-multiscale approach when the shifts get larger [68]. Also, the least squares approach with one iteration is an excellent candidate when the shift magnitude is short or the SNR is high. When this is not the case and a fast method is required, the bidirectional bias correction approach proved to be the best candidate. Once again, gradient correlation methods were more accurate than phase correlation-based approaches, particularly as the noise increases.

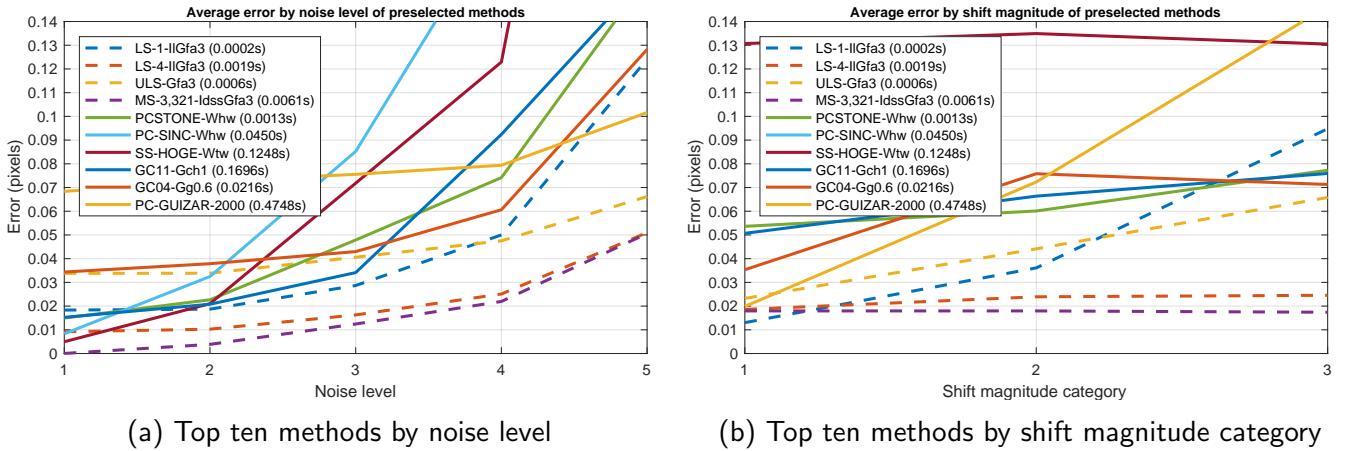


Figure 16: Average error for some variants of preselected methods. **Left:** varying noise averaging over all shift magnitudes. **Right:** varying shift magnitude averaging over all noise levels.

4.2.2 A short study on multiscale gradient-based shift estimation

We studied whether it is more convenient to iterate the standard GBSE approach in a direct fashion, i.e. using only one scale, or in a multiscale approach. To this end, both methodologies described in section 2.3 were evaluated under different noise conditions, shifts and gradient estimators. To show the most representative results, four SNR conditions were considered: noiseless, low noise ($\sigma = 75$), medium noise ($\sigma = 150$) and high noise ($\sigma = 300$). Noise values are according to 12 bit images. Each table is organized in groups of four lines corresponding to each of these four noise configurations. Also, the four most significative shifts in terms of results are shown: a big shift $(0.5, -0.9)$, a medium shift $(0.2, -0.2)$, a small shift $(0.024, 0.052)$ and no shift.

The performance of each algorithm under each condition was evaluated by simulation using the same setup employed throughout this section. The results shown contain only valid shift estimations.

In tables 12 and 13 results are shown for two iterations and bicubic interpolation, and for three iterations with spline interpolation respectively. From these results several conclusions can be drawn.

- First, as expected, the multiscale method is much more robust when the shift magnitude is

		$\ \mathbf{v}\ \leq 0.1$	$0.1 < \ \mathbf{v}\ \leq 0.5$	$0.5 < \ \mathbf{v}\ \leq 1.1$	$\ \mathbf{v}\ > 1.1$
σ	0.0000	MS-3,321-IdssGfa3	0.0000 MS-3,321-IdssGfa3	0.0001 MS-3,321-IdssGfa3	0.0196 PC-SINC-Whw
	0.0003	MS-2,31-IscGfa3	0.0017 MS-2,31-IscGfa3	0.0016 MS-2,31-IscGfa3	0.0197 MS-3,321-IdssGfa3
	0.0011	GC11-Gg0.6	0.0044 SS-HOGE-Wtw	0.0085 SS-HOGE-Wtw	0.0317 SS-HOGE-Wtw
	0.0012	PCSTONE-Whw	0.0087 PC-SINC-Whw	0.0097 TLS-4-IIGfa3	0.0911 GC04-Gg0.6
	0.0018	PC-SINC-Whw	0.0126 PCSTONE-Whw	0.0097 LS-4-IIGfa3	0.1756 GC11-Gg0.6
	0.0020	SS-HOGE-Wtw	0.0140 GC11-Gg0.6	0.0145 PC-SINC-Whw	0.2735 PC-GUIZAR-2000
	0.0035	LS-4-IIGfa3	0.0143 TLS-4-IIGfa3	0.0311 PCSTONE-Whw	0.7801 MS-2,31-IscGfa3
	0.0035	TLS-4-IIGfa3	0.0143 LS-4-IIGfa3	0.0364 GC11-Gg0.6	2.1838 PCSTONE-Whw
	0.0036	TLS-1-IIGh	0.0363 ULS-Gfa3	0.0482 GC04-Gg0.6	15.7022 ULS-Gfa3
	0.0036	LS-1-IIGh	0.0388 TLS-1-IIGh	0.0578 ULS-Gfa3	Inf LS-1-IIGh
	0.0059	GC04-Gg0.6	0.0403 LS-1-IIGh	0.1174 TLS-1-IIGh	Inf TLS-1-IIGh
	0.0061	PC-GUIZAR-2000	0.0490 GC04-Gg0.6	0.1364 PC-GUIZAR-2000	Inf LS-4-IIGfa3
	0.0073	ULS-Gfa3	0.0627 PC-GUIZAR-2000	0.1708 LS-1-IIGh	Inf TLS-4-IIGfa3
σ	0.0037	MS-3,321-IdssGfa3	0.0040 MS-3,321-IdssGfa3	0.0039 MS-3,321-IdssGfa3	0.0045 MS-3,321-IdssGfa3
	0.0038	MS-2,31-IscGfa3	0.0042 MS-2,31-IscGfa3	0.0042 MS-2,31-IscGfa3	0.0477 PC-SINC-Whw
	0.0050	TLS-1-IIGh	0.0144 TLS-4-IIGfa3	0.0102 LS-4-IIGfa3	0.0552 SS-HOGE-Wtw
	0.0052	LS-1-IIGh	0.0144 LS-4-IIGfa3	0.0104 TLS-4-IIGfa3	0.0981 GC04-Gg0.6
	0.0062	TLS-4-IIGfa3	0.0180 GC11-Gg0.6	0.0225 SS-HOGE-Wtw	0.1881 GC11-Gg0.6
	0.0062	LS-4-IIGfa3	0.0189 PCSTONE-Whw	0.0367 PCSTONE-Whw	0.2415 PC-GUIZAR-2000
	0.0065	GC11-Gg0.6	0.0199 SS-HOGE-Wtw	0.0432 PC-SINC-Whw	2.1241 PCSTONE-Whw
	0.0068	PC-GUIZAR-2000	0.0345 PC-SINC-Whw	0.0434 GC11-Gg0.6	2.6144 LS-1-IIGh
	0.0077	GC04-Gg0.6	0.0350 ULS-Gfa3	0.0564 GC04-Gg0.6	13.5597 ULS-Gfa3
	0.0090	ULS-Gfa3	0.0411 TLS-1-IIGh	0.0575 ULS-Gfa3	Inf TLS-1-IIGh
	0.0124	PCSTONE-Whw	0.0449 LS-1-IIGh	0.1203 TLS-1-IIGh	Inf LS-4-IIGfa3
	0.0197	PC-SINC-Whw	0.0495 GC04-Gg0.6	0.1439 PC-GUIZAR-2000	Inf TLS-4-IIGfa3
	0.0207	SS-HOGE-Wtw	0.0671 PC-GUIZAR-2000	0.1799 LS-1-IIGh	Inf MS-2,31-IscGfa3
σ	0.0099	TLS-1-IIGh	0.0121 MS-3,321-IdssGfa3	0.0130 MS-2,31-IscGfa3	0.0221 MS-3,321-IdssGfa3
	0.0104	LS-1-IIGh	0.0123 MS-2,31-IscGfa3	0.0131 MS-3,321-IdssGfa3	0.1171 GC04-Gg0.6
	0.0113	PC-GUIZAR-2000	0.0185 LS-4-IIGfa3	0.0167 LS-4-IIGfa3	0.1300 SS-HOGE-Wtw
	0.0121	MS-2,31-IscGfa3	0.0185 TLS-4-IIGfa3	0.0215 TLS-4-IIGfa3	0.1764 PC-SINC-Whw
	0.0121	MS-3,321-IdssGfa3	0.0272 GC11-Gg0.6	0.0546 GC11-Gg0.6	0.2345 GC11-Gg0.6
	0.0136	TLS-4-IIGfa3	0.0431 ULS-Gfa3	0.0599 PCSTONE-Whw	0.2796 PC-GUIZAR-2000
	0.0136	LS-4-IIGfa3	0.0432 PCSTONE-Whw	0.0616 ULS-Gfa3	0.6751 MS-2,31-IscGfa3
	0.0137	GC04-Gg0.6	0.0519 GC04-Gg0.6	0.0634 GC04-Gg0.6	2.2229 PCSTONE-Whw
	0.0171	ULS-Gfa3	0.0534 TLS-1-IIGh	0.0792 SS-HOGE-Wtw	2.6654 LS-1-IIGh
	0.0194	GC11-Gg0.6	0.0627 PC-GUIZAR-2000	0.1115 PC-SINC-Whw	17.7786 ULS-Gfa3
	0.0405	PCSTONE-Whw	0.0689 LS-1-IIGh	0.1527 PC-GUIZAR-2000	Inf TLS-1-IIGh
	0.0645	SS-HOGE-Wtw	0.0700 PC-SINC-Whw	0.1541 TLS-1-IIGh	Inf LS-4-IIGfa3
	0.0742	PC-SINC-Whw	0.0718 SS-HOGE-Wtw	0.2446 LS-1-IIGh	Inf TLS-4-IIGfa3
σ	0.0145	TLS-1-IIGh	0.0199 MS-2,31-IscGfa3	0.0221 MS-2,31-IscGfa3	0.0261 MS-3,321-IdssGfa3
	0.0150	LS-1-IIGh	0.0203 MS-3,321-IdssGfa3	0.0229 MS-3,321-IdssGfa3	0.2217 SS-HOGE-Wtw
	0.0196	PC-GUIZAR-2000	0.0246 LS-4-IIGfa3	0.0274 LS-4-IIGfa3	0.2261 PC-SINC-Whw
	0.0218	MS-2,31-IscGfa3	0.0267 TLS-4-IIGfa3	0.0716 ULS-Gfa3	0.2354 GC11-Gg0.6
	0.0227	MS-3,321-IdssGfa3	0.0442 ULS-Gfa3	0.0845 GC04-Gg0.6	0.2685 PC-GUIZAR-2000
	0.0234	LS-4-IIGfa3	0.0559 GC11-Gg0.6	0.0851 GC11-Gg0.6	0.6882 MS-2,31-IscGfa3
	0.0268	ULS-Gfa3	0.0627 PCSTONE-Whw	0.0924 PCSTONE-Whw	2.3369 PCSTONE-Whw
	0.0304	GC04-Gg0.6	0.0670 GC04-Gg0.6	0.1439 SS-HOGE-Wtw	2.7269 LS-1-IIGh
	0.0623	GC11-Gg0.6	0.0694 PC-GUIZAR-2000	0.1492 PC-GUIZAR-2000	42.2713 ULS-Gfa3
	0.0674	PCSTONE-Whw	0.0773 TLS-1-IIGh	0.2029 TLS-1-IIGh	Inf TLS-1-IIGh
	0.1206	SS-HOGE-Wtw	0.1037 LS-1-IIGh	0.2205 PC-SINC-Whw	Inf LS-4-IIGfa3
	0.1666	PC-SINC-Whw	0.1043 SS-HOGE-Wtw	0.3185 LS-1-IIGh	Inf TLS-4-IIGfa3
	Inf TLS-4-IIGfa3	Inf TLS-4-IIGfa3	Inf TLS-4-IIGfa3	Inf GC04-Gg0.6	Inf GC04-Gg0.6
σ	0.0219	LS-1-IIGh	0.0478 LS-4-IIGfa3	0.0461 MS-2,31-IscGfa3	0.0688 MS-3,321-IdssGfa3
	0.0225	TLS-1-IIGh	0.0498 MS-2,31-IscGfa3	0.0472 MS-3,321-IdssGfa3	0.2764 GC04-Gg0.6
	0.0463	LS-4-IIGfa3	0.0534 MS-3,321-IdssGfa3	0.0589 LS-4-IIGfa3	0.3321 PC-GUIZAR-2000
	0.0475	MS-2,31-IscGfa3	0.0621 ULS-Gfa3	0.0804 ULS-Gfa3	0.4214 GC11-Gg0.6
	0.0512	MS-3,321-IdssGfa3	0.0998 PC-GUIZAR-2000	0.0964 GC11-Gg0.6	0.7207 SS-HOGE-Wtw
	0.0555	PC-GUIZAR-2000	0.1451 TLS-1-IIGh	0.1039 GC04-Gg0.6	0.7655 PC-SINC-Whw
	0.0560	ULS-Gfa3	0.1617 GC04-Gg0.6	0.1495 PC-GUIZAR-2000	0.7991 MS-2,31-IscGfa3
	0.1191	GC04-Gg0.6	0.1631 PCSTONE-Whw	0.1662 PCSTONE-Whw	2.4070 PCSTONE-Whw
	0.1468	PCSTONE-Whw	0.1669 GC11-Gg0.6	0.3531 TLS-1-IIGh	2.8674 LS-1-IIGh
	0.4459	SS-HOGE-Wtw	0.1783 LS-1-IIGh	0.3983 SS-HOGE-Wtw	8.1707 ULS-Gfa3
	0.5200	PC-SINC-Whw	0.4742 SS-HOGE-Wtw	0.4705 LS-1-IIGh	Inf TLS-1-IIGh
	Inf TLS-4-IIGfa3	0.5693 PC-SINC-Whw	0.5198 PC-SINC-Whw	Inf LS-4-IIGfa3	Inf TLS-4-IIGfa3
	Inf GC11-Gg0.6	Inf TLS-4-IIGfa3	Inf TLS-4-IIGfa3	Inf TLS-4-IIGfa3	Inf TLS-4-IIGfa3

Table 11: Average error of preselected methods per shift and magnitude. **Rows:** five noise levels. **Columns:** four shift magnitudes.

Shift (px)	Noise	IT2Gh	IT2Gg1	IT2Gg0.3	MS2Gh	MS2Gg1	MS2Gg0.3
(0.5000,-0.9000)	$\sigma = 0$	0.0514	0.0818	0.0472	0.0387	0.1600	0.0316
	$\sigma = 75$	0.1375	0.1053	0.1103	0.0744	0.1808	0.0582
	$\sigma = 150$	0.2875	0.1414	0.2305	0.1267	0.2130	0.1009
	$\sigma = 300$	0.4927	0.2327	0.4292	0.2319	0.2872	0.1909
(0.0000,0.0000)	$\sigma = 0$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	$\sigma = 75$	0.0114	0.0188	0.0132	0.0164	0.0188	0.0175
	$\sigma = 150$	0.0168	0.0330	0.0219	0.0300	0.0326	0.0313
	$\sigma = 300$	0.0191	0.0539	0.0307	0.0527	0.0549	0.0573
(0.2000,-0.2000)	$\sigma = 0$	0.0115	0.0117	0.0154	0.0115	0.0360	0.0159
	$\sigma = 75$	0.0249	0.0267	0.0223	0.0192	0.0470	0.0225
	$\sigma = 150$	0.0652	0.0424	0.0487	0.0360	0.0591	0.0358
	$\sigma = 300$	0.1295	0.0765	0.1103	0.0738	0.0899	0.0694
(0.0240,0.0520)	$\sigma = 0$	0.0040	0.0019	0.0052	0.0039	0.0073	0.0054
	$\sigma = 75$	0.0122	0.0181	0.0138	0.0156	0.0189	0.0166
	$\sigma = 150$	0.0198	0.0326	0.0231	0.0296	0.0326	0.0311
	$\sigma = 300$	0.0300	0.0543	0.0354	0.0547	0.0564	0.0581
Avg.	$\sigma = 0$	0.0167	0.0238	0.0170	0.0135	0.0508	0.0132
	$\sigma = 75$	0.0465	0.0422	0.0399	0.0314	0.0664	0.0287
	$\sigma = 150$	0.0973	0.0624	0.0811	0.0556	0.0843	0.0498
	$\sigma = 300$	0.1678	0.1043	0.1514	0.1033	0.1221	0.0939

Table 12: Estimation error (in pixels) per shift of every method using **two iterations** and **bicubic** interpolation from valid estimations. For each shift and estimation method, four SNR conditions were tested. The first three columns are for the iterative method (IT) while the last three are for the multiscale approach (MS) with a single iteration per scale. In each case, three gradient estimation methods were used: backward difference and Gaussian derivative with $\sigma = 1$ and with $\sigma = 0.3$ respectively.

- high. In fact, even at a shift as low as (0.2,-0.2) it is recommendable to use the multiscale method instead of the standard iterative version.
- Second, when no shift or a small shift is present, the single-scale methods achieve much better accuracies. Apparently, the multiscale algorithms are not suited for such small shifts since their poor performance on lower scales results in less accurate results. This result contradicts several state-of-the-art methods and is worth remarking.
 - Third, regarding the amount of iterations/scales to use, in presence of high noise, performing more iterations in the original scale or using more scales in the multiscale approach gives worse results in terms of accuracy. When dealing with a noisy situation, the resampling operation proved to be negative for the shift estimation algorithm. This result is more accentuated for the multiscale approach.
 - Finally, the multiscale algorithm proved to be a better contender when dealing with noise in general, although this factor is greatly influenced by the shift magnitude. However, except when the shift magnitude is lower than a fifth of a pixel, its use is recommended. Moreover, its computational cost is lower than the iterative counterpart since the resampling is performed on lower resolution images.

Shift (px)	Noise	IT3Gh	IT3Gg1	IT3Gg0.3	MS3Gh	MS3Gg1	MS3Gg0.3
(0.5000,-0.9000)	$\sigma = 0$	0.0156	0.0238	0.0065	0.0114	0.1007	0.0086
	$\sigma = 75$	0.0646	0.0437	0.0437	0.0293	0.1194	0.0251
	$\sigma = 150$	0.1869	0.0727	0.1326	0.0533	0.1488	0.0497
	$\sigma = 300$	0.4092	0.1480	0.3337	0.1093	0.2143	0.1001
(0.0000,0.0000)	$\sigma = 0$	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	$\sigma = 75$	0.0116	0.0199	0.0133	0.0261	0.0330	0.0260
	$\sigma = 150$	0.0194	0.0357	0.0246	0.0468	0.0547	0.0467
	$\sigma = 300$	0.0250	0.0621	0.0391	0.0980	0.1065	0.1001
(0.2000,-0.2000)	$\sigma = 0$	0.0027	0.0045	0.0043	0.0206	0.0205	0.0235
	$\sigma = 75$	0.0201	0.0229	0.0182	0.0304	0.0430	0.0334
	$\sigma = 150$	0.0475	0.0395	0.0373	0.0483	0.0593	0.0489
	$\sigma = 300$	0.1096	0.0718	0.0913	0.0982	0.1195	0.0977
(0.0240,0.0520)	$\sigma = 0$	0.0007	0.0009	0.0013	0.0068	0.0041	0.0080
	$\sigma = 75$	0.0120	0.0188	0.0132	0.0219	0.0283	0.0225
	$\sigma = 150$	0.0207	0.0348	0.0248	0.0471	0.0548	0.0464
	$\sigma = 300$	0.0313	0.0622	0.0412	0.1011	0.1056	0.1001
Avg.	$\sigma = 0$	0.0047	0.0073	0.0030	0.0097	0.0313	0.0100
	$\sigma = 75$	0.0271	0.0263	0.0221	0.0269	0.0559	0.0268
	$\sigma = 150$	0.0686	0.0457	0.0548	0.0489	0.0794	0.0479
	$\sigma = 300$	0.1438	0.0860	0.1263	0.1017	0.1365	0.0995

Table 13: Estimation error (in pixels) per shift of every method using **three iterations** and **spline** interpolation from valid estimations. Table configuration is the same as in Table 12.

4.2.3 Conclusions

To summarize, the conclusions for this section are:

- The **best overall method** proved to be the **multiscale GBSE** approach using three scales. The best iteration/resampling configuration on the scales was to do a single iteration on the coarsest scale, followed by performing two iterations in the middle scale using spline interpolation, and applying three iterations on the finest (original) scale using FFT with symmetrization as the interpolation method.
- The best method type in general is gradient-based, and **the amount of iterations and/or scales depends on the shift magnitude to estimate**.
- The **derivative** for GBSE methods that proved to be the **most robust to noise** was the one proposed by **Farid [29] using a 3×3 support**, although the 3×3 variant of the Simoncelli derivative [82] achieved similar results.
- For **low shift magnitudes** (category 1 in Table 7) the best methods are again gradient-based and the recommended gradient estimation method to use was the **hypomode**.
- For **shift magnitudes larger than 0.5**, a **multiscale approach** is mandatory.
- **Phase correlation methods suffer from low SNR scenarios**.
- In general, **gradient correlation-based methods improve over phase correlation methods**, and this difference becomes more important as the noise increases.
- For phase-correlation-based methods, the **best windows** used for apodization proved to be the **Hamming** window, the **Blackman** window and the **Tukey** window. Also, in several cases, **avoiding apodization** yielded the best results.
- The TLS GBSE method proved in general to be more accurate than both the standard LS and the CLS variant. For the TLS approach, a singular value decomposition is required, which may be expensive in some cases. In contrast, the CLS method requires the knowledge of the noise

standard deviation, which may not be available in some situations. **If the TLS** method is used, **validating** using existing approaches [95, 91] **must be done** since it proved to be unstable on many situations, particularly under lower SNRs.

- On shifts with **magnitudes between 0.1 and 1.1**, the **bidirectional bias corrected method** of Pham *et al.* [63] achieves good results without requiring interpolation nor an iterative procedure, thus singling it out as the **best overall low-cost method**. As an exception, if the shift magnitude is *a-priori* known to be lower than 0.1, a single iteration of the total least squares method gets the best results among the fast methods.
- For **large shift magnitudes** (last category in Table 7), the **multiscale GBSE** method always achieved the best results. However, it should be noted that the amount of scales used depends on the maximum possible shift magnitude to estimate. Since in the evaluated setup, the maximum displacement was known *a-priori*, the amount of scales was set according to it. If however this information is not known, phase/gradient correlation-based methods should be used. In particular, for high SNR scenarios, either fitting a sinc on the PCS [32] or using Hoge’s subspace method [43] yielded the best results. Under **lower SNR, gradient correlation** approaches should be employed.

4.3 Robustness to violation of the brightness constancy constraint

While GBSE methods rely on the brightness constancy constraint, the main strength of phase-correlation methods is their higher tolerance against image changes. Therefore, we performed an experiment, using the same setup described in this section, where a square of size $K \times K$ was removed (i.e., set to white) from the center of the first image before estimating the shift. We tested with $K = [3, 5, 7, 9, 11]$, which, given that the shift is estimated using 50×50 images, this implies removing up to 5% of the image. The results of this are displayed in Fig. 17 for the five values of K . All methods were evaluated using two noise levels: low noise ($\sigma = 0.005$) or high noise ($\sigma = 0.055$).

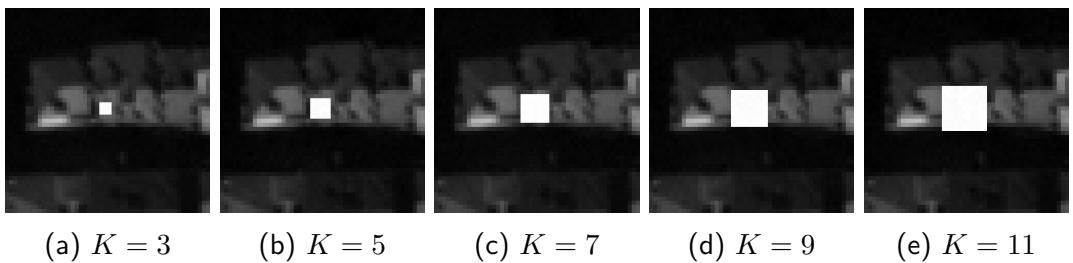


Figure 17: Example evaluated subimages for the five different values of K .

In Table 14 we observe the top 5 methods averaged over shift magnitudes one to three. Although a decrease in performance is observed for GBSE methods, they still improve over phase correlation methods under shift magnitudes lower than one pixel. However, when evaluating shift magnitudes higher than one pixel (category four from Table 7), the situation is different, as seen from Table 15. In fact, GBSE methods still achieve the best accuracies when the size of the removed square is $K = 3$ or, in case of high noise, when $K \leq 7$. On every other case, GBSE methods are not recommended. For example, with $K = 11$, the best results were obtained using phase correlation methods as expected, and most importantly, under high SNR scenarios, the difference between them and the best performing GBSE variant was up to four times better in terms of the measured error in pixels. What is worse, since the deleted square was removed from the center of each subimage, then the windowing procedure applied before computing the phase correlation removes almost all the image information (this explains why the best performing PC methods did not perform apodization). Since the difference between both images does not necessarily have to be in the center, this implies

phase correlation-based methods should obtain even better results accentuating the improvement with respect to GBSE methods.

	$K = 3$	$K = 5$	$K = 7$	$K = 9$	$K = 11$
σ 0.005	0.0231 MS-3,321-IfssGh	0.0376 LS-4-IfGch2	0.0632 LS-4-IfGch2	0.0762 LS-4-IfGch2	0.0940 PC-SINC-Wnw
	0.0234 LS-4-IdGch2	0.0376 MS-2-LS4-IfGch2	0.0632 MS-2-LS4-IfGch2	0.0762 MS-2-LS4-IfGch2	0.0944 LS-4-IfGch2
	0.0241 TLS-4-IdGg0.3	0.0428 TLS-4-IdGch1	0.0739 CLS-4-IdGch1	0.0883 CLS-4-IdGch1	0.0944 MS-2-LS4-IfGch2
	0.0279 CLS-4-IfGg0.6	0.0505 CLS-4-IfGg0.6	0.0766 TLS-4-IdGch1	0.0967 PC-SINC-Wnw	0.1474 CLS-4-IfGfa3
	0.0565 ADF2-2QI	0.0730 ULS-Gch1	0.0991 PC-SINC-Wnw	0.1196 ULS-Gh	0.1506 PCSTONE-Wnw
σ 0.055	0.0645 MS-2,31-IsGfa3	0.0815 CLS-4-IdGg0.6	0.1016 LS-4-IfGfa3	0.1221 LS-4-IfGfa3	0.1486 LS-4-IfGch2
	0.0664 CLS-4-IfGg0.6	0.0831 LS-4-IdGfa3	0.1016 MS-2-LS4-IfGfa3	0.1221 MS-2-LS4-IfGfa3	0.1486 MS-2-LS4-IfGch2
	0.0665 LS-4-IdGfa3	0.0831 MS-2-LS4-IdGfa3	0.1030 CLS-4-IfGg0.6	0.1380 CLS-4-IfGfa3	0.1546 CLS-4-IdGch1
	0.0796 TLS-4-IfGh	0.1117 TLS-4-IdGh	0.1221 ULS-Gh	0.1559 PCSTONE-Wnw	0.1651 PCSTONE-Wnw
	0.0933 ULS-Gh	0.1207 ULS-Gch2	0.1460 TLS-4-IdGh	0.1622 ULS-Gg0.3	0.1814 TLS-4-IdGch1

Table 14: Average error of top 5 evaluated methods per noise and K value for shift magnitudes one to three.

Rows: two noise levels. **Columns:** five K values. Only the best variant per method is displayed.

	$K = 3$	$K = 5$	$K = 7$	$K = 9$	$K = 11$
σ 0.005	0.0324 MS-4,4321-IfssssGh	0.0525 ADF2-2QI	0.1121 PCSINC-Wnw	0.1511 PCSINC-Wnw	0.1888 PCSINC-Wnw
	0.0484 ADF2-2QI	0.0781 MS-5,54321-IfssssGch1	0.1190 PCESINC-Wnw	0.1593 PCESINC-Wnw	0.1935 PCESINC-Wnw
	0.0874 ADF-2QI	0.0868 ADF-2QI	0.1284 PCREN2010-Wnw	0.1669 PCREN2010-Wnw	0.2018 PCREN2010-Wnw
	0.1286 GC04-Gch3	0.1274 PCSINC-Wnw	0.2022 MS-4,4321-IfssssGh	0.3735 PCQUADFIT	0.5107 PCQUADFIT
	0.1935 GC11-Gg0.3	0.1359 PCESINC-Wnw	0.3194 PCQUADFIT	0.4677 MS-3,321-IsGfa3	0.6405 PCGAUSSFIT
	0.2012 PCSINC-Wnw	0.1461 PCREN2010-Wnw	0.4191 SS-HOGE-Wnw	0.5036 PCGAUSSFIT	0.7407 SS-HOGE-Wnw
	0.2098 PCESINC-Wnw	0.3391 SS-HOGE-Wnw	0.4525 PCGAUSSFIT	0.5449 SS-HOGE-Wnw	0.7662 MS-3,321-IsGfa3
σ 0.055	0.1434 MS-4,4321-IsGfa3	0.1512 MS-5,54321-IsGfa3	0.4121 MS-3,321-IsGfa3	0.6444 PCESINC-Wnw	0.6636 PCESINC-Wex
	0.4550 PCREN2010-Wnw	0.4168 PCESINC-Wnw	0.5034 PCREN2010-Wnw	0.6476 PCREN2010-Wnw	0.6654 PCREN2010-Wex
	0.4589 PCESINC-Wnw	0.4176 PCREN2010-Wnw	0.5051 PCESINC-Wnw	0.6496 PCESINC-Wnw	0.6717 PCESINC-Wex
	0.4649 PCSINC-Wnw	0.4228 PCSINC-Wnw	0.5117 PCSINC-Wnw	0.6640 MS-3,321-IsGfa3	0.7000 PCQUADFIT
	0.6876 GC04-Gch3	0.6585 PCQUADFIT	0.7304 PCQUADFIT	0.8098 PCQUADFIT	0.7557 PCGAUSSFIT
	0.7474 PCQUADFIT	0.6693 GC04-G5	0.7952 PCGAUSSFIT	0.8154 GC11-Gch2	0.7859 MS-3,321-IsGfa3
	0.8013 PCGAUSSFIT	0.7211 PCFOO	0.9658 PCFOO	0.8858 PCGAUSSFIT	0.9326 PCFOO

Table 15: Average error of top 7 evaluated methods per noise and K value for the fourth shift magnitude. **Rows:** two noise levels. **Columns:** five K values. Only the best variant per method is displayed.

4.3.1 Conclusions from these experiments

In this section, we studied the behaviour of shift estimation methods when both images differ. This difference could be caused for example, by occlusions, different information appearing on each spectra on multi-spectral images or because both images were taken at different moments in time. In these cases, when the underlying displacement is lower than one pixel (categories one to three from Table 7), GBSE methods yield the best results. However, with larger displacement magnitudes, phase correlation approaches should be considered. In particular, with high SNR, phase correlation methods that use local function fitting in the spatial domain, described in section 3.1, yield the best results and dramatically improve over GBSE methods. Under lower SNR scenarios, the improvement of PC over GBSE methods becomes less significative, and both approaches should be considered.

4.4 Robustness to aliased sample processes

We evaluated the robustness of each approach to aliasing. To simulate aliasing, we first applied a Gaussian low-pass filter of parameter σ_A to a high-resolution image, yielding $\bar{I} = g_{\sigma_A} * I$ where g denotes the Gaussian kernel and $*$ the convolution operator. Then we took both images

$$\tilde{I}_1 = \text{subsample}(\bar{I}) \quad \tilde{I}_2 = \text{subsample}(\text{shift}(\bar{I}, d)) \quad (125)$$

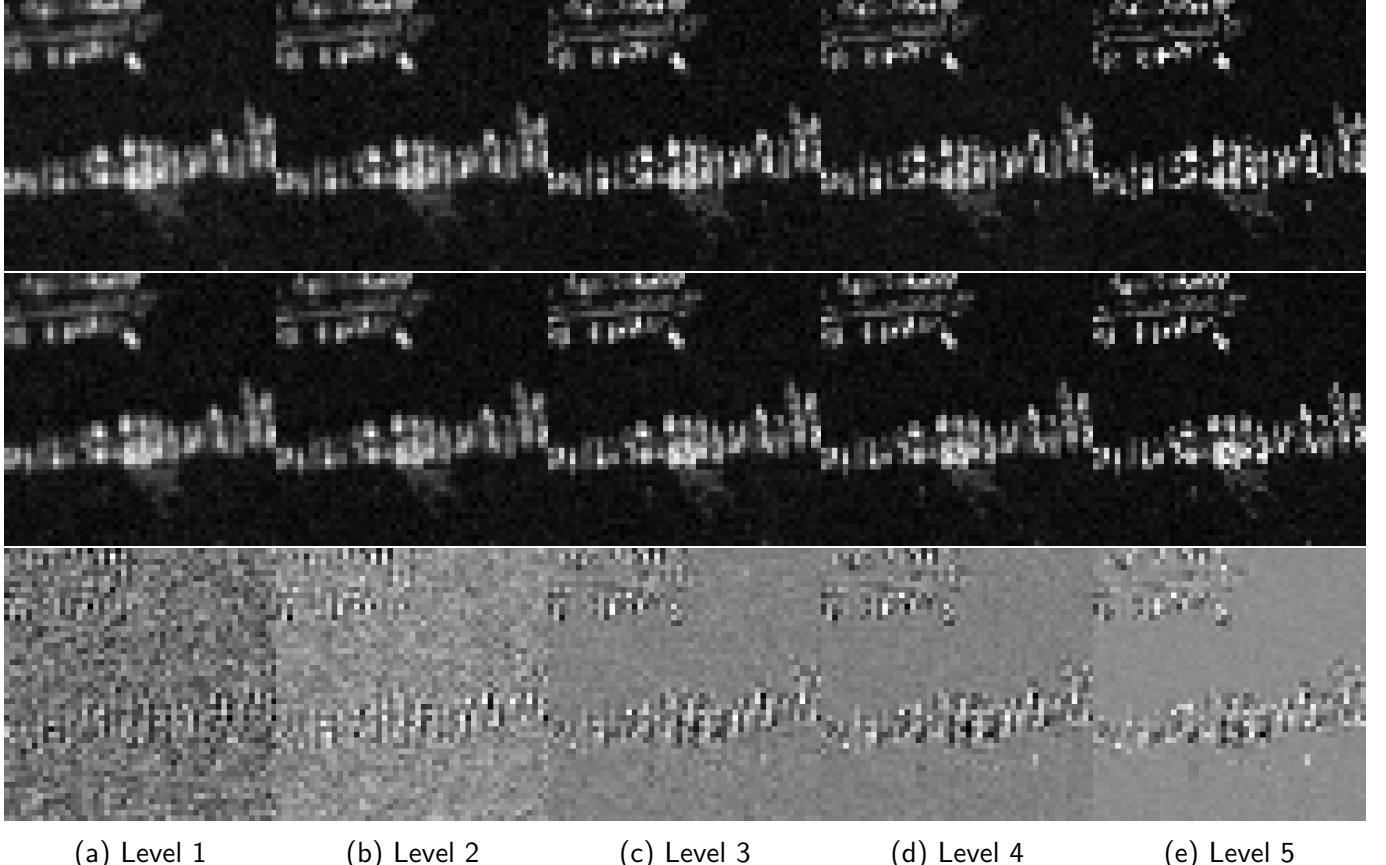


Figure 18: Input images shifted by $(-0.25, 0.5)$ used for testing aliasing influence for each aliasing level. **Top:** \tilde{I}_1 . **Center:** \tilde{I}_2 . **Bottom:** Difference between compensated images (aliasing + noise effects). Dynamic ranges extended.

where $shift$ is an integer resampling method (only moves the pixels), d is 16 times the desired subpixel displacement and $subsample$ subsamples the image by regularly sampling 1 out of 16 pixels. We evaluated $d = [-7, -4, 0, 8] \times [-7, -4, 0, 8]$, where in this case \times corresponds to the cartesian product. In total, this implied estimating 16 subpixel shifts where for each dimension the shifts were $[-0.4375, -0.25, 0, 0.5]$. To test different levels of aliasing, we used five values for σ_A displayed in Table 16.

Level	1	2	3	4	5
σ_A	8	6.4	4	2.8	0.8

Table 16: Values of the standard deviation of the Gaussian used to prefilter the image before subsampling to generate different aliasing levels

All experiments were performed using the same two subimages extracted from the same location of both \tilde{I}_1 and \tilde{I}_2 respectively, by adding a mild noise of $\sigma_N = 75$ which is equivalent to noise level 3. Both images are displayed in Fig. 18 for the five simulated levels of aliasing for the shift $(-0.25, 0.5)$ which proved to be troublesome for several methods. The difference between compensated images $\tilde{I}_1(x, y) - \tilde{I}_2(x + 0.25, y - 0.5)$, shown on the bottom row of the figure, illustrates the effects of aliasing on the shift estimation problem. As the aliasing level increases, more differences appear on the edges of the objects, and these differences get larger in value. These differences may also disorient shift estimation methods generating non-existing secondary motions which alter the final results.

Nevertheless, results in Fig. 19 prove that gradient-based methods are still able to deal with

aliasing even though they break the brightness constancy constraint on which they are based. To generate this figure, one candidate variant from each evaluated shift estimation algorithm was selected, particularly the variant obtaining the highest average error among all aliasing levels and evaluated shifts. The figure then displays the top ten of these variants. As seen, GBSE approaches using spline interpolation and Gaussian derivatives with $\sigma = 0.6$ gave the best results on average using few computational resources. Among the phase correlation methods, the improved gradient correlation approach of [93] seems the least affected by aliasing, particularly when using short supported filters for gradient calculation (either Gaussian with $\sigma = 0.3$ or the first order Christmas kernel). The Stone method [86] also achieved accurate results without performing apodization of the input images, and should be considered, thanks to its fast implementation (in our computer, among these methods, it turned out to be the fastest). Finally, the Pham bidirectional approach [63] offers a good balance between speed and robustness against aliasing.

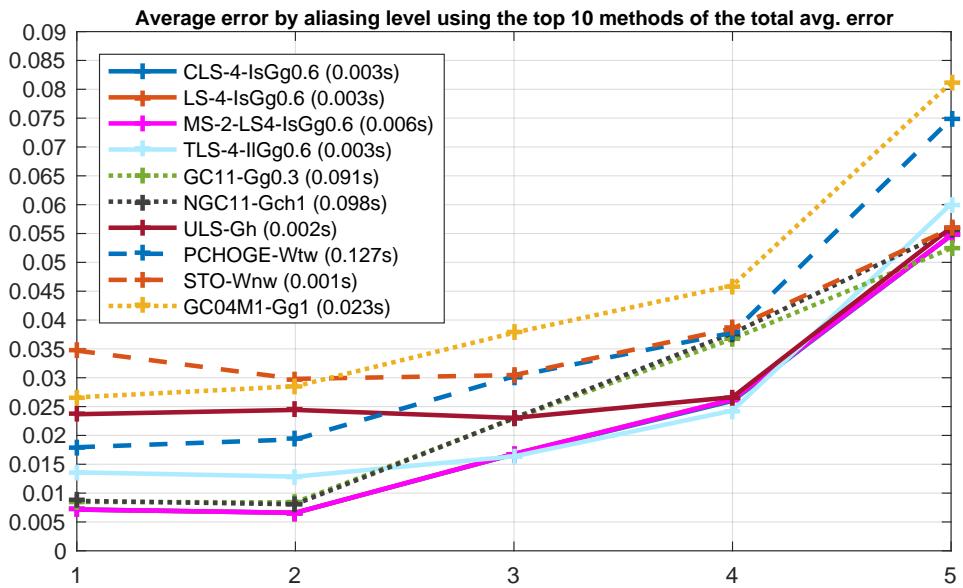


Figure 19: Average error by aliasing level of top ten overall methods

Since aliasing is in general unpredictable and may impact more or less the accuracy depending on the sampling, averaging over all shifts may hide interesting cases. Indeed, for most evaluated displacements, the average accuracy of the top 10 approaches never exceed 0.04 pixels of error, however, we observed two situations for which the error was considerably higher. Results for two different shifts $(-0.437, 0.5)$ and $(-0.25, 0.5)$ displayed in Fig. 20 show a radical change on the methods obtaining the best results. Particularly in those cases, the variants presented by Tzimiropoulos *et al.* [93] based on gradient correlation obtained the best results, which allows us to conclude that their method is the most robust against aliasing. As for differential methods, the ULS approach again turned out to outperform its class. From these figures two important conclusions should be made. First, that indeed the claims of aliasing robustness of the Christmas derivatives (of first and second order) are true and should be considered when high aliasing exists. And secondly, as expected, that iterative approaches should avoid resampling in the frequency domain, due to a potential increase of artifacts appearing from working with images sampled below the Nyquist rate.

On the other side, an example of the top 10 methods usually resulting for every tested displacement is shown in Fig. 21. GBSE approaches in general proved to be quite robust against low to moderate aliasing scenarios, while also achieving the most accurate results using the fewest computational resources.

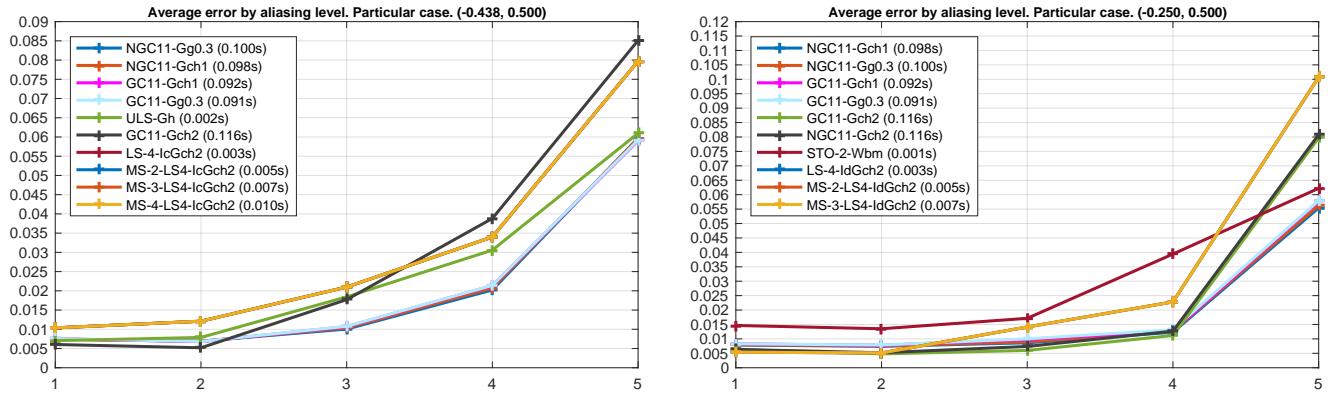


Figure 20: Average error by aliasing level of top 10 for two particular shifts

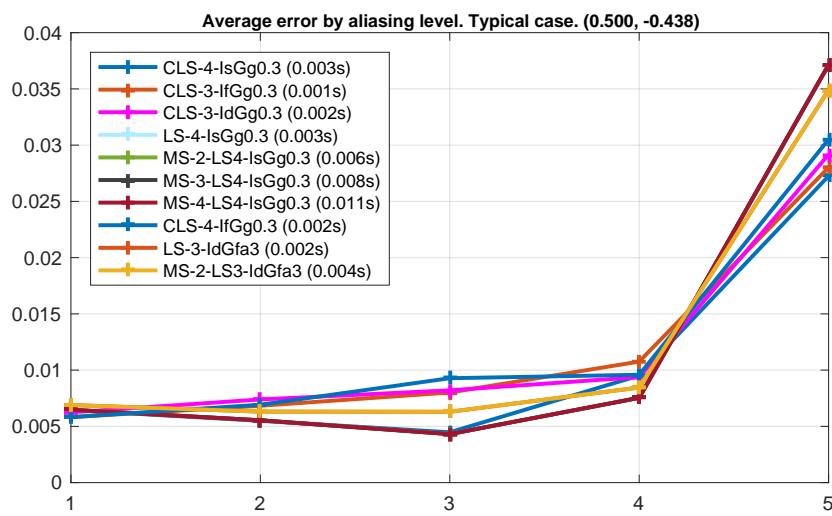


Figure 21: Average error by aliasing level of top 10 methods for a typical shift

4.4.1 Conclusions from these experiments

Three conclusions were drawn from these experiments when dealing with aliasing. First, that GBSE methods (in particular CLS or ULS) offer the best results in terms of accuracy over time consumed. Second, that gradient estimation should be performed using either the kernels proposed by Christmas [21] or Gaussian derivatives with $\sigma = 0.3$ or $\sigma = 0.6$. Finally, under extreme aliasing scenarios, the slightly more expensive gradient correlation method proved to offer the best results.

4.5 Computational cost comparison

To provide an idea of the computational resources required for each method, we averaged execution times for 1000 executions using several shift magnitudes and noises. The processing time of each method was measured using non-optimized Matlab implementations on an Intel Xeon E5-2650 CPU. Results, shown in Fig. 22, are orientative and should not be taken as a definitive measurement. Indeed, some displayed times are evidently not representative. As an example, the subspace method of Robinson (SS-ROBINSON-Wtw) should reflect lower processing time given its low complexity. Both Sinc fitting (PC-SINC-Wtw) together with gradient correlation methods based on non-linear kernel fitting (GC11) employ considerable time due to the optimization performed. The optimization is done by means of the *lsqcurvefit* function in Matlab which besides the optimization procedure, also does an important amount of overhead work used to initialize internal components that would not exist if a more direct implementation were available.

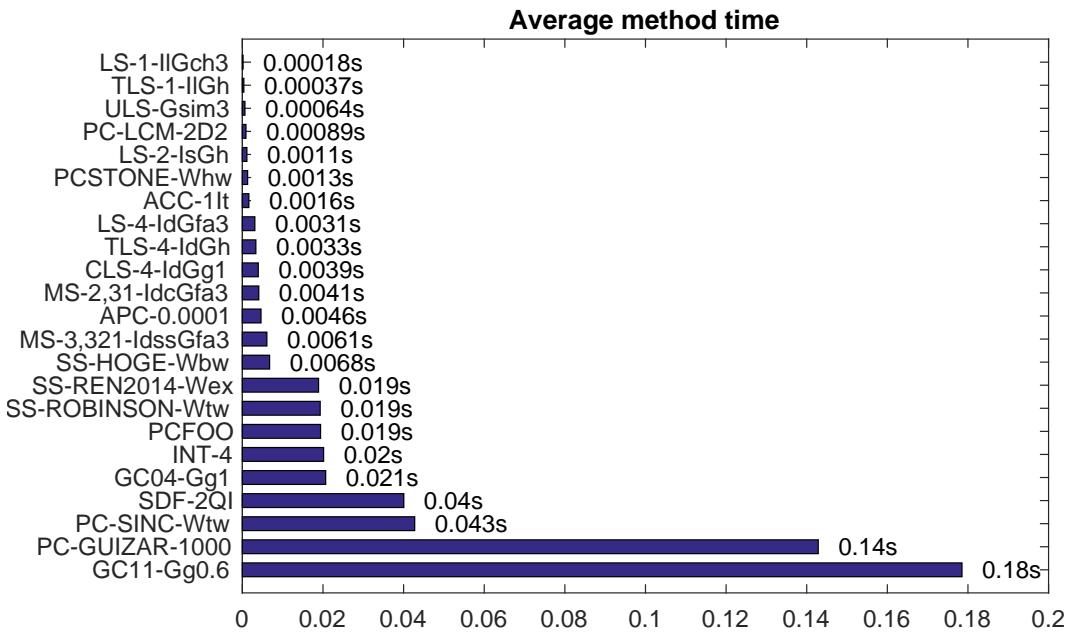


Figure 22: Average execution time for some representative methods

To contrast processing time with precision, we averaged the errors for each method over all noise levels and over the first three categories of shift magnitudes. We also made this comparison for each shift magnitude individually. Results displayed in Fig. 23 and Fig. 24 represent a general summary of this review.

Based on the provided figures, we conclude that gradient-based methods not only achieve more accurate results but also require fewer computational resources. When the shift estimation must cope with severe computational constraints, single iterated GBSE approaches are recommended, and the best option, as seen before, depends on the shift size. If the underlying shift magnitudes

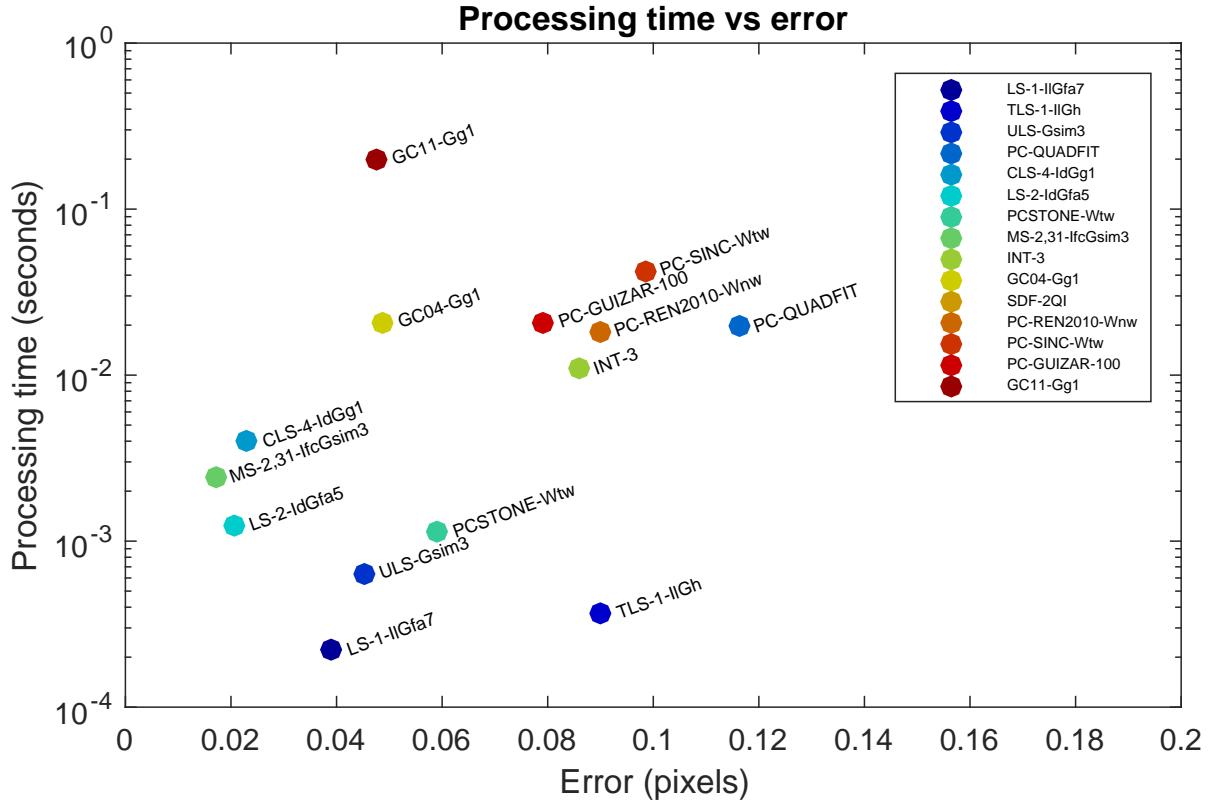


Figure 23: Average execution time (log scale) vs accuracy for some representative methods averaged over the first three shift magnitudes and all noise levels

are small, LS or TLS methods using short gradient estimation kernels support should be selected. If shifts are uniformly distributed along the $[-1,1]$ interval, the bidirectional bias corrected (ULS) approach proves to be the best choice. For larger displacements, the recommended approach is multiscale GBSE, the method of Hoge [43] or the approach of Stone [86]. Under less demanding time constraints, gradient-correlation-based methods should be considered.

4.6 Evaluation on real MRI images

We evaluated every method against a set of real MRI images of a grapefruit. This set, provided by Hoge [43] together with its ground truth, is used in several articles of the state-of-the-art to contrast existing approaches. It consists of five different images shifted. To offer a more complete evaluation, we tested all shift estimation methods under five different noise levels injected to the images. Again, we assumed additive white Gaussian noise and the standard deviation for each noise level is shown in Table 6. In Fig. 25 we show the original image along with two other versions generated by manually adding white Gaussian noise.

For this experiment and due to the length of the underlying shifts to estimate, we included two multiscale GBSE methods using four and five scales, with increasing iterations across scales, spline interpolation and hypomode gradient, denoted MS-4 and MS-5 respectively. Results in Table 17 show that errors were considerably higher than what was obtained in the previous simulated experiments. To explain this, we used the ground truth to resample each image with respect to the other. In Fig. 26 we observe the absolute difference between an image and another aligned with it using the ground truth. As can be noted, for both left and middle images, the difference is considerably high on the image edges, which proves the images are not correctly registered, even though the ground truth was used. For the third case, the images are better registered and what is observed is mainly

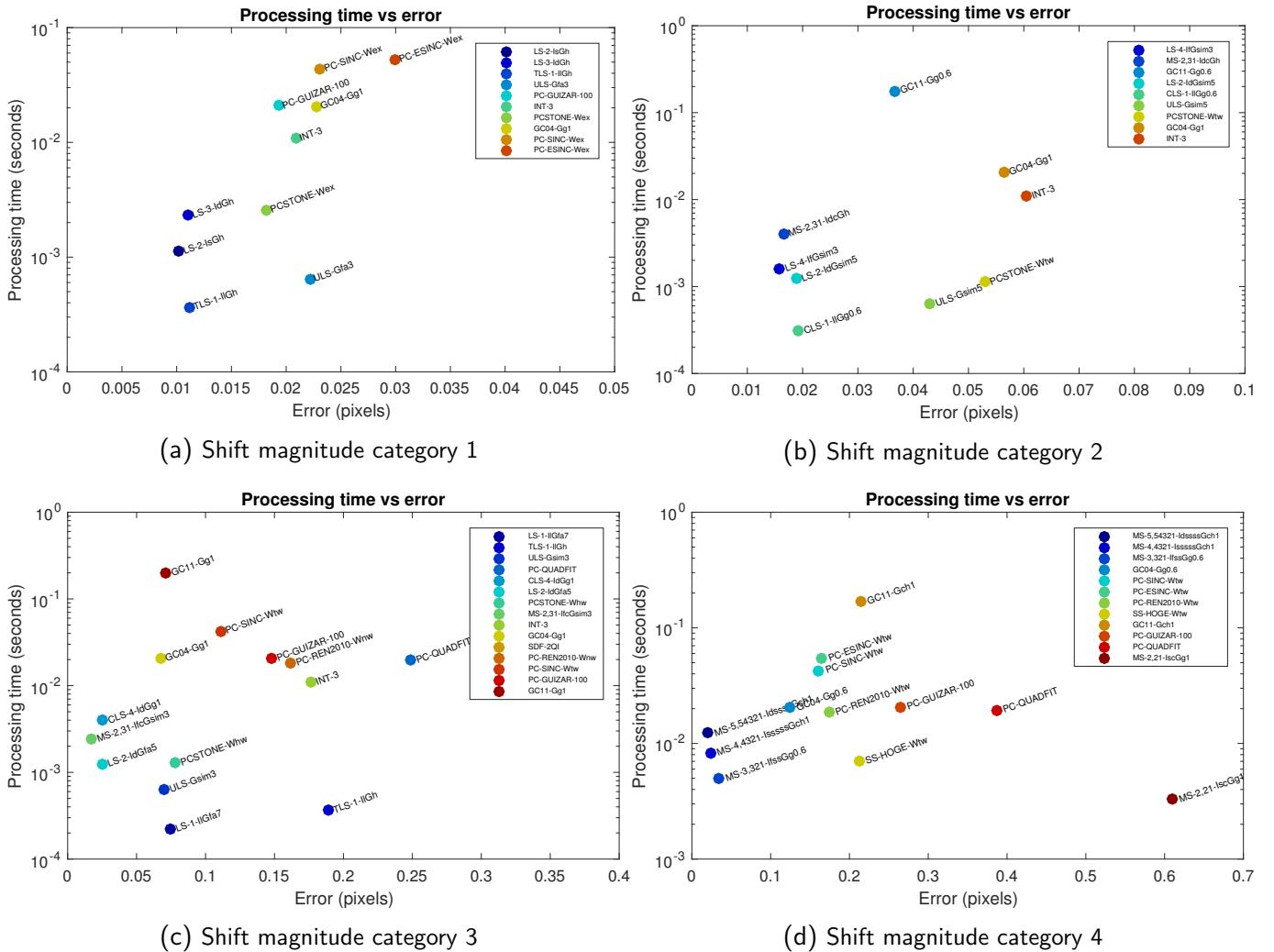


Figure 24: Average execution time (log scale) vs accuracy for some representative methods for each shift magnitude category averaged over all noise levels

(signal dependent) noise and some minor aliasing. Therefore, this explains the significative errors achieved by most methods. This fact is indeed of major importance, since several articles compared their methods using this dataset [43, 72, 73, 93]. Based on this result, the analysis of performance for every method for the whole dataset is not feasible, thus ommited. However, as an example, we evaluated every method by estimating the shift between the fourth and the fifth images of the dataset, for which the provided ground truth appeared to be faithful (by observing the difference between both aligned images). Results appearing in Table 18 again prove that the multiscale GBSE approach is the best overall option, however, by a shorter margin in this case. The next best competitor is again gradient-correlation-based methods [93, 6]. The iterative periodic correlation method [79] also gathered acceptable results, however the obtained error was too high for the lowest evaluated SNR. As for phase-correlation approaches, the esinc [7] almost always appear as the best option for this images. The top 10 methods computed by taking the averages over all noise levels for these two images is shown in Table 19. We observe that only six method types achieved acceptable results for all noise levels.

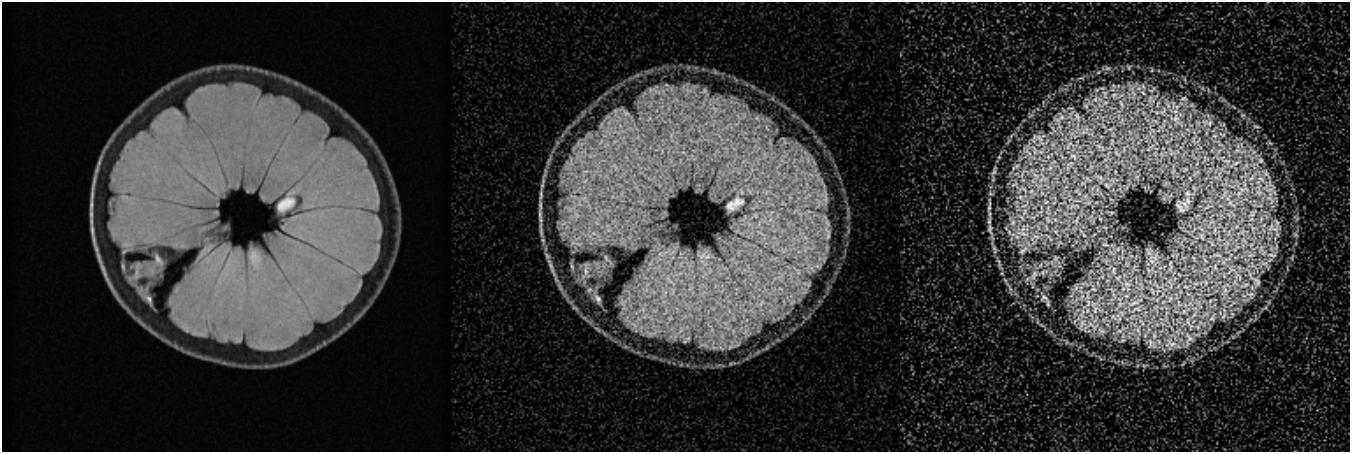


Figure 25: First image of the MRI set provided by Hoge [43] under different noise levels. **Left:** Original image (level 1). **Middle:** $\sigma = 0.015$ (level 3). **Right:** $\sigma = 0.055$ (level 5)

$\sigma = 0.000$	$\sigma = 0.005$	$\sigma = 0.015$	$\sigma = 0.025$	$\sigma = 0.055$
0.1753 SS-HOGE-Wbh	0.1742 PC-LCM-2D2	0.1913 NGC11-Gch3	0.2021 GBSE-MS-4	0.2132 GBSE-MS-5
0.1815 PC-LCM-1D2	0.1856 SS-HOGE-Wbh	0.1927 PC-LCM-2D1	0.2075 APC-0.0001	0.2137 NGC11-Gg0.6
0.1850 APC-0.0001	0.1908 APC-0.0001	0.1941 APC-0.0001	0.2119 NGC11-G6	0.2210 GC11-Gg1
0.1894 NGC04-Gg0.6	0.1935 NGC11-G6	0.1953 GC11-Gch3	0.2121 GC11-G6	0.2241 GC04-Gg1
0.1942 NGC11-G5	0.1938 GC11-G6	0.1985 GC04-G6	0.2142 GC04-Gg1	0.2403 ADF-2QI
0.1942 GC11-G5	0.1948 MS-4	0.2039 GBSE-MS-5	0.2161 PC-ESINC-Wbh	0.2403 ADF2-2QI
0.1962 MS-4,4321-IssssssGch3	0.2005 GC04-Gg1	0.2065 SDF-2QI	0.2270 SDF-2QI	0.2461 CFI-2QI
0.2025 GC04-G6	0.2046 CFI-2QI	0.2068 CFI-2QI	0.2279 CFI-2QI	0.2472 SDF-2QI
0.2043 SDF-2QI	0.2046 SDF-2QI	0.2155 ADF2-2QI	0.2304 ADF2-2QI	0.2493 PC-GUIZAR-10
0.2045 CFI-2QI	0.2115 ADF2-2QI	0.2160 ADF2-2QI	0.2307 ADF2-2QI	0.2549 APC-0.0001

Table 17: Error values for top 10 best variants of each method averaged over all measurements for the grapefruit MRI dataset by injecting different amounts of additive white Gaussian noise. Note that these results cannot be relied upon, as the ground truth provided for this dataset appears to be incorrect.

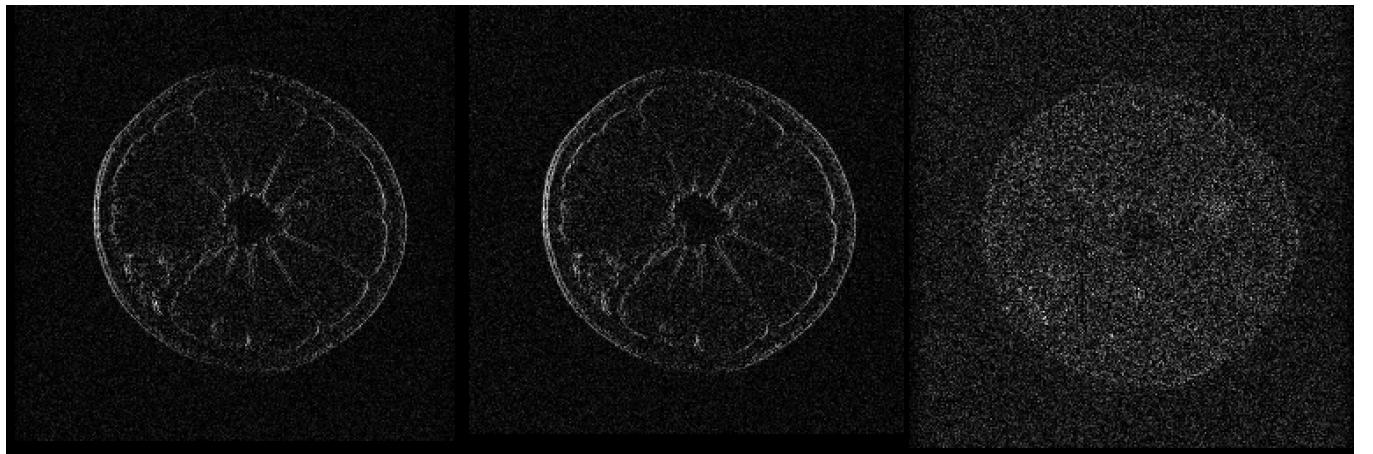


Figure 26: Absolute value of the difference between aligned images based on the ground truth information provided by Hoge [43] using the original images. **Left:** Between the first and the third image. **Middle:** Between the first and the fifth image. **Right:** Between the third and the fifth image. Dynamic ranges extended for improved visual perception.

$\sigma = 0.000$	$\sigma = 0.005$	$\sigma = 0.015$	$\sigma = 0.025$	$\sigma = 0.055$
0.0024 MS-3,321-IfssGg1	0.0136 MS-3,321-IsssGsim3	0.0229 MS-3,321-IsssGsim5	0.0296 MS-3,321-IsssGfa3	0.0587 MS-3,321-IdssGfa5
0.0056 PC-ESINC-Wex	0.0162 GC11-Gch1	0.0264 GC11-Gg0.3	0.0357 GC11-Gg0.6	0.0597 GC11-Gg1
0.0062 NGC04-Gg1	0.0167 PC-GUIZAR-10	0.0275 NGC11-Gch1	0.0377 NGC11-Gg0.6	0.0634 NGC11-Gg1
0.0095 SS-HOGE-Wtw	0.0175 NGC11-Gch2	0.0307 GC04-Gg1	0.0420 GC04-Gg1	0.0672 GC04-Gg1
0.0115 NGC11-Gg1	0.0199 APC-3It	0.0394 APC-2It	0.0625 APC-0.01	0.1260 PC-ESINC-Wtw
0.0118 APC-0.0001	0.0245 GC04-Gg1	0.0423 PC-ESINC-Wtw	0.0690 PC-ESINC-Wtw	0.1458 APC-2It
0.0127 GC11-Gch3	0.0269 SS-HOGE-Wtw	0.1838 SS-HOGE-Whw	0.3425 PCFOO	0.4030 PCFOO
0.0243 GC04-Gg1	0.2594 PCFOO	0.2930 PCFOO	0.3645 SS-REN2014-Wtw	0.9873 ACC-2It
0.0842 ACC-0.0001	0.3413 ACC-3It	0.4506 ACC-3It	0.7081 ACC-2It	1.5331 SS-REN2014-Whw
0.1370 PCFOO	3.5684 CLS-4-IdGfa7	4.2313 CLS-4-IfGfa7	4.2080 LS-4-IdGfa7	4.7611 LS-4-IsGfa7

Table 18: Average error values for top 10 best variants of each method by estimating the shift between the fourth and the fifth images of the grapefruit MRI dataset (ground truth shift: (7.68, 0)) by injecting different amounts of additive white Gaussian noise.

Avg
0.0282 MS-4,4321-IssssssGsim3
0.0317 GC11-Gg0.6
0.0332 NGC11-Gg0.6
0.0377 GC04-Gg1
0.0558 PC-ESINC-Wtw
0.0563 APC-3It
0.2870 PCFOO
0.5266 ACC-3It
0.5683 SS-REN2014-Whw
4.0268 CLS-4-IdGfa7

Table 19: Average error values for all levels of noise for top 10 best variants of each method by estimating the shift between the fourth and the fifth images of the grapefruit MRI dataset (ground truth shift: (7.68, 0)).

5 Concluding Remarks

In this article we performed an in-depth review of fast and accurate shift estimation methods. Although most recent work on shift estimation is based on phase-correlation methods, GBSE approaches proved to be more accurate, stable and run faster than phase correlation approaches, although they are limited to subpixel estimation or required to be used in a pyramidal multiscale approach in order to estimate larger shifts. On the other side, gradient-correlation-based methods showed an improvement over phase correlation methods, particularly under higher noise or highly aliased scenarios, although they usually require more computational resources. Throughout this article, we found out that if *a-priori* knowledge is available regarding the maximum shift magnitude to estimate, then better estimation accuracies can be achieved by selecting a more suitable variant for each method. We provided the reader with such fine-tuning tips. Another interesting discovery is that one of the most used ground truth data used for comparison among state-of-the-art methods is not precise and should be avoided. Finally and most importantly, based on this study and on the characteristics of the underlying shift estimation problem, a practical recipe is offered to the community in order to achieve fast and accurate shift estimation.

Despite the extensive length of this review, we plan to include other useful information. First, some interesting methods were excluded from the final comparison due to the complexity of their implementations or high computational times. These were the methods of Takita [87], Balci [12], Leprince [51] and the recent method of Tong *et al.* [90]. We plan to add them in the future. Another considered task, is to test the robustness of each method under varying illumination conditions between the images. We believe that in these cases, if no histogram equalization is possible, GBSE methods should be avoided and phase or gradient correlation approaches would become first options. Also, we plan on combining different presented approaches to obtain a more general method yielding good results on every situation. Finally, we plan to provide publicly a ground truth dataset to be used for comparison of shift estimation methods, considering all difficulties usually faced on such

task.

Acknowledgements

I would like to thank Prof. Vasileios Argyriou for providing several source codes helping the development of this review. We also thank Dr. Tuan Q. Pham for fruitful comments and discussions.

5.1 Copyright

Authors must warrant to be the authors of every part of the final article including the images, or must include permission of the copyright holder. If you use images you do not own, you must include their authors and permissions to be used in the article in a special section at the end of the article. We want other researchers to be able to reuse these images for comparisons in other articles; try to use only images under a permissive license like “CC-BY” or “CC0”.

6 References

References

- [1] IKRAM E. ABDOU, *Practical approach to the registration of multiple frames of video images*, 1998.
- [2] ANDREW ADAMS, NATASHA GELFAND, AND KARI PULLI, *Viewfinder alignment*, Computer Graphics Forum, 27 (2008), pp. 597–606.
- [3] ALFONSO ALBA, J. FLAVIO VIGUERAS-GOMEZ, EDGAR R. ARCE-SANTANA, AND RUTH M. AGUILAR-PONCE, *Phase correlation with sub-pixel accuracy: A comparative study in 1d and 2d*, Computer Vision and Image Understanding, 137 (2015), pp. 76 – 87.
- [4] S. ALLINEY AND C. MORANDI, *Digital image registration using projections*, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8 (1986), pp. 222–233.
- [5] F. J. ANSCOMBE, *The transformation of poisson, binomial and negative-binomial data*, Biometrika, 35 (1948), pp. 246–254.
- [6] V. ARGYRIOU AND T. VLACHOS, *Using gradient correlation for sub-pixel motion estimation of video sequences*, in Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on, vol. 3, May 2004, pp. iii–329–32 vol.3.
- [7] V. ARGYRIOU AND T. VLACHOS, *A study of sub-pixel motion estimation using phase correlation*, in Proc. BMVC, 2006, pp. 40.1–40.10. doi:10.5244/C.20.40.
- [8] V. ARGYRIOU AND T. VLACHOS, *Quad-tree motion estimation in the frequency domain using gradient correlation*, IEEE Transactions on Multimedia, 9 (2007), pp. 1147–1154.
- [9] ALIREZA BAB-HADIASHAR AND DAVID SUTER, *Robust optic flow computation*, International Journal of Computer Vision, 29 (1998), pp. 59–77.
- [10] ALIREZA BAB-HADIASHAR AND DAVID SUTER, *Robust total least squares based optic flow computation*, in Proc. 3rd ACCV, Hong Kong, 1998, pp. 566–573.

- [11] SIMON BAKER AND IAIN MATTHEWS, *Lucas-kanade 20 years on: A unifying framework*, IJCV, 56 (2004), pp. 221–255.
- [12] M. BALCI AND H. FOROOSH, *Estimating sub-pixel shifts directly from the phase difference*, in IEEE International Conference on Image Processing 2005, vol. 1, Sept 2005, pp. I–1057–60.
- [13] SHAUL K. BAR-LEV AND PETER ENIS, *On the classical choice of variance stabilizing transformations and an application for a poisson variate*, Biometrika, 75 (1988), pp. 803–804.
- [14] M. S. BARTLETT, *The square root transformation in analysis of variance*, Supplement to the Journal of the Royal Statistical Society, 3 (1936), pp. 68–78.
- [15] STEVEN S. BEAUCHEMIN AND JOHN L. BARRON, *The computation of optical flow*, ACM computing surveys (CSUR), 27 (1995), pp. 433–466.
- [16] G. M. BERNSTEIN AND D. GRUEN, *Resampling Images in Fourier Domain*, Publications of the Astronomical Society of the Pacific, 126 (2014), pp. 287–295.
- [17] LISA GOTTESFELD BROWN, *A survey of image registration techniques*, ACM computing surveys (CSUR), 24 (1992), pp. 325–376.
- [18] R. BRUNELLI, *Template Matching Techniques in Computer Vision: Theory and Practice*, Wiley, 2009.
- [19] PETER J BURT AND EDWARD H ADELSON, *The laplacian pyramid as a compact image code*, Communications, IEEE Transactions on, 31 (1983), pp. 532–540.
- [20] HAIYANG CHAO, YU GU, AND MARCELLO NAPOLITANO, *A survey of optical flow techniques for robotics navigation applications*, Journal of Intelligent & Robotic Systems, 73 (2014), pp. 361–372.
- [21] WILLIAM J CHRISTMAS, *Spatial filtering requirements for gradient-based optical flow measurement.*, in BMVC, 1998, pp. 1–10.
- [22] J. H. CURTISS, *On transformations used in the analysis of variance*, The Annals of Mathematical Statistics, 14 (1943), pp. 107–122.
- [23] CARL DE BOOR, *A practical guide to splines; rev. ed.*, Applied mathematical sciences, Springer, Berlin, 2001.
- [24] JULIE DELON, *Midway image equalization*, Journal of Mathematical Imaging and Vision, 21 (2004), pp. 119–134.
- [25] CARL ECKART AND GALE YOUNG, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.
- [26] CHURCHILL EISENHART, *The assumptions underlying the analysis of variance*, Biometrics, 3 (1947), pp. 1–21.
- [27] M. ELAD, P. TEO, AND Y. HEL-OR, *Optimal filters for gradient-based motion estimation*, in Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, vol. 1, 1999, pp. 559–565 vol.1.
- [28] S ERTURK, *Digital image stabilization with sub-image phase correlation based global motion estimation*, IEEE Transactions on Consumer Electronics, 49 (2003), pp. 1320–1325.

- [29] HANY FARID AND EERO P SIMONCELLI, *Differentiation of discrete multidimensional signals*, IEEE Transactions on Image Processing, 13 (2004), pp. 496–508.
- [30] HOMA FASHANDI, REZA FAZEL-REZAI, AND STEPHEN PISTORIUS, *Optical flow and total least squares solution for multi-scale data in an over-determined system*, in Advances in Visual Computing, George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Nikos Paragios, Syeda-Mahmood Tanveer, Tao Ju, Zicheng Liu, Sabine Coquillart, Carolina Cruz-Neira, Torsten Mller, and Tom Malzbender, eds., vol. 4842 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 33–42.
- [31] DAVID J. FLEET AND YAIR WEISS, *Optical flow estimation*, 2005.
- [32] H. FOROOOSH, J.B. ZERUBIA, AND M. BERTHOD, *Extension of phase correlation to subpixel registration*, Image Processing, IEEE Transactions on, 11 (2002), pp. 188 –200.
- [33] MURRAY F. FREEMAN AND JOHN W. TUKEY, *Transformations related to the angular and the square root*, Ann. Math. Statist., 21 (1950), pp. 607–611.
- [34] DENNIS C GHIGLIA AND MARK D PRITT, *Two-dimensional phase unwrapping: theory, algorithms, and software*, vol. 4, Wiley New York, 1998.
- [35] ANDREW GILMAN AND ARNO LEIST, *Global Illumination-Invariant Fast Sub-Pixel Image Registration*, July 2013, pp. 95–100.
- [36] G.H. GOLUB, ALAN HOFFMAN, AND G.W. STEWART, *A generalization of the eckart-young-mirsky matrix approximation theorem*, Linear Algebra and its Applications, 88 (1987), pp. 317 – 327.
- [37] R.C. GONZALEZ AND R.E. WOODS, *Digital Image Processing*, Pearson/Prentice Hall, 2008.
- [38] A. GOSHTASBY, G. C. STOCKMAN, AND C. V. PAGE, *A region-based approach to digital image registration with subpixel accuracy*, IEEE Transactions on Geoscience and Remote Sensing, GE-24 (1986), pp. 390–399.
- [39] MANUEL GUIZAR-SICAIROS, SAMUEL T. THURMAN, AND JAMES R. FIENUP, *Efficient subpixel image registration algorithms*, Opt. Lett., 33 (2008), pp. 156–158.
- [40] FUMIO HAYASHI, *Econometrics*, Princeton University Press, Princeton, 2000.
- [41] HUY THO HO AND ROLAND GOECKE, *Optical flow estimation using fourier mellin transform*, in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.
- [42] WILLIAM SCOTT HOGE, *A subspace identification extension to the phase correlation method [mri application]*, Medical Imaging, IEEE Transactions on, 22 (2003), pp. 277–280.
- [43] W. S. HOGE, *A subspace identification extension to the phase correlation method [mri application]*, IEEE Transactions on Medical Imaging, 22 (2003), pp. 277–280.
- [44] PAUL VC HOUGH, *Method and means for recognizing complex patterns*, tech. report, 1962.
- [45] SABINE VAN HUFFEL, JOOS VANDEWALLE, AND ANN HAEGEMANS, *An efficient and reliable algorithm for computing the singular subspace of a matrix, associated with its smallest singular values*, Journal of Computational and Applied Mathematics, 19 (1987), pp. 313 – 330.

- [46] HUI JI AND CORNELIA FERMLLER, *Noise causes slant underestimation in stereo and motion*, Vision Research, 46 (2006), pp. 3105 – 3120.
- [47] R. KEYS, *Cubic convolution interpolation for digital image processing*, IEEE Transactions on Acoustics, Speech, and Signal Processing, 29 (1981), pp. 1153–1160.
- [48] S.J. KIRKPATRICK AND D.D. DUNCAN, *Performance of a gradient-based shift estimator in a spatially sparse data environment: tracking the sub-pixel motion of fluorescent particles*, Journal of Microscopy, 232 (2008), pp. 64–72.
- [49] PER KNUTSSON, METTE OWNER-PETERSEN, AND CHRIS DAINTY, *Extended object wavefront sensing based on the correlation spectrum phase.*, Optics express, 13 (2005), pp. 9527–9536.
- [50] C. D. KUGLIN AND D. C. HINES, *The phase correlation image alignment method*, IEEE Conference on Cybernetics and Society, (1975), pp. 163–165.
- [51] S. LEPRINCE, S. BARBOT, F. AYOUB, AND J. P. AVOUAC, *Automatic and precise orthorectification, coregistration, and subpixel correlation of satellite images, application to ground deformation measurements*, IEEE Transactions on Geoscience and Remote Sensing, 45 (2007), pp. 1529–1558.
- [52] MARLOES MARIA JOHANNA LETTEBOER, PETER WA WILLEMS, MAX A VIERGEVER, AND WIRO J NIJSEN, *Brain shift estimation in image-guided neurosurgery using 3-d ultrasound*, IEEE Transactions on Biomedical Engineering, 52 (2005), pp. 268–276.
- [53] KENNETH LEVENBERG, *A method for the solution of certain non-linear problems in least squares*, Quarterly of Applied Mathematics, 2 (1944), pp. 164–168.
- [54] MATS G LÖFDAHL, *Evaluation of image-shift measurement algorithms for solar Shack-Hartmann wavefront sensors*, Astronomy & Astrophysics, 524 (2010), p. A90.
- [55] DAVID G. LOWE, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision, 60, pp. 91–110.
- [56] BRUCE D. LUCAS AND TAKEO KANADE, *An iterative image registration technique with an application to stereo vision*, in Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI’81, San Francisco, CA, USA, 1981, Morgan Kaufmann Publishers Inc., pp. 674–679.
- [57] RAMAN MAINI AND HIMANSHU AGGARWAL, *A comprehensive review of image enhancement techniques*, CoRR, abs/1003.4053 (2010).
- [58] JB ANTOINE MAINTZ AND MAX A VIERGEVER, *A survey of medical image registration*, Medical image analysis, 2 (1998), pp. 1–36.
- [59] IVAN MARKOVSKY AND SABINE VAN HUFFEL, *Overview of total least-squares methods*, Signal processing, 87 (2007), pp. 2283–2302.
- [60] JEAN MICHEL MOREL AND GUOSHEN YU, *Is sift scale invariant?*, Inverse Problems and Imaging, 5 (2011), pp. 115–136.
- [61] MOHAMMAD ABDUL MUQUIT, TAKUMA SHIBAHARA, AND TAKAFUMI AOKI, *A high-accuracy passive 3d measurement system using phase-based image matching*, IEICE Trans. Fundam. Electron. Commun. Comput. Sci., E89-A (2006), pp. 686–697.

- [62] L. NG AND V. SOLO, *Errors-in-variables modeling in optical flow estimation*, Image Processing, IEEE Transactions on, 10 (2001), pp. 1528–1540.
- [63] T.Q. PHAM AND M. DUGGAN, *Bidirectional bias correction for gradient-based shift estimation*, in Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on, Oct 2008, pp. 829–832.
- [64] T. Q. PHAM, M. BEZUIJEN, L. J. VAN VLIET, K. SCHUTTE, AND C. L. L. HENDRIKS, *Performance of optimal registration estimators*, in Proc. SPIE, vol. 5817, 2005, pp. 133–144.
- [65] T. Q. PHAM AND P. COX, *Multi-hypothesis projection-based shift estimation for sweeping panorama reconstruction*, in 2012 IEEE International Conference on Multimedia and Expo, July 2012, pp. 97–102.
- [66] LISA A POYNEER, *Scene-based Shack-Hartmann wave-front sensing: analysis and simulation*, Applied Optics, 42 (2003), pp. 5807–5815.
- [67] LISA A. POYNEER, DAVID W. PALMER, KAI N. LAFORTUNE, AND BRIAN BAUMAN, *Experimental results for correlation-based wavefront sensing*, in Optics & Photonics 2005, Mark T. Gruneisen, John D. Ginglewski, and Michael K. Giles, eds., International Society for Optics and Photonics, Aug. 2005, pp. 58940N–58940N–142.
- [68] MARTIN RAIS, J.-M. MOREL, AND GABRIELE FACCIOLO, *Iterative gradient-based shift estimation: To multiscale or not to multiscale?*, in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - 20th Iberoamerican Congress, CIARP 2015, Montevideo, Uruguay, November 9-12, 2015, Proceedings, 2015, pp. 416–423.
- [69] M. RAIS, C. THIEBAUT, J. M. DELVIT, AND J. M. MOREL, *A tight multiframe registration problem with application to earth observation satellite design*, in 2014 IEEE International Conference on Imaging Systems and Techniques (IST) Proceedings, Oct 2014, pp. 6–10.
- [70] B.S. REDDY AND B.N. CHATTERJI, *An fft-based technique for translation, rotation, and scale-invariant image registration*, Image Processing, IEEE Transactions on, 5 (1996), pp. 1266 –1271.
- [71] RA REED, *Comparison of subpixel phase correlation methods for image registration*, tech. report, DTIC Document, 2010.
- [72] J. REN, J. JIANG, AND T. VLACHOS, *High-accuracy sub-pixel motion estimation from noisy images in Fourier domain*, IEEE Transactions on Image Processing, 19 (2010), pp. 1379–1384.
- [73] JINCHANG REN, THEODORE VLACHOS, YI ZHANG, JIANGBIN ZHENG, AND JIANMIN JIANG, *Gradient-based subspace phase correlation for fast and effective image alignment*, Journal of Visual Communication and Image Representation, 25 (2014), pp. 1558 – 1565.
- [74] D. ROBINSON AND P. MILANFAR, *Efficiency and accuracy tradeoffs in using projections for motion estimation*, in Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on, vol. 1, Nov 2001, pp. 545–550 vol.1.
- [75] D. ROBINSON AND P. MILANFAR, *Fundamental performance limits in image registration*, Image Processing, IEEE Transactions on, 13 (2004), pp. 1185–1199.

- [76] DIRK ROBINSON AND PEYMAN MILANFAR, *Bias minimizing filter design for gradient-based image registration*, Signal Processing: Image Communication, 20 (2005), pp. 554 – 568. Special Issue on Advanced Aspects of Motion Estimation Special Issue on Advanced Aspects of Motion Estimation.
- [77] N. SABATER, S. LEPRINCE, AND J. P. AVOUAC, *Contrast invariant and affine sub-pixel optical flow*, in 2012 19th IEEE International Conference on Image Processing, Sept 2012, pp. 53–56.
- [78] H. SHEKARFOROUSH, M. BERTHOD, AND J. ZERUBIA, *Subpixel image registration by estimating the polyphase decomposition of cross power spectrum*, in Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on, Jun 1996, pp. 532–537.
- [79] ERKIN SIDICK, *Adaptive periodic-correlation algorithm for extended scene Shack-Hartmann wavefront sensing*, in Imaging and Applied Optics, Optical Society of America, 2011, p. CPDP1.
- [80] E. SIDICK ET AL., *An Adaptive cross-correlation algorithm for extended scene Shack-Hartmann wavefront sensing*, in Adaptive Optics: Methods, Analysis and Applications, OSA, 2007.
- [81] E. SIDICK, R. MORGAN, J. GREEN, C. OHARA, AND D. REDDING, *Adaptive Cross-Correlation Algorithm and Experiment of Extended Scene Shack-Hartmann Wavefront Sensing*, in UV/Optical/IR Space Telescopes: Innovative Technologies and Concepts III, Howard A. MacEwen and James B. Breckinridge, eds., vol. 6687, Sept. 2007, pp. 668710–668711.
- [82] E.P. SIMONCELLI, *Design of multi-dimensional derivative filters*, in Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference, vol. 1, Nov 1994, pp. 790–794 vol.1.
- [83] JULIUS O. SMITH, *Spectral Audio Signal Processing*, <http://ccrma.stanford.edu/~jos/sasp/>. online book, 2011 edition.
- [84] R. SOUMMER, L. PUEYO, A. SIVARAMAKRISHNAN, AND R. J. VANDERBEI, *Fast computation of lyot-style coronagraph propagation*, Opt. Express, 15 (2007), pp. 15935–15951.
- [85] J.-L. STARCK, F. MURTAGH, AND A. BIJAOUI, *Image Processing and Data Analysis: The Multiscale Approach*, Cambridge University Press, New York, NY, USA, 1998.
- [86] H.S. STONE, M.T. ORCHARD, E.-C. CHANG, AND S.A. MARTUCCI, *A fast direct fourier-based algorithm for subpixel registration of images*, Geoscience and Remote Sensing, IEEE Transactions on, 39 (2001), pp. 2235–2243.
- [87] KENJI TAKITA, TAKAFUMI AOKI, YOSHIFUMI SASAKI, TATSUO HIGUCHI, AND KOJI KOBAYASHI, *High-Accuracy Subpixel Image Registration Based on Phase-Only Correlation*, IEEE transactions on fundamentals of electronics, communications and computer sciences, 86 (2003), pp. 1925–1934.
- [88] P. THEVENAZ, U. E. RUTTIMANN, AND M. UNSER, *A pyramid approach to subpixel registration based on intensity*, IEEE Transactions on Image Processing, 7 (1998), pp. 27–41.
- [89] QI TIAN AND MICHAEL N HUHNS, *Algorithms for subpixel registration*, Comput. Vision Graph. Image Process., 35 (1986), pp. 220–233.

²<http://ccrma.stanford.edu/~jos/sasp/>

- [90] XIAOHUA TONG, ZHEN YE, YUSHENG XU, SHIJIE LIU, LINGYUN LI, HUAN XIE, AND TIANPENG LI, *A novel subpixel phase correlation method using singular value decomposition and unified random sample consensus*, IEEE Transactions on Geoscience and Remote Sensing, 53 (2015), pp. 4143–4156.
- [91] CHUN-JEN TSAI, N.P. GALATSANOS, AND AK. KATSAGGELOS, *Total least squares estimation of stereo optical flow*, in Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on, vol. 2, Oct 1998, pp. 622–626 vol.2.
- [92] G. TZIMIROPOULOS AND V. ARGYRIOU, *Gradient schemes for robust fft-based motion estimation*, in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), March 2016, pp. 1781–1785.
- [93] G. TZIMIROPOULOS, V. ARGYRIOU, AND T. STATHAKI, *Subpixel registration with gradient correlation*, IEEE Transactions on Image Processing, 20 (2011), pp. 1761–1767.
- [94] S. VAN HUFFEL AND J. VANDEWALLE, *The Total Least Squares Problem: Computational Aspects and Analysis*, SIAM, Philadelphia, 1991.
- [95] JOSEPH WEBER AND JITENDRA MALIK, *Robust computation of optic flow in a multiscale differential framework*, International Journal of Computer Vision, (1995), pp. 67–81.
- [96] JOACHIM WEICKERT, ANDRÉS BRUHN, THOMAS BROX, AND NILS PAPENBERG, *A Survey on Variational Optic Flow Methods for Small Displacements*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 103–136.
- [97] BARBARA ZTOV AND JAN FLUSSER, *Image registration methods: a survey*, Image and Vision Computing, 21 (2003), pp. 977–1000.

Acknowledgment

Include an acknowledgment of grant and funding organizations at the end of the article. For example: This work was partially funded by the ONR grant 1234ABCD, ERC grant 5678XYZ, and Treilles Foundation.

Image Credits

You must mention the origin of *every* image included in the article, with their authors and license. Use a small font size for this section, and set the height of the thumbnails to 2em. For example: