SANDIA REPORT

SAND200X-XXXX
Unlimited Release
February, 2015

Evaluation of Various Interpolants Available in DICe

Daniel Z. Turner (1444), Phillip L. Reu (1512), and Paul S. Crozier (1444)

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: http://www.ntis.gov/search

SAND200X-XXXX
Unlimited Release
Printed February 2015

Evaluation of Various Interpolants Available in DICe

DZ Turner (1444), PL Reu (1512), and PS Crozier (1444)
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1323

Abstract

This report evaluates several interpolants implemented in the Digital Image Correlation Engine (DICe), an image correlation software package developed by Sandia. By *interpolants* we refer to the basis functions used to represent discrete pixel intensity data as a continuous signal. Interpolation is used to determine intensity values in an image at non-pixel locations. It is also used, in some cases, to evaluate the x and y gradients of the image intensities. Intensity gradients subsequently guide the optimization process. The goal of this report is to inform analysts as to the characteristics of each interpolant and provide guidance towards the best interpolant for a given dataset. This work also serves as an initial verification of each of the interpolants implemented.

Contents

1. Summary of Findings

A summary of the results for each of the interpolants studied in this work is given in Table 1. For each interpolant, a reference is given for a representative paper that describes the theory and implementation (with the exception of bilinear and bicubic which are straightforward). The fastest interpolant to evaluate is the bilinear, but it also has the highest bias error and is therefore discouraged. The B-spline interpolants have high orders of accuracy, but they are considerably more expensive to evaluate. All things considered, the fourth-order Keys interpolant performs the best. The fourth-order Keys interpolant has a small bias error, negligible rotation error, and is only slightly more expensive to compute than the bilinear interpolant. The reader should note that the filtering properties of the Keys interpolants are not as good as the B-splines (Spline3 and Spline5), but in general this interpolant performs quite well for all of the datasets tested. Nonetheless, the signal to noise ratio of the Keys interpolants is much better than bilinear.

			Computed			Time	FFT		Max Bias	
Interpolant	D	O	Order	S	R	Factor	Needed	Gradients	Error (px)	SNR (dB)
Bilinear	1	1	1.84	2	0	1.0	No	FD	0.03	25
Bicubic	3	3	3.52	3	1	3.2	No	FD	0.03	–
German5[5]	4	5	–	8	1	5.1	No	Coefs*	0.06	15
Keys3[7]	3	3	3.52	4	1	4.4	No	Coefs	0.03	38
Keys4[7]	3	4	4.78	6	1	2.9	No	Coefs	0.007	40
NLVC3	1	1	–	3	-1	8.0	No	Coefs**	–	–
NLVC5	1	1	–	5	-1	5.1	No	Coefs**	–	–
OMOMS[1]	3	4	3.94	4	-1	16.0	Yes	Coefs	0.0015	62
Deserno3[3]	2	0	1.71	3	1	1.0	No	Coefs**	0.015	–
Deserno5[3]	4	0	1.54	5	1	1.6	No	Coefs**	0.0009	–
Deserno7[3]	6	0	1.38	7	1	3.3	No	Coefs**	0.00005	–
Spline3[14]	3	0	4.59	4	2	31.5	Yes	Coefs	0.003	50
Spline5[14]	5	0	0.75	6	4	32.8	Yes	Coefs	0.0002	60

Table 1. Summary of results for various interpolants. D: Degree, O: Order (for computed order see Figure 9), S: Support size in pixels, R: Regularity (0 denotes C^0 continuity). A negative value for regularity is caused by discontinuities (or kinks) in the interpolation function. The mark "–" denotes either an unknown value or a value not computed. FD: finite difference, Coeffs: Gradient calculated using derivatives of the coefficients of the interpolant, *: Errors in the gradients are too high to be useful, **: Overly diffusive, SNR: Signal to noise ratio (higher is better).

Some comments are warranted regarding the various columns of Table 1. The degree is the polynomial degree of the basis functions. The order of accuracy defines how well the interpolated signal matches the true signal. The support size determines how many of the neighboring pixels are involved in the convolution. Larger support sizes greatly increase the computational expense. The

regularity describes how smooth the interpolated signal will be. High values of regularity mean that higher order derivatives can be computed successfully. Low values of regularity mean that the interpolated signal will be less smooth than the input data. Fast Fourier Transforms (FFT) are required in cases where the coefficients of the basis functions must be determined. In some of the interpolants the coefficients are simply the pixel values so an FFT is not required. The signal to noise ratio (SNR) defines how susceptible the interpolant is to image noise errors. Lastly, bias errors are position errors that arise in the correlation process due to the interpolant.

We have included in this study some results from using an interpolant derived from nonlocal vector calculus (NLVC) [4]. This is a particularly naive interpolant for several reasons. It has three discontinuities in the interpolation kernel which decrease smoothness. Additionally, it is also extremely diffusive and not interpolatory (the values of the interpolated signal do not match the pixel values at the pixel locations). This interpolant was included to explore ways of bringing ideas from signal processing to NLVC. It is not intended for use as a legitimate interpolant in its current form. Lastly, the German5 interpolant performs extremely bad in the image correlation context and is highly discouraged for use in DICe.

2. Motivation

The reasons this study was done are as follows:

- Bias errors are inversely dependent on the speckle size (large speckles produce smaller errors) so the bias errors computed by the authors previously, for the experimental images, are only valid for that particular speckle size. We needed to see how the bias errors are affected by various speckle sizes.
- The previous investigations used a sum-squared-difference (SSD) correlation citerion, which introduces amplitude attenuation errors as well as phase errors. In this study zero-normalized-sum-squared-differences (ZSSD) was used to remove amplitude attenuation.
- Although some reports have looked at rotation errors in and of themselves, none have studied the effect of interpolation-induced rotation errors on position error. This plays a critical role in several problems of interest to Sandia so more understanding about this is required.
- There are profound implications for the convolution-based interpolants for NLVC in terms of the kernel properties. This study has brought out a number of useful concepts from signal processing that have applications in NLVC.
- Lastly, we intended to perform a study where both the reference and deformed images are synthetically generated using analytical functions rather than apply a transformation to the reference image to get the deformed. This gives a lot better control over issues introduced by creating synthetic images.

3. Literature Survey

A number of articles from the literature were consulted in implementing the various interpolants. A wealth of research on the performance of each of these interpolants also exists. In this section we point the reader to selected works for further information.

A unifying framework for deriving interpolation methods is presented in [2]. Given the desired polynomial order, regularity, degree, and support size, this work defines a process to derive an interpolation kernel in a straight forward fashion. This is a great work to consult for background information on the properties of various kernels.

In [10] the authors explore position errors introduced into the image correlation process by various interpolants. In summary, the results showed that cubic polynomial interpolation greatly outperforms bilinear, but quintic polynomials have a diminishing return. This work also points out the importance of low-pass filtering to get rid of high frequency image content. An important finding in this work is a relative measure of strain error introduced by the interpolants. The bilinear interpolant can cause strain errors of up to 20% on average, suggesting that bilinear interpolation should be avoided. We do not consider strain errors in this present work.

An excellent theoretical background for B-spline interpolation is presented in [13] and [14]. Both of these works provide the necessary foundation for using B-spline interpolation in image procesing. In terms of the use of splines for interpolation, there are a number of publications that address optimizing the order of accuracy for the smallest support size or polynomial order. A good overview of optimizing spline-based interpolants is given in Chapter 7 of [6]. Another such work is [5] in which the short kernel fifth-order interpolation scheme is presented. Although the short kernel scheme (German5) allows for better approximation on smaller supports, in the present study we found that computing derivatives based on the optimized basis functions leads to significant errors. For this reason, the short kernel scheme is not recommended for DIC. The idea of obtaining the maximal order of interpolation for small support sizes is developed further in [1].

In [8], the authors compare a number of interpolation methods in terms of computational expense and accuracy. Although this work is applied to medical imaging and not digital image correlation, it does present a number of useful data points for comparing the various methods. Another good comparison of interpolation methods is found in [12]. In [12], a good representation of the signal to noise ratio of several interpolants is given. The most robust interpolants, in terms of noise, are the optimized maximal order on minimal support (OMOMS) interpolants.

Among the various convolution-based interpolation methods, DICe has available the third and fourth order interpolants from [7] (Keys3 and Keys4). The Keys interpolants do not require an FFT to determine the coefficients since the coefficients are simply the values of the intensity profile at the pixel locations. This leads to substantial decrease in computation time. A number of optimized version of the Keys interpolants can be derived by varying a parameter α. In this work we use $\alpha = -0.5$ throughout. A similar convolution approach to interpolation is presented in [9]. In [9] the authors investigate non-seperable imaging systems which render the Keys interpolants sub-optimal.

Lastly, in this study we have included an interpolant from the physical sciences literature used to compute electrostatic energies of charged particles on a regular grid [3]. This was done to expand the selection of interpolants beyond the typical options found in the signal processing literature. It should be remarked that this interpolant is intended for distribution of energies in the Fourier domain. It is not clear to the authors if the same FFT process used for the B-spline interpolants is applicable here. Using constant coefficients from the time domain leads to an extremely diffusive interpolant for wave numbers similar to common speckle patterns, but the resulting bias errors are

exceptionally small. The overly diffusive nature of this interpolant makes it function well in the registration process, but not for computing image gradients.

4. Synthetic Image Construction

To create the synthetic images, the following process was used:

(1) The intensity value at each pixel, I_i, in the reference image was generated using the following profile:

$$I_i = \bar{I} + 0.8\bar{I}\cos(kx/\pi)\sin(ky/\pi)$$

where $\bar{I} = 128$ (for 8-bit images black is 0 and white is 255) and k is the speckle size parameter.

(2) For the translated images the intensity profile was generated as

$$I_i = \bar{I} + 0.8\bar{I}\cos(k(x+\tilde{x})/\pi)\sin(ky/\pi)$$

where $\tilde{x}$ is the prescribed translation.

(3) For the rotated images, the intensity profile was generated using a rotation matrix, $\mathbf{R}$, where

$$R_{xx} = \cos(\theta)\ ;\ R_{xy} = \text{-}\sin(\theta)\ ;\ R_{yx} = \sin(\theta)\ ;\ R_{yy} = \cos(\theta)$$
$$x' = R_{xx}x + R_{xy}y\ ;\ y' = R_{yx}x + R_{yy}y$$

such that

$$I_i = \bar{I} + 0.8\bar{I}\cos(kx'/\pi)\sin(ky'/\pi)$$

Above, θ is the prescribed rotation.

(4) Once the analytical profiles were generated analytically, `boost` libraries were used to generate the tiff images.

In this way, interpolation is not necessary to create the synthetic images because the deformed intensity profiles are generated analytically and not from deforming pixels from the reference image. An example image is shown in Figure 1. Ordinarily the regularity of the speckle pattern would be problematic from a registration standpoint (the speckles are not unique), but the correlation process was seeded with values close enough to the exact solution that the iterative updates would not jump to a nearby solution.

To make the comparison more straightforward, a simplex method was used in the correlation instead of a gradient-based method (like the Lucas-Kinade algorithm). This removes the need to compute image gradients which are not necessarily readily available for all of the interpolants. Even though the image gradients are not used in the present study, we still evaluate how well each interpolant works for computing gradients as this is needed for DIC in general.

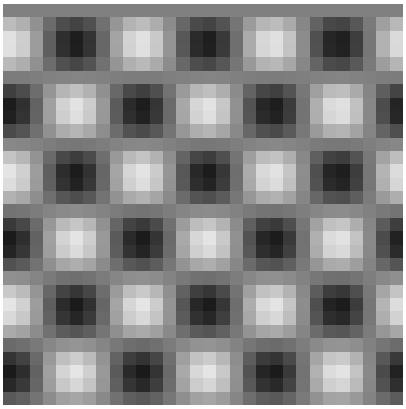


FIGURE 1. Example synthetic image with speckle size of approximately 4 pixels ($k = 2.0$).

5. PHASE ERRORS

Following the recommendations in Sutton, *et. al.* [11] regarding amplitude errors, the one-pixel shift numerical experiments were done using ZNSSD for the correlation criteria instead of SSD so that amplitude errors are removed. Figure 2 shows the position error for an incremental one-pixel shift of the reference image for various speckle sizes. Note the wave form matches the results in Sutton with the error being zero at the ends and mid-point and maximum at 0.25 and 0.75. The results show that the polynomial interpolants translation error is on the order of 0.01 to 0.03 pixels for a speckle pattern with speckles of sizes between 2 and 3 pixels. This speckle size range is typical in practice.

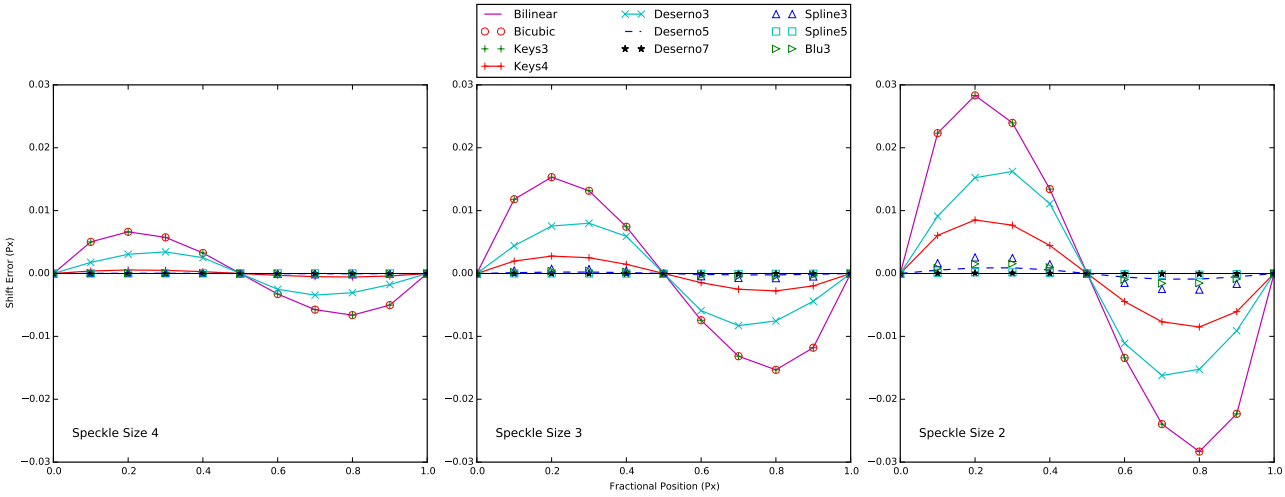


FIGURE 2. Phase error for one pixel translation.

Interestingly, the error is almost identical between bilinear and bicubic interpolation for the pixel shift example. (The results were double checked to make sure this was not a plotting error.) In

terms of bias error, the Deserno interpolants and the splines performed the best, but the fourth-order Keys interpolant significantly decreases the bias error over the polynomial interpolants.

6. Interpolation Bias for Rotations

Although the translation bias error for various interpolants has been extensively studied, less studies are available regarding interpolation bias for rotations. Figure 3 shows the rotation bias error for each of the interpolants for incremental rotations from 0 degrees to 90 degrees. The bias errors follow the familiar sin wave pattern with the errors being zero at the ends and mid-point which is expected since at 0 degress and 90 degrees the reference and deformed images match identically. The error being zero at 45 degrees is not yet fully understood. Note that for all interpolants and speckle sizes, the error is below 0.001 radians or 0.05 degrees suggesting that rotation errors are not of concern.

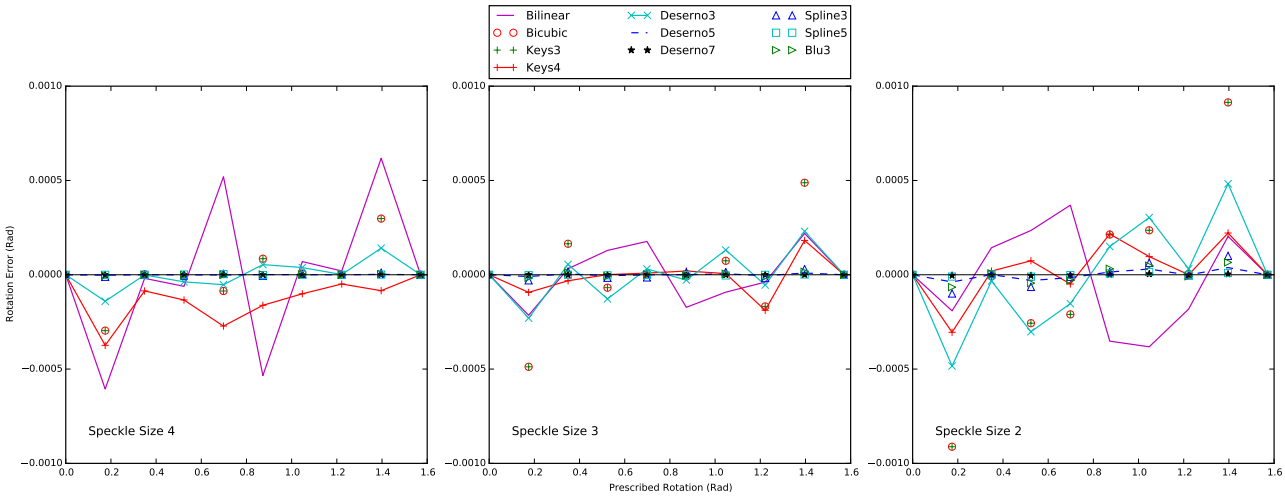


Figure 3. Interpolation bias error in the computed rotation for various rotation values.

Where the weaker interpolants suffer is in the displacement error for prescribed rotations. The displacement should be zero for this example (i.e. the position of the subset centroid does not change), but errors are introduced by the optimization process. Figure 4 shows the position error in x for the incremental rotations. The errors for y-position are similar, but not shown. In terms of position error, the bilinear interpolant performs significantly poorer that the rest of the interpolants. This is one of the main reasons bilinear interpolants should be avoided.

7. Interpolation Order Accuracy

The accuracy of each of the interpolants in terms of how well they represent the true continuous solution, for the cases of speckle size 4 and 2, is shown in Figures 5 through 8. While the polynomial, splines, Keys and OMOMS interpolants seem to capture the signal well, the NLVC and German5 interpolants introduce discontinuities. The Deserno interpolants exhibit an overly diffusive characteristic. Note that even if the particular interpolation scheme in question has a

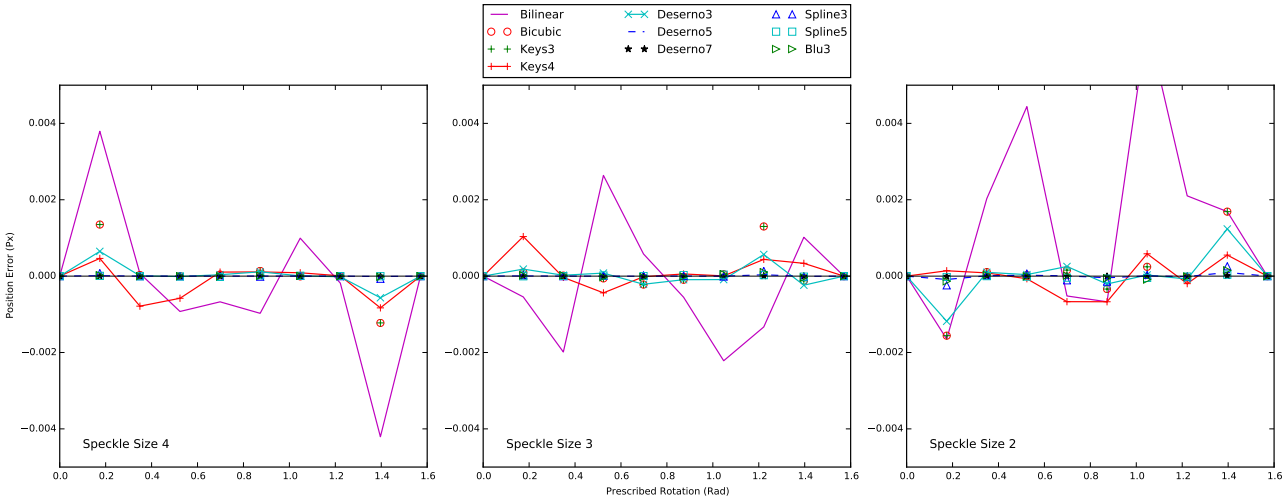


FIGURE 4. Interpolation bias error in position for various rotation values.

higher order of convergence, the error for a given speckle size to pixel size ratio may be larger than an interpolant of lower order. The order of convergence determines how the error decreases with increasing resolution, but not the initial error value.

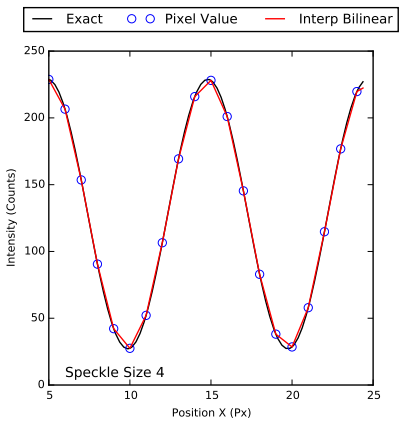
The convergence rate for selected interpolants is shown in Figure 9. The slope of each of the lines in the log-log plot determines the computed convergence rate. These values are given in Table 1. Since for image analysis, the pixel size is fixed, the convergence study in this plot was conducted by altering k such that the wave profile of the prescribed intensity became more and more smooth. While this is not a formal convergence study it does shed some light on the expected order of accuracy of the various interpolants. In general, while the measured convergence order is high for the optimized interpolants, as the resolution improves, the order decreases significantly. This suggests that there is a diminishing return from the optimized interpolants as the wave frequency decreases (large speckle sizes).

8. Computing Image Gradients

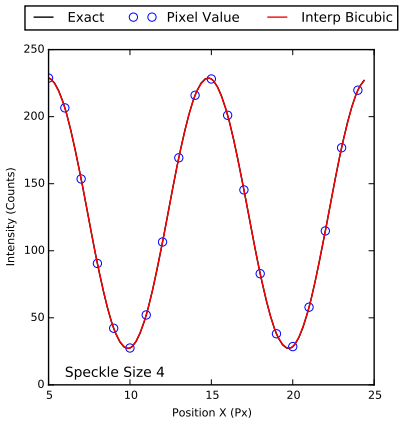
A similar study was performed for computing the gradients of the prescribed intensity profile. The accuracy of each of the interpolants for computing image gradients for the cases of speckle size 4 and 2 is shown in Figures 10 and 11. The spline, Keys, and OMOMS interpolants perform the best for computing image gradients, while the Deserno and NLVC interpolants once again exhibit their overly diffusive characteristics.

9. Execution Speed

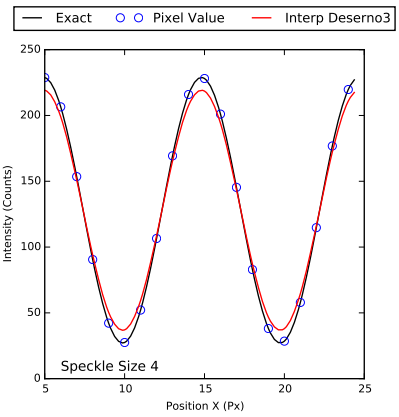
For a single subset, the execution speed is fastest for the bilinear interpolation. If this is treated as the standard (1.0), the comparative cost of all of the interpolants is shown in Figure 12. The computation time includes the total cost to converge to the solution (all iterations). Thus, even if one interpolant can be computed quicker for a single iteration the overall cost may be more than another because more iterations may be required. The results in Figure 12 show that the



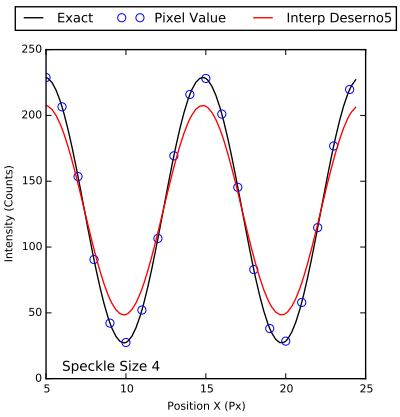
(A) Bilinear



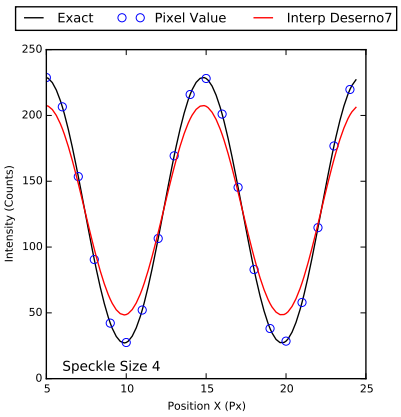
(B) Bicubic



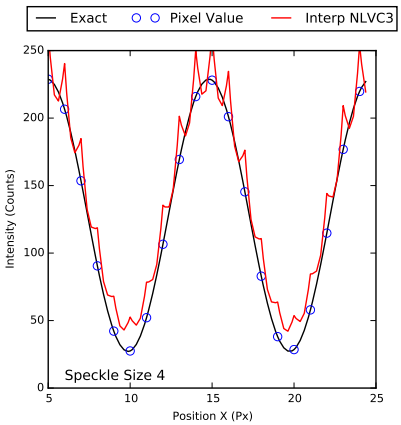
(C) Deserno3



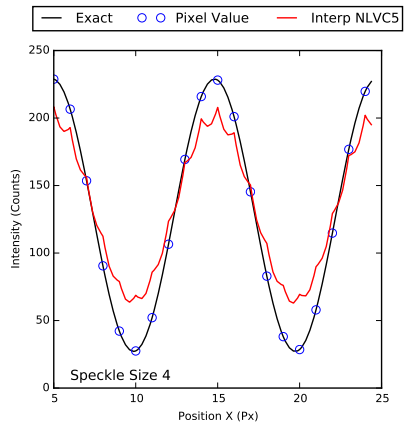
(D) Deserno5



(E) Deserno7

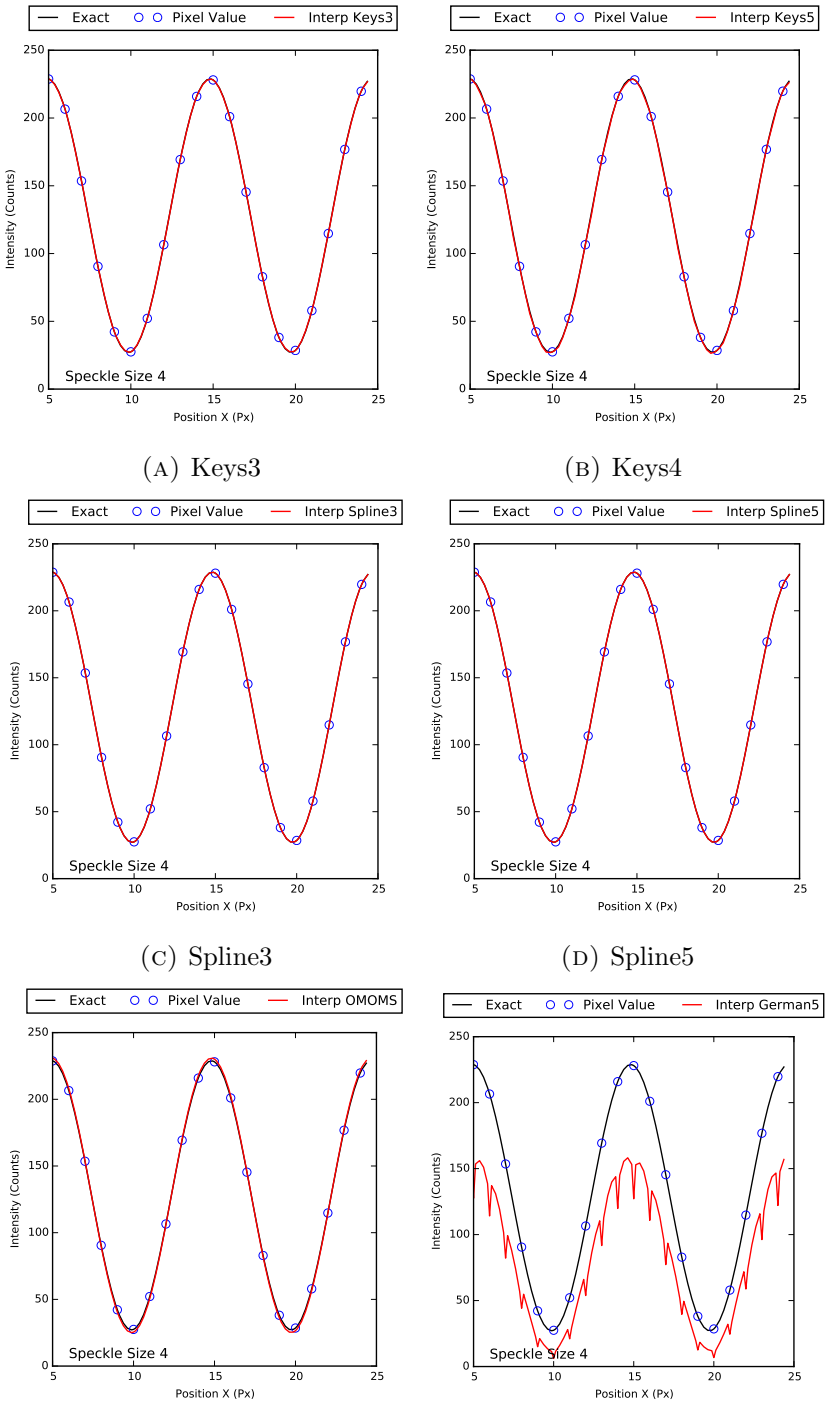


(F) NLVC3



(G) NLVC5

FIGURE 5. Interpolation of prescribed function $I(x, y)$, Part I: Polynomial, Deserno and NLVC with speckle size 4

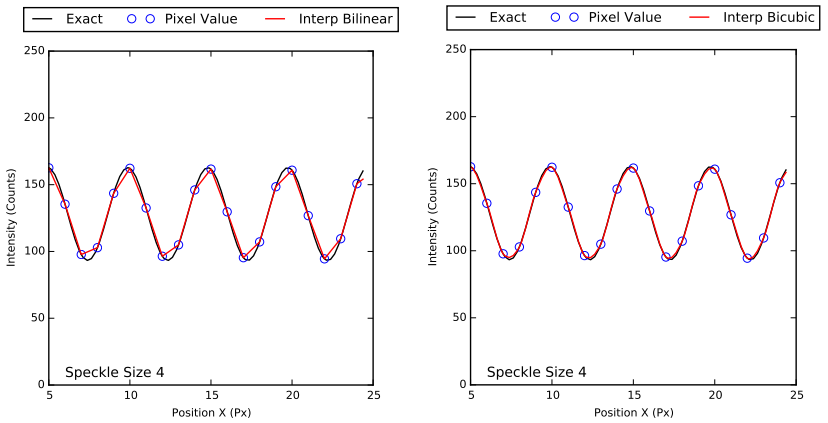


(A) Keys3 (B) Keys4

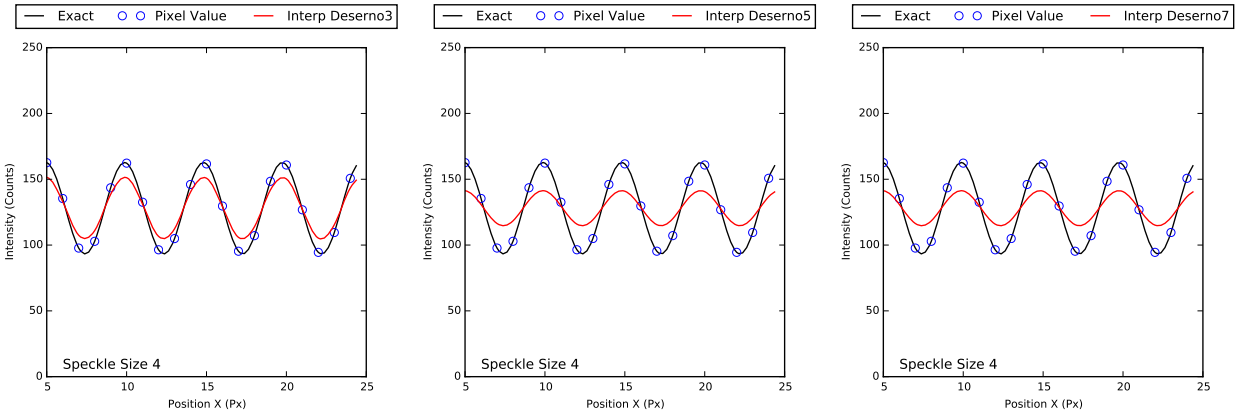
(C) Spline3 (D) Spline5

(E) OMOMS (F) German5

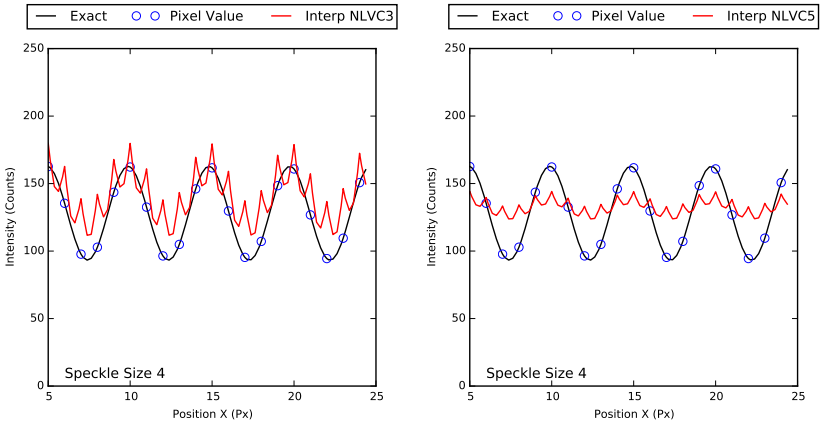
FIGURE 6. Interpolation of prescribed function $I(x, y)$, Part 2: Keys, Splines, OMOMS and German with speckle size 4



(A) Bilinear (B) Bicubic

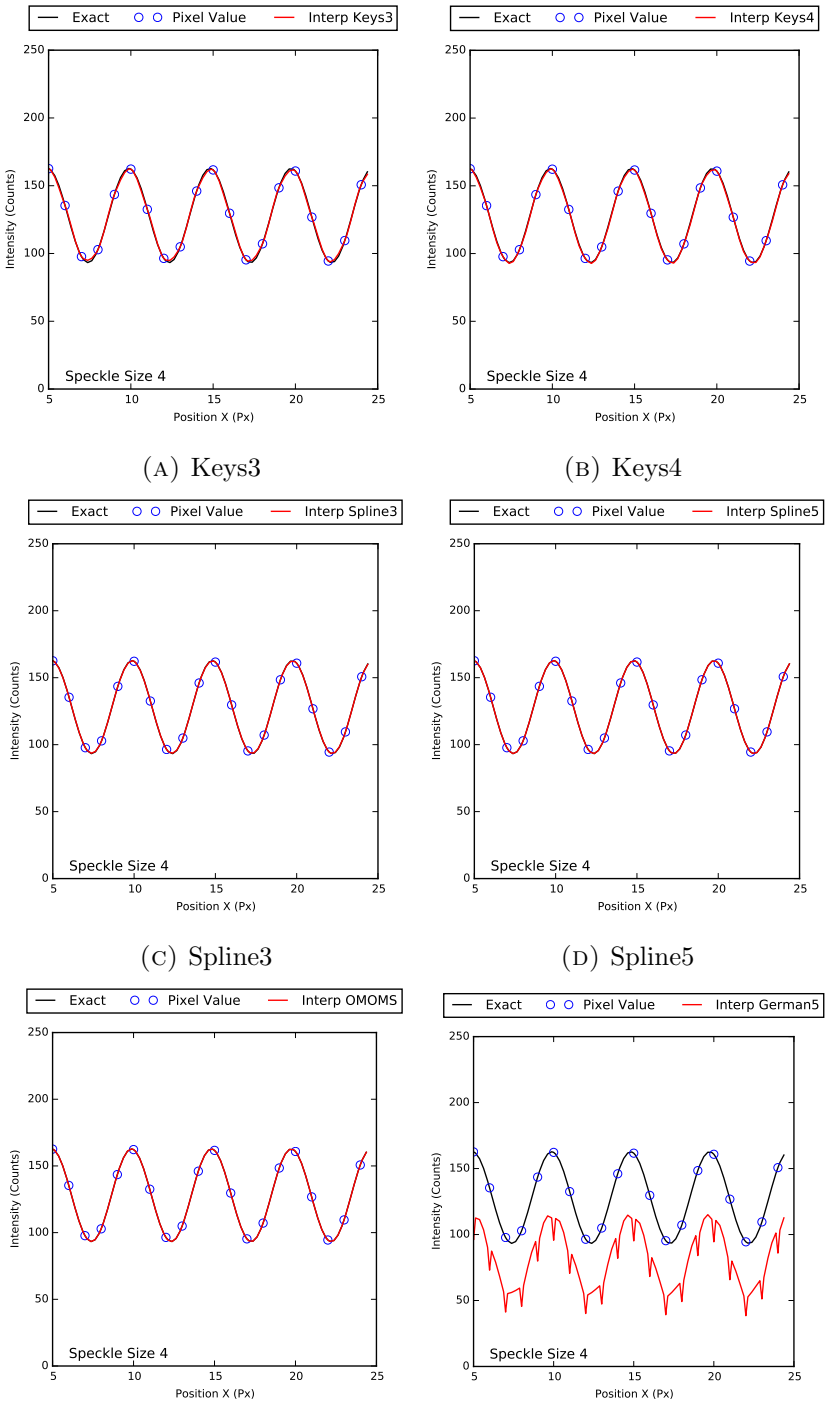


(C) Deserno3 (D) Deserno5 (E) Deserno7



(F) NLVC3 (G) NLVC5

FIGURE 7. Interpolation of prescribed function $I(x, y)$, Part I: Polynomial, Deserno and NLVC with speckle size 2



(A) Keys3 (B) Keys4

(C) Spline3 (D) Spline5

(E) OMOMS (F) German5

FIGURE 8. Interpolation of prescribed function $I(x, y)$, Part 2: Keys, Splines, OMOMS and German with speckle size 2

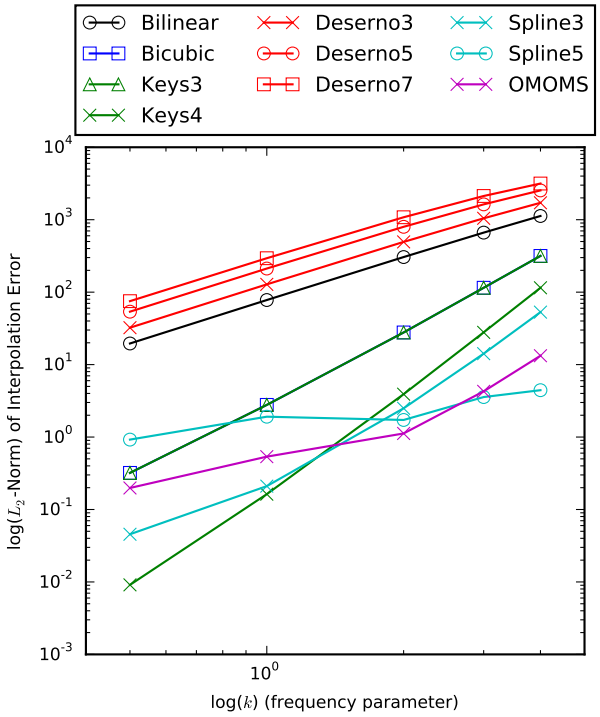
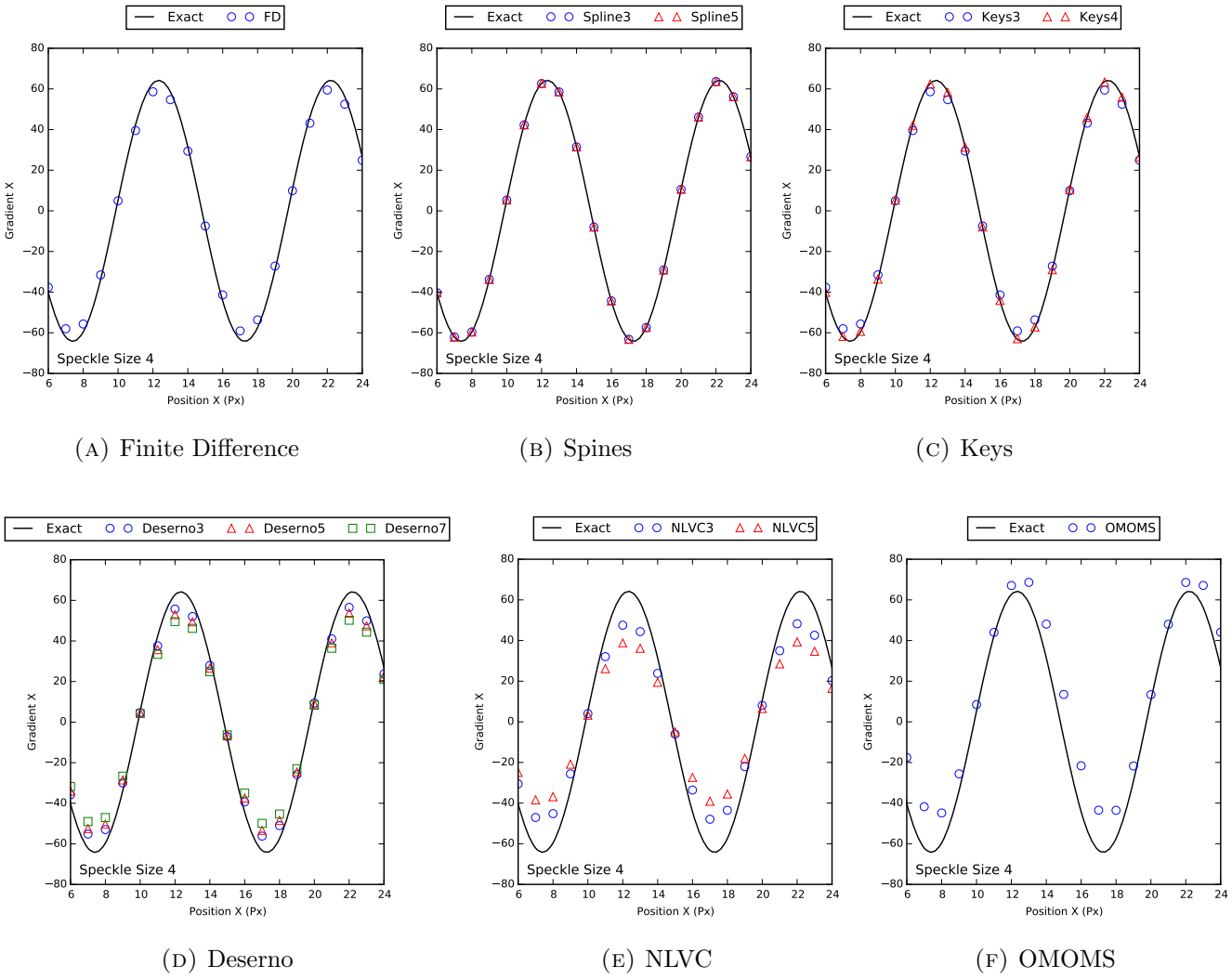


FIGURE 9. Log-log plot of the L_2-norm of the interpolation error for increasing k (the wave frequency parameter). By interpolation error we refer to the difference between the interpolated values of I, and the exact value of the intensity profile. Note that for small pixel sizes, the order of accuracy significantly degrades for the quintic spline, fourth order keys, and OMOMS interpolants.

spline interpolants are extremely expensive to compute. Typically, the argument is given that DIC takes place off-line such that this expense is not an issue, but as DIC applications evolve (e.g. as an integrated part of an experiment itself) computation expense becomes a critical factor. The OMOMS interpolants are about half as expensive as the splines, but the gradients computed using the OMOMS interpolants are not as smooth.

10. Conclusion

While it is not easy to say with absolute confidence which interpolant is best in all cases, this document has outlined several of the strengths and weaknesses of each of the interpolants implemented in DICe. It is intended that an analyst can use this information to choose the right interpolant for a given application. For example if the noise content is high, the spline interpolants may be worth the computational expense in exchange for a high signal to noise ratio. Clearly, the bilinear interpolants cause a number of undesirable effects and should be avoided. In general, the fourth-order Keys interpolant performs best given all the factors considered above and should be the default interpolant.



(A) Finite Difference (B) Spines (C) Keys

(D) Deserno (E) NLVC (F) OMOMS

FIGURE 10. Image gradients calculated using various interpolants with speckle size 4

REFERENCES

[1] T. Blu, P. Thévenaz, and M. Unser. MOMS: Maximal-order interpolation of minimal support. *IEEE Transactions on Image Processing*, 10:1069–1080, 2001.

[2] T. Blu, P. Thévenaz, and M. Unser. Complete parameterization of piecewise-polynomial interpolation kernels. *IEEE Transactions on Image Processing*, 12:1297–1309, 2003.

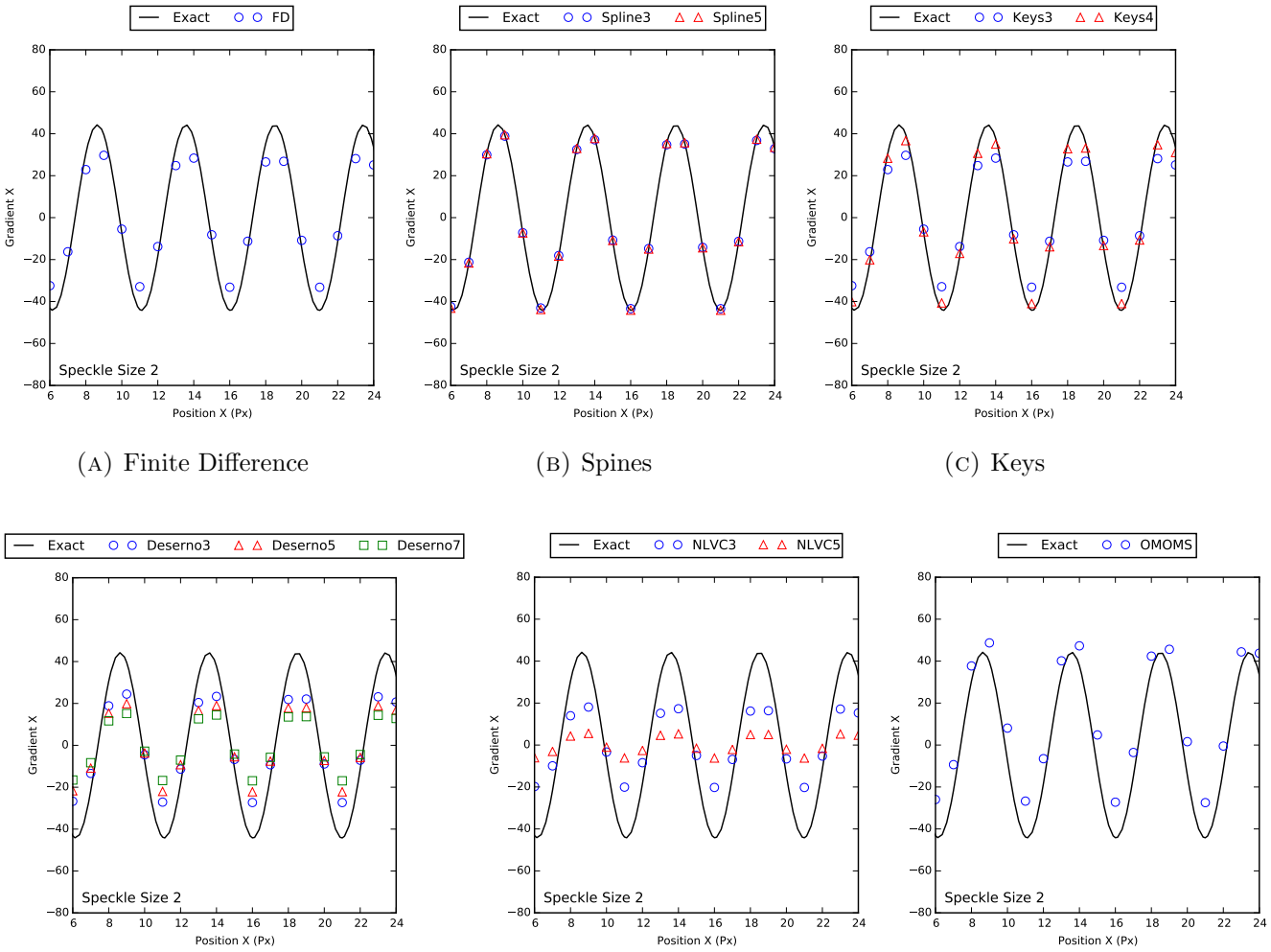
[3] M. Deserno and C. Holm. How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines. *Journal of Chemical Physics*, 109:7678–7693, 1998.

[4] Q. Du, M. Gunzburger, R. B. Lehoucq, and K. Zhou. A nonlocal vector calculus, nonlocal volume-constrained problems, and nonlocal balance laws. *Mathematical Models and Methods in Applied Sciences*, 23(3):493–540, 2013.

[5] I. German. Short kernel fifth-order interpolation. *IEEE Transactions on Signal Processing*, 45:1355–1359, 1997.

[6] S. Howison. *Advances in Signal Transforms*. Hindawi Publishing Corporation, New York, NY, 2007.

[7] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on acoustics, speech and signal processing*, 29:1153–1160, 1981.



(A) Finite Difference (B) Spines (C) Keys

(D) Deserno (E) NLVC (F) OMOMS

FIGURE 11. Image gradients calculated using various interpolants with speckle size 2

[8] J. A. Parker, R. V. Kenyon, and D. E. Troxel. Comparison of interpolating methods for image resampling. *IEEE Transactions on Medical Imaging*, 2:31–39, 1983.

[9] S. E. Reichenbach and F. Geng. Two-dimensional cubic convolution. *IEEE Transactions on Image Processing*, 12:857–865, 2003.

[10] H. W. Schreier, J. R. Braasch, and M. A. Sutton. Systematic erros in digital image correlation caused by intensity interpolation. *Optical Engineering*, 39:2915–2921, 2000.

[11] M. A. Sutton, J. J. Orteu, and H. Schreier. *Image Correlation for Shape, Motion and Deformation Measurements: Basic Concepts, Theory and Applications*. Springer, New York, USA, 2009.

[12] P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Transactions on medical imaging*, 19:739–758, 2000.

[13] M. Unser, A. Aldroubi, and M. Eden. B-Spline signal processing: Part I–Theory. *IEEE Transactions on Signal Processing*, 41:821–833, 1993.

[14] M. Unser, A. Aldroubi, and M. Eden. B-Spline signal processing: Part II–Efficient design and applications. *IEEE Transactions on Signal Processing*, 41:834–848, 1993.

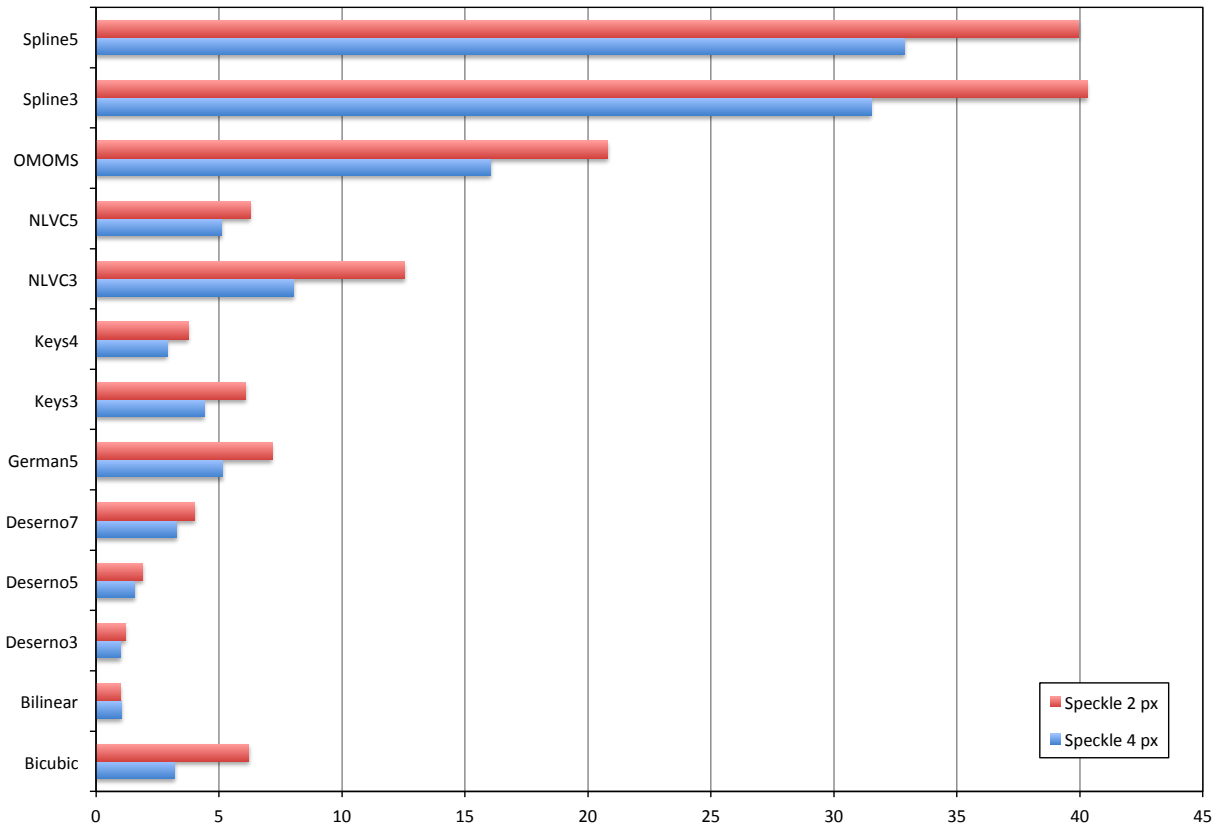


FIGURE 12. Comparison of computation cost for various interpolants.