

Image Coregistration in SAR Interferometry only by means of Arithmetic Operations

Jesús Selva and Juan M. Lopez-Sanchez
DFISTS, EPS, University of Alicante, P.O. Box 99, Alicante, Spain
Email: jesus.selva@ua.es, juanma-lopez@ieee.org

Abstract—An alternative interpolation technique for SAR image coregistration in interferometric processing is formulated and described. The proposed algorithm is based on the 1-D Farrow interpolator and, when combined with an adequate implementation, it involves a smaller computational burden than the conventional method and yields high accuracy. Basically, this technique enables to carry out the coregistration without resorting to a set of functional weights, and it only requires arithmetic operations and 2-D FFTs. This paper includes several results and comparisons that confirm its validity: the proposed technique is combined with two different polynomial interpolation procedures (Taylor and Chebyshev); bounds on its interpolation error in the 1-D and 2-D cases are derived; and, additionally, it is tested numerically with a synthetic image.

I. INTRODUCTION

One of the first steps in the processing chain required in SAR interferometry is the coregistration of two SAR images. Basically, the coregistration consists of interpolating a slave image at a set of locations which correspond to the pixel locations of a different master image. Usually, this interpolation is carried out by means of a space-domain convolution with a kernel. Specifically, the slave image value at a given position is interpolated using a weighted sum of the near-by samples, in which the weights are kernel samples.

One of the problematic aspects of this procedure is how to obtain the required sum weights (all kernel samples for each location). Since to compute them as they are needed would be very time consuming, the usual approach consists of pre-computing the kernel in a very fine grid of positions and storing the resulting values in a table. Then, during the coregistration, to obtain a given weight only requires a look-up operation. However, in applications in which extreme accuracy is required, like advanced differential SAR interferometry with permanent scatterers, this approach is too cumbersome since the size of the stored data grows exponentially with the required accuracy.

In this paper, we apply a different approach for this problem. The rationale of this interpolator is to approximate every kernel sample involved in the space-domain interpolation by a low-order polynomial in the fractional displacement variable. Once these polynomials have been inserted into the usual convolution formula, it turns out that one may switch the convolution and polynomial evaluation operations. More precisely,

This work has been supported by the Spanish Ministry of Education and Science (MEC) and EU FEDER, under Project TEC2005-06863-C02-02, and under the “Ramón y Cajal” Programme.

if one considers the sequences formed by the polynomial coefficients of equal order, then one would convolve the signal with each sequence in this set. The result is a set of polynomials in the fractional displacement variable, one for each sample position, that accurately approximate the signal in consecutive intervals of length equal to the sampling period. With a proper implementation, this interpolation exhibits the following features: it does not require the evaluation of any kernel sample; it involves less floating-point operations (flops) than the usual convolution method; its memory requirements are small; and it is much faster than the usual method if the image has to be over-sampled.

The formulation of this method is explained in Section II for the one-dimensional case, and it is extended to the two-dimensional case in Section IV. Its performance for InSAR coregistration is analyzed in Section V.

II. INTERPOLATION OF ONE-DIMENSIONAL SIGNALS BY MEANS OF THE FARROW STRUCTURE

Given a band-limited 1-D signal $s(t)$ which is known at $t = nT$ for integer n , the conventional interpolation formula has the form

$$s(t) = \sum_{p=-P}^P s((n-p)T)g_p(u) + \theta_o(t), \quad (1)$$

under the following assumptions or definitions:

- The variable t is decomposed modulo T as $t = nT + u$ with integer n and $-1/2T \leq u < 1/2T$.
- $s(t)$ has two-sided bandwidth B and Nyquist condition $BT < 1$ holds.
- $g_p(u)$ is a set of functions in the fractional shift u and $\theta_o(t)$ is the interpolation error.

As is shown in [1]–[3], there are sets of weights $g_p(u)$ for which the interpolation error is bounded, provided there is a bound A_s on the signal samples, $|s(t)| \leq A_s$.

The Farrow interpolator in [4] has been recently applied to this interpolation problem in [5], [6]. When compared with the conventional interpolation method in (1), this technique eliminates the need to compute or store any set of functional weights $g_p(u)$, and exhibits a smaller computational burden.

Basically, the Farrow interpolator is the result of substituting into (1) a set of polynomial approximations for the weights $g_p(u)$. Let $\tilde{g}_p(u)$ denote the polynomial approximation for

$g_p(u)$ with error $\phi_p(u)$; i.e., $\tilde{g}_p(u)$ is the polynomial

$$\tilde{g}_p(u) \equiv \sum_{q=1}^Q b_{p,q} u^{q-1}, \quad p = -P, \dots, P, \quad (2)$$

and $g_p(u)$ and $\tilde{g}_p(u)$ are related by

$$g_p(u) = \tilde{g}_p(u) + \phi_p(u). \quad (3)$$

The Farrow interpolator is derived by substituting this last equation into (1) and switching the summations:

$$\begin{aligned} s(t) &= \sum_{p=-P}^P s((n-p)T) \left(\sum_{q=1}^Q b_{p,q} u^{q-1} + \phi_p(u) \right) + \theta_o(t) \\ &= \sum_{q=1}^Q u^{q-1} \sum_{p=-P}^P s((n-p)T) b_{p,q} \\ &\quad + \left(\sum_{p=-P}^P s((n-p)T) \phi_p(u) + \theta_o(t) \right) \end{aligned} \quad (4)$$

Note that the inner summation in (4) is actually the convolution of the sequence $s[n] \equiv s(nT)$ with the sequence

$$b_q[n] \equiv \begin{cases} b_{n,q} & \text{if } -P \leq n < P \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

So, it follows from (4) that it is possible to compute first the convolutions $(s * b_q)[n]$ through fast convolution techniques, and then interpolate $s(t)$ by simply evaluating a $(Q-1)$ -order polynomial. In short, the Farrow interpolator technique consists of the following steps:

- 1) Compute the convolutions

$$s_q[n] \equiv (s * b_q)[n] \quad (6)$$

for $q = 1, \dots, Q$.

- 2) Decompose each interpolation abscissa t modulo- T in the form $t = nT + u$, with $n = \lfloor t/T + 1/2 \rfloor$ (floor function) and $u = t - nT$.
- 3) The interpolated value is

$$s(t) \approx \sum_{q=1}^{Q-1} s_q[n] u^q. \quad (7)$$

This interpolation procedure requires to convolve the sequence $s[n]$ with the Q sequences $b_q[n]$. Since these last sequences are known a priori, these convolutions only involve $Q+1$ FFTs if they are carried out in the following way:

- 1) Add zero padding to $s[n]$ and compute its FFT, $S[k]$.
- 2) Multiply $S[k]$ by each of the FFTs of $b_q[n]$, $B_q[k]$. Note that the sequences $B[k]$ can be pre-computed since they are known a priori. The result is the set of sequences

$$S_q[k] \equiv S[k] B_q[k]. \quad (8)$$

- 3) Compute the inverse FFT of the sequences $S_q[k]$ in order to obtain the polynomial coefficients $s_q[n]$ in (7).

III. BOUND ON THE INTERPOLATION ERROR IN THE ONE-DIMENSIONAL CASE

Let $\theta(t)$ denote the error term in (4). A bound on $\theta(t)$ can be easily derived from equations (2) and (3). If $|\theta_o(t)|$ is bounded by $A_s \epsilon_T$ for a constant ϵ_T (truncation error), then

$$\begin{aligned} |\theta(t)| &= \left| s(t) - \sum_{p=-P}^P s((n-p)T) \tilde{g}_p(u) \right| \\ &= \left| s(t) - \sum_{p=-P}^P s((n-p)T) (g_p(u) - \phi_p(u)) \right| \\ &= \left| \theta_o(t) - \sum_{p=-P}^P s((n-p)T) \phi_p(u) \right| \\ &\leq |\theta_o(t)| + \left| \sum_{p=-P}^P s((n-p)T) \phi_p(u) \right| \\ &\leq A_s \epsilon_T + A_s \sum_{p=-P}^P |\phi_p(u)|. \end{aligned} \quad (9)$$

If $\epsilon_P(u)$ denotes the summation

$$\epsilon_P(u) \equiv \sum_{p=-P}^P |\phi_p(u)|, \quad (10)$$

and ϵ_P is an upper bound on $\epsilon_P(u)$ for $-T/2 \leq u < T/2$,

$$\epsilon_P \geq \max_u \epsilon_P(u), \quad (11)$$

then the total error can be bounded as

$$|\theta(t)| \leq A_s (\epsilon_T + \epsilon_P). \quad (12)$$

So, the polynomial interpolation has increased the error bound by $A_s \epsilon_P$. The value of ϵ_P depends on the selected interpolation scheme as described in the next two sub-sections.

A. Specific bound assuming Taylor polynomials

Assume that the polynomials in (2) are truncated Taylor expansions of order $Q-1$ at $u=0$. Then, it follows from Taylor's theorem that the approximation error is

$$\phi_p(u) = \frac{g_p^{(Q)}(\gamma)}{Q!} u^Q \quad (13)$$

for a γ that depends on u , $|\gamma| \leq T/2$. In practice, the weight functions $g_p(u)$ are bounded by a constant A_g and they are band-limited with two-sided bandwidth B_g . If this is assumed, Bernstein's inequality [7, chap.6] provides a bound on $|g_p^{(Q)}(\gamma)|$,

$$|g_p^{(Q)}(\gamma)| \leq (\pi B_g)^Q A_g, \quad (14)$$

and then $\max_{p,u} |\phi_p(u)|$ can be bounded as

$$\max_{p,u} |\phi_p(u)| \leq \frac{A_g}{Q!} \left(\frac{\pi B_g T}{2} \right)^Q. \quad (15)$$

So, in (2), it follows that

$$\max_u \sum_{p=-P}^P |\phi_p(u)| \leq \frac{A_g (2P+1)}{Q!} \left(\frac{\pi B_g T}{2} \right)^Q, \quad (16)$$

and the right side of this inequality is a valid value for ϵ_P .

This bound demonstrates that the polynomial interpolation procedure cannot fail, i.e., that it is possible to find a priori bounds for the polynomial interpolation error for any bounded band-limited signal. However, in practice, the numerical approximation through Chebyshev polynomials, which is addressed in the next subsection, yields better accuracy.

B. Bound for Chebyshev approximation assuming a specific pulse

In order to assess the performance of the Chebyshev approximation, it is necessary to specify the interpolation pulse. So, consider the formula in (1) with functional weights given by $g_p(u) \equiv g(pT + u)$, where $g(t)$ is the Approximate Prolate pulse in [3]. In this reference, the error is bounded by $A_s / \sinh(P\pi(1 - BT))$. For ERS images, it is $B = 0.82/T$. If $P = 10$ then this bound is

$$A_s / \sinh(P\pi(1 - BT)) = A_s \cdot 7 \cdot 10^{-3}. \quad (17)$$

Figure 1(a) shows the polynomial interpolation error for this pulse for all sub-ranges of the form $[-T/2 + pT, T/2 + pT]$ for $-P \leq p \leq P$. The error is smaller than 10^{-4} in the whole range. On the other hand, figure 1(b) shows the bound on the polynomial interpolation error $\phi_p(u)$ for several polynomial orders. For $Q = 5$, $\phi_p(u)$ is smaller than the bound in (17) for the truncation error by roughly factor 2.5.

IV. EXTENSION TO THE TWO-DIMENSIONAL CASE

The interpolation procedure derived in the previous section can be easily extended to the 2-D case. If $s(x, y)$ denotes a 2-D signal with two-sided bandwidth B in both variables, then it can be interpolated through a formula of the form

$$s(x, y) = \sum_{q_x, q_y=1}^Q s_{q_x, q_y}[n_x, n_y] u_x^{q_x-1} u_y^{q_y-1} + \theta(x, y), \quad (18)$$

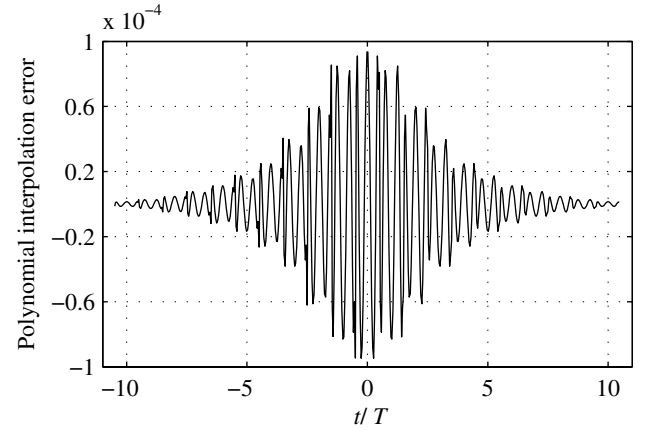
assuming the following:

- D is a sampling period fullfilling $BD < 1$.
- The modulo- D decompositions of x and y are $n_x \equiv \lfloor x/D \rfloor$, $u_x \equiv x - n_x D$, $n_y \equiv \lfloor y/D \rfloor$, and $u_y \equiv y - n_y D$.
- $s_{q_x, q_y}[n_x, n_y]$ is the 2-D equivalent of $s_q[n]$

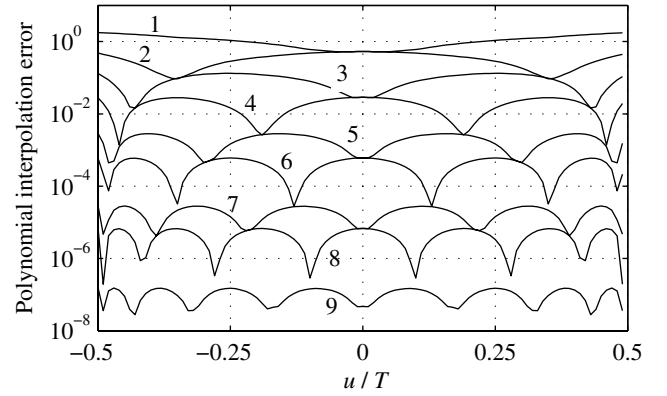
$$s_{q_x, q_y}[n_x, n_y] \equiv \sum_{p_x, p_y} s[n_x - p_x, n_y - p_y] b_{q_x}[p_x] b_{q_y}[p_y]. \quad (19)$$

- The polynomial coefficients $b_q[n]$ are the same as in the one-dimensional case.

Regarding the computational burden of the convolutions in (19), it can be shown that only $Q^2 + Q + 1$ FFTs along one dimension are required. This justifies the fact that the proposed method is less complex than the conventional one, even if the functional weights have been pre-computed in the latter. For implementation details see [6].



(a) Error in approximating Knab's pulse in range $[-T/2 - PT, T/2 + PT]$; i.e., the function plotted is $\phi_p(u)$ with $p = \lfloor (t + 1/2)/T \rfloor$ and $u = t - pT$.



(b) Bound on polynomial interpolation error.

Fig. 1. 1(a) Polynomial interpolation error in range $[-T/2 - PT, T/2 + PT]$, and 1(b) bound on the polynomial interpolation error

A. Bound on 2-D interpolation error

The interpolation formula for the 2-D signal $s(x, y)$ in the x variable is, from (1),

$$s(x, y) = \sum_{p_x=-P}^P s((n_x - p_x)D, y) \tilde{g}_p(u_x) + \theta_1(x, y), \quad (20)$$

where $x = n_x D + u_x$, n_x is an integer and $-D/2 \leq u_x < D/2$. In the same way, the interpolation formula in the y variable is

$$s(x, y) = \sum_{p_y=-P}^P s(x, (n_y - p_y)D) \tilde{g}_p(u_y) + \theta_2(x, y) \quad (21)$$

with $y = n_y D + u_y$, integer n_y and $-D/2 \leq u_y < D/2$. The signal $s((n_x - p_x)D, y)$ in (20) can be interpolated in the y variable, simply by substituting $x = (n_x - p_x)D$ in (21),

$$s((n_x - p_x)D, y) = \sum_{p_y=-P}^P s((n_x - p_x)D, (n_y - p_y)D) \tilde{g}_p(u_y) + \theta_2((n_x - p_x)D, y). \quad (22)$$

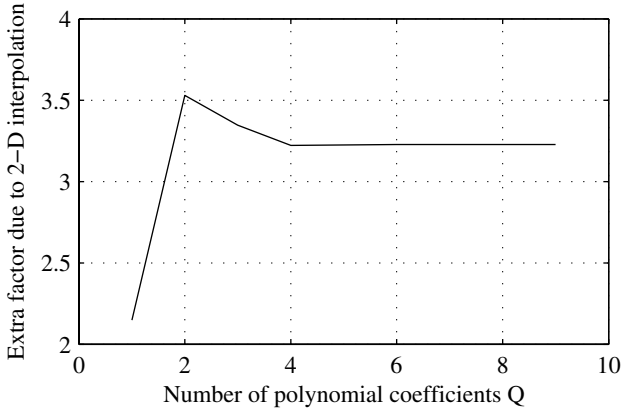


Fig. 2. Extra factor due to the two-dimensional interpolation in (25), $A_{\tilde{g}}$, versus Q for $P = 10$ and $B = 0.82/T$.

So, substituting this expression into (20), the following 2-D interpolation formula follows

$$\begin{aligned} s(x, y) &= \sum_{p_x} \left[\sum_{p_y} s((n_x - p_x)D, (n_y - p_y)D) \tilde{g}_{p_y}(u_y) \right. \\ &\quad \left. + \theta_2((n_x - p_x)D, y) \right] \tilde{g}_{p_x}(u_x) + \theta_1(x, y) \\ &= \sum_{p_x} \sum_{p_y} s((n_x - p_x)D, (n_y - p_y)D) \tilde{g}_{p_y}(u_y) \tilde{g}_{p_x}(u_x) \\ &\quad + \sum_{p_x} \theta_2((n_x - p_x)D, y) \tilde{g}_{p_x}(u_x) + \theta_1(x, y). \end{aligned} \quad (23)$$

Now, $\theta_1(x, y)$ and $\theta_2(x, y)$ are bounded by $A_s(\epsilon_T + \epsilon_P)$. If $\theta(x, y)$ denotes the last line in (23), this error can be bounded in the following way:

$$\begin{aligned} |\theta(x, y)| &= \left| \sum_{p_x} \theta_2((n_x - p_x)D, y) \tilde{g}_{p_x}(u_x) \right. \\ &\quad \left. + \theta_1(x, y) \right| \leq A_s(\epsilon_T + \epsilon_P) \left(1 + \sum_{p_x} |\tilde{g}_{p_x}(u_x)| \right). \end{aligned} \quad (24)$$

If $A_{\tilde{g}}$ is the maximum value of the last factor, $A_{\tilde{g}} \equiv 1 + \max_{u_x} \sum_{p_x=-P}^P |\tilde{g}_{p_x}(u_x)|$, the final bound yields

$$|\theta(x, y)| \leq A_s(\epsilon_T + \epsilon_P) A_{\tilde{g}}. \quad (25)$$

Figure 2 shows the value of $A_{\tilde{g}}$ as a function of Q for $P = 10$ and $B = 0.82/T$. Note that the precision lost relative to the one-dimensional case is only factor 3.2.

V. COMPUTATIONAL BURDEN VERSUS ACCURACY

The relationship between the numerical accuracy and the computational burden has already been analyzed in [6]. For the numerical example in this paper ($P = 10$, $B = 0.82/T$), figure 3 shows the number flops per interpolated point for the proposed and the conventional methods. The image had size $L_x \times L_y = 200 \times 200$ and was formed by randomly placing 80 sinc pulses of amplitude one and random phase on the image extent. Then, it was interpolated at $\alpha^2 L_x L_y$ locations.

From this figure it is clear that the number of flops per interpolated point decreases dramatically with α (coregistration ratio). With $Q = 5$, the numerical trials show that

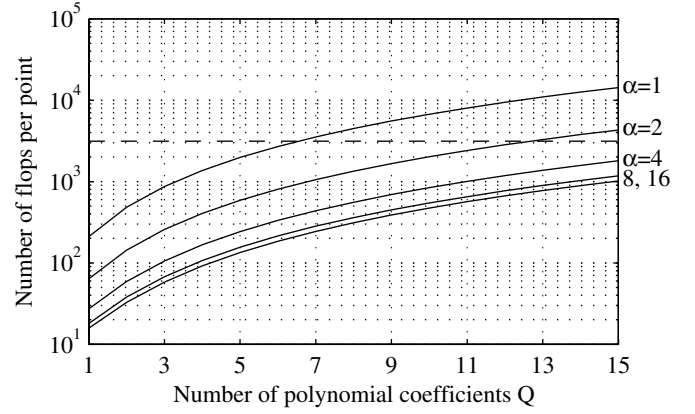


Fig. 3. Computational burden per interpolated point for several coregistration ratios α . The dashed horizontal line is the cost of the conventional method.

the conventional (dashed line) and the proposed interpolation methods provide similar accuracies, but the proposed one requires less flops even for $\alpha = 1$ (no over-sampling). In addition, it must be noted that this comparison is biased in favor of the conventional method in two ways. First, the cost of computing the functional weights in the conventional method is taken as zero. And second, the cost of one N -point FFT is assumed to be $5N \log_2 N$, but there are faster algorithms like the split-radix FFT in [8] where its cost is reduced to $3.9N \log_2 N$.

VI. CONCLUSION

An interpolation method for the coregistration of SAR images has been presented. It can be regarded as an extension of the Farrow interpolator to the two-dimensional case. The proposed technique is useful for any given kernel, but eliminating the need to store or evaluate any kernel sample. Its implementation only involves 2-D FFTs and arithmetic operations. It is shown in this paper that the computational burden of this technique is smaller than the one of the conventional spatial convolution.

REFERENCES

- [1] H. D. Helms, "Truncation error of sampling theorem expansions," *Proceedings of the IRE*, pp. 179–184, Feb 1962.
- [2] K. E. Grue, "Optimal reconstruction of bandlimited bounded signals," *IEEE Trans. Inform. Theory*, vol. IT-31, no. 5, pp. 594–601, Sep. 1985.
- [3] J. J. Knab, "Interpolation of band-limited functions using the approximate prolate series," *IEEE Trans. Inform. Theory*, vol. IT-25, no. 6, pp. 717–720, Nov. 1979.
- [4] C. W. Farrow, "A continuously variable digital delay element," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Espoo, Finland, June 1988, pp. 2641–2645.
- [5] J. Selva, "Interpolation of bounded band-limited signals and applications," *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4244–4260, Nov. 2006.
- [6] J. Selva and J. M. Lopez-Sanchez, "Efficient interpolation of SAR images for coregistration in SAR interferometry," to appear in the *IEEE Geoscience and Remote Sensing Letters*.
- [7] J. R. Higgins, *Sampling Theory in Fourier and signal analysis. Foundations*. Oxford Science Publications, 1996.
- [8] S. G. Johnson, "A modified Split-Radix FFT with fewer arithmetic operations," *IEEE Trans. Signal Processing*, vol. 55, no. 1, pp. 111–119, Jan 2007.