An Overflow Free Fixed-point Eigenvalue Decomposition Algorithm: Case Study of Dimensionality Reduction in Hyperspectral Images

Bibek Kabi, Anand S Sahadevan, Tapan Pradhan

HAL Id: hal-01635958
https://hal.inria.fr/hal-01635958

Submitted on 30 Nov 2017

An Overflow Free Fixed-point Eigenvalue Decomposition Algorithm: Case Study of Dimensionality Reduction in Hyperspectral Images

Bibek Kabi[1] and Anand S Sahadevan[2] and Tapan Pradhan[3]

[1]Laboratoire d'Informatique de l'Ecole Polytechnique, CNRS, Université Paris-Saclay, 91128 Palaiseau, France
[2]Space Application Center, Indian Space Research Organisation, India
[3]Techno India University, West Bengal, India
[1]bibek@lix.polytechnique.fr, [2]anandss@sac.isro.gov.in, [3]hod.ee.tiu@gmail.com

Abstract— We consider the problem of enabling robust range estimation of eigenvalue decomposition (EVD) algorithm for a reliable fixed-point design. The simplicity of fixed-point circuitry has always been so tempting to implement EVD algorithms in fixed-point arithmetic. Working towards an effective fixed-point design, integer bit-width allocation is a significant step which has a crucial impact on accuracy and hardware efficiency. This paper investigates the shortcomings of the existing range estimation methods while deriving bounds for the variables of the EVD algorithm. In light of the circumstances, we introduce a range estimation approach based on vector and matrix norm properties together with a scaling procedure that maintains all the assets of an analytical method. The method could derive robust and tight bounds for the variables of EVD algorithm. The bounds derived using the proposed approach remain same for any input matrix and are also independent of the number of iterations or size of the problem. Some benchmark hyperspectral data sets have been used to evaluate the efficiency of the proposed technique. It was found that by the proposed range estimation approach, all the variables generated during the computation of Jacobi EVD is bounded within ± 1.

Index Terms— Affine arithmetic, eigenvalue decomposition, fixed-point arithmetic, formal methods, integer bit-width allocation, interval arithmetic, overflow, range analysis, satisfiability-modulo-theory.

I. Introduction

Eigenvalue decomposition (EVD) is a key building block in signal processing and control applications. The fixed-point development of eigenvalue decomposition (EVD) algorithm have been extensively studied in the past few years [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] because fixed-point circuitry is significantly simpler and faster. Owing to its simplicity, fixed-point arithmetic is ubiquitous in low cost embedded platforms. Fixed-point arithmetic has played an important role in supporting the field-programmable-gate-array (FPGA) parallelism by keeping the hardware resources as low as possible. The most crucial step involved in the float-to-fixed conversion process is deciding the integer wordlengths (IWLs) in order to avoid overflow. This step has a significant impact on accuracy and hardware resources.

IWLs can be determined either using simulation [1], [11], [12] or by analytical (formal) methods [13], [14], [15], [16]. Existing works on fixed-point EVD have mainly used simulation-based approach for finding the IWLs [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [17] because of its capability to

be performed on any kind of systems. In simulation-based methods, variable bounds are estimated using the extreme values obtained from the simulation of the floating-point model. This method needs a large amount of input matrices to obtain a reliable estimation of ranges. Thus, the method is quite slow. Moreover, it cannot guarantee to avoid overflow for non-simulated matrices. This is primarily due to the diverse range of input data. A stochastic range estimation method is discussed in [18] which computes the ranges by propagating statistical distributions through operations. It requires large number of simulations to estimate system parameters and an appropriate input data set to estimate quality parameters [19], [14] and therefore, it does not produce absolute bounds [20].

There are several limitations associated with analytical (formal) methods. An analytical method based on L_1 norm and transfer function is described in [21]. This method produces theoretical bounds that guarantee no overflow will occur, but the approach is only limited to linear time-invariant systems [19]. Interval arithmetic (IA) ignores correlation among signals resulting in an overestimation of ranges [13]. Affine arithmetic (AA) is a preferable approach that takes into account the interdependency among the signals [22], but ranges determined through AA explode during division if the range of divisor includes zero [15], [16]. IA also suffers from the same problem. Both IA and AA are pessimistic approaches leading to higher implementation cost [23]. Satisfiability modulo theory (SMT) produces tight ranges compared to IA and AA [15], [24]. However, it is computationally expensive and much slower as compared to IA and AA [16], [25]. IA and AA methods compute the ranges of the intermediate variables by propagating the bounds of the input data through the arithmetic operations. SMT refines the range results provided by IA and AA. There are common issues associated with IA, AA and SMT. Given a particular range of the input matrix, these analytical methods compute certain ranges of the intermediate variables based on the arithmetic operations. However, if the range of the input matrix changes, the bounds for the variables no longer remain the same. Another issue with these analytical methods is that the bounds of the variables obtained using these methods are not independent of the number of iterations or the size of the problem. We exemplify these common issues associated with

IA, AA and SMT in the next section.

II. MOTIVATION

In this section, we illustrate the issues associated with the existing range estimation methods through a motivational example (dimensionality reduction of hyperspectral images using fixed-point EVD). Along with the covariance matrices of hyperspectral images, we have also used some random symmetric positive semi-definite matrices generated from MATLAB. We have chosen such an instance because it is discovered from the literature that some of the works on dimensionality reduction of hyperspectral images highlight the overflow issues while using fixed-point EVD algorithm. A sincere effort has been made to contemplate them.

The diverse range of the elements of the input data matrices for different hyperspectral images (HSIs) limits the use of fixed-point EVD for dimensionality reduction [26], [27]. If the range of the input data is diverse, selecting a particular IWL may not avoid overflow for all range of input cases. Egho *et al.* [28] stated that fixed-point implementation of EVD algorithm leads to inaccurate computation of eigenvalues and eigenvectors due to overflow. Therefore, the authors implemented Jacobi algorithm in FPGA using floating-point arithmetic. Lopez *et al.* [26] reported overflow issues while computing EVD in fixed-point arithmetic. Burger *et al.* [29] mentioned that while processing millions of HSI, numerical instability like overflow should be avoided. Hence, determination of proper IWLs for variables of fixed-point EVD algorithm in order to free it from overflow for all range of input data remains a major research issue.

The most widely used algorithm for dimensionality reduction is principal component analysis (PCA). PCA requires computation of eigenvalues (λ) and eigenvectors (x) given by

$$A = X\Lambda X^{\mathrm{T}}, \tag{1}$$

where A is the covariance matrix, Λ is a diagonal matrix containing the eigenvalues and the columns of X contain the eigenvectors. X is a new coordinate basis for the image. There are several algorithms developed in the literature for EVD of symmetric matrices [30], [31], [32]. Among all, two-sided Jacobi algorithm is most accurate and numerically stable [30], [33]. Most of the work attempted for dimensionality reduction via EVD uses two-sided Jacobi algorithm [28], [34], [35]. The same algorithm is used in this paper for computing EVD of the covariance matrix of the hyperspectral data. Apart from the accuracy and stability of Jacobi algorithm, it also has high degree potential for parallelism, and hence can be implemented on FPGA [5], [6]. In [3], [5], [4], [6], [7], [8], [9], [10] this algorithm is implemented on FPGA with fixed-point arithmetic to reduce power consumption and silicon area. However, in all the works, fixed-point implementation of Jacobi algorithm uses the simulation-based approach for estimating the ranges of variables. It does not produce promising bounds (as discussed earlier in section I).

Jacobi method computes EVD of a symmetric matrix A

Algorithm 1 Two-sided Jacobi EVD algorithm

1: $X = I$;
2: **for** $l = 1$ to n **do**
3:    **for** $i = 1$ to n **do**
4:       **for** $j = i + 1$ to n **do**
5:          $a = A(i,i)$;
6:          $b = A(j,j)$;
7:          $c = A(i,j) = A(j,i)$;
    /* compute the Jacobi rotation which diagonalizes $\begin{pmatrix} A(i,i) & A(i,j) \\ A(j,i) & A(j,j) \end{pmatrix} = \begin{pmatrix} a & c \\ c & b \end{pmatrix}$ */
8:          $t = \dfrac{sign(\frac{b-a}{c}) \cdot |c|}{|\frac{b-a}{2}| + \sqrt{c^2 + (\frac{b-a}{2})^2}}$;
9:          $cs = 1/\sqrt{1 + t^2}$;
10:        $sn = cs \cdot t$;
    /* update the 2×2 submatrix */
11:        $A(i,i) = a - c \cdot t$;
12:        $A(j,j) = b + c \cdot t$;
13:        $A(i,j) = A(j,i) = 0$;
    /* update the rest of rows and columns i and j */
14:        **for** $k = 1$ to n **do** except i and j
15:          $tmp = A(i,k)$;
16:          $A(i,k) = cs \cdot tmp - sn \cdot A(j,k)$;
17:          $A(j,k) = sn \cdot tmp + cs \cdot A(j,k)$;
18:          $A(k,i) = A(i,k)$;
19:          $A(k,j) = A(j,k)$;
20:        **end for**
    /* update the eigenvector matrix X */
21:        **for** $k = 1$ to n **do**
22:          $tmp = X(k,i)$;
23:          $X(k,i) = cs \cdot tmp - sn \cdot X(k,j)$;
24:          $X(k,j) = sn \cdot tmp + cs \cdot X(k,j)$;
25:        **end for**
26:       **end for**
27:    **end for**
28: **end for**
    /* eigenvalues are diagonals of the final A */
29: **for** $i = 1$ to n **do**
30:    $\lambda_i = A(i,i)$;
31: **end for**

---

by producing a sequence of orthogonally similar matrices, which eventually converges to a diagonal matrix [30] given by

$$\Lambda = J^{\mathrm{T}} AJ, \tag{2}$$

where J is the Jacobi rotation and Λ is a diagonal matrix containing eigenvalues (λ). In each step, we compute a Jacobi rotation with J and update A to $J^T AJ$, where J is chosen in such a way that two off-diagonal entries of a 2×2 matrix of A are set to zero. This is called two-sided or classical Jacobi method. Algorithm 1 lists the steps for Jacobi method. In order to investigate the challenges with fixed-point EVD algorithm, we have used four different types of HSI collected by the space-borne (Hyperion), air-borne (ROSIS and AVIRIS), handheld sensors (Landscape) and

Synthetic (simulated EnMap). The selected Hyperion image subset contains the Chilika Lake (latitude: 19.63 N - 19.68 N, longitude: 85.13 E - 85.18 E) and its catchment areas [36], [37]. ROSIS data was acquired during a flight campaign at Pavia University, northern Italy [38]. AVIRIS data was gathered by the AVIRIS sensor over the Indian Pines test site in North-western Indiana. Landscape data is obtained from the database available from Stanford University [39]. The simulated EnMap image subset contains the Maktesh Ramon, Israel (30.57 N, 34.83 E) [40]. The sizes of the covariance matrix for the images are 120×120 for Hyperion, 103×103 for ROSIS, 200×200 for AVIRIS, 148×148 for Landscape and 244×244 for simulated EnMap. Out of 120, 103, 200, 148 and 244 bands, only a certain number of bands are sufficient for obtaining suitable information due to the large correlation between adjacent bands. Hence, the dimension of the image should be reduced to decrease the redundancy in the data. The principal components (PCs) are decided from the magnitudes of the eigenvalues. The numbers of PCs which explain 99.0% variance are retained for the reconstruction purpose. The following paragraph describes the shortcomings of the existing range estimation methods while computing bounds for EVD algorithm.

Tables I and II shows the ranges obtained for Hyperion and ROSIS using simulation, IA, AA and the proposed method. The simulation-based range analysis is performed by feeding the floating-point algorithm with each input matrix separately and observing the data range. Notice that the ranges or the required IWLs (Table III) estimated using the simulation-based approach for Hyperion cannot avoid overflow in case of ROSIS. In other words, based on the ranges obtained using the simulation of Hyperion data one would allocate 24 bits to the integer part, but these number of bits cannot avoid overflow in case of ROSIS. Simulation-based method can only produce exact bounds for the simulated cases. Thus, simulation-based method is characterized by a need for stimuli, due to which it cannot be relied upon in practical implementations. In contrast, the static or analytical or formal methods like IA and AA which depends on the arithmetic operations always provide worst-case bounds so that no overflow occurs. However, the bounds are highly overestimated compared to the actual bounds produced by simulation-based method as shown in Tables I and II. This increases the hardware resources unnecessarily.

In order to examine the range explosion problem of IA and AA, we computed the range of A using IA and AA for random symmetric positive semi-definite matrices of different sizes generated from MATLAB. Table IV shows how the range of A explodes when computed through IA and AA. All the range estimation using IA and AA have been carried out using double precision floating-point format. According to the IEEE 754 double precision floating-point format, the maximum number that can be represented is in the order of 10^{308}. It is noticed in Table IV that whenever the range is more than the maximum representable number, it is termed as infinity. It is apparent from the algorithm that variable A is some or the other way related to the

computation of all other variables. So, with the range of A becoming infinity, the range of other variables also result in infinity as shown in Tables I and II. The range of variable A goes unbounded because of the pessimistic nature of bounds produced by IA and AA. All the issues with existing range estimation methods are handled meticulously by the proposed method that produces unvarying or robust bounds while at the same time tightens the ranges. This is quite apparent from Tables I and II. In order to combat this,

TABLE I

COMPARISON BETWEEN THE RANGES COMPUTED BY SIMULATION, INTERVAL ARITHMETIC AND AFFINE ARITHMETIC WITH RESPECT TO THE PROPOSED APPROACH FOR HYPERION DATA WITH THE RANGE OF THE COVARIANCE MATRIX AS $[-2.42e-05, 4.46e+05]$.

Var	Simulation	IA	AA	Proposed
A	$[-1.02e+06, 9.58e+06]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
t	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
cs	$[0.71, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$
sn	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
a	$[0, 9.58e+06]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$
b	$[0, 2.23e+06]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$
c	$[-1.02e+06, 1.16e+06]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
X	$[-0.874, 1]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
λ	$[6.47e-10, 9.58e+06]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$

TABLE II

COMPARISON BETWEEN THE RANGES COMPUTED BY SIMULATION, INTERVAL ARITHMETIC AND AFFINE ARITHMETIC WITH RESPECT TO THE PROPOSED APPROACH FOR ROSIS DATA WITH THE RANGE OF THE COVARIANCE MATRIX AS $[-2.67e-05, 5.81e+05]$.

Var	Simulation	IA	AA	Proposed
A	$[-3.27e+06, 2.04e+07]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
t	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
cs	$[0.71, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$
sn	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
a	$[0, 2.04e+07]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$
b	$[0, 2.51e+06]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$
c	$[-3.27e+06, 2.13e+06]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
X	$[-0.768, 1]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
λ	$[2.23e-10, 2.04e+07]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$

TABLE III

COMPARISON BETWEEN THE INTEGER WORDLENGTHS REQUIRED BASED ON THE RANGES ESTIMATED BY SIMULATION-BASED APPROACH SHOWN IN TABLES I AND II

Var	Hyperion	ROSIS
A	24	25
a	22	22
b	24	25
λ	24	25

SMT has arisen which produce tight bounds compared to IA and AA. However, SMT is again computationally costly. Its runtime grows abruptly with application complexity. Hence, applying SMT for large size matrices would be too complex. Amidst the individual issues of the analytical methods, there are also some common issues. Provided with a particular range of the input matrix, the analytical methods (IA, AA and

TABLE IV

Size	Start	l=1, i=1, j=2	l=3, i=3, j=4	l=4, i=4, j=6	l=6, i=6, j=8	End
n=2	$[0.65, 0.95]$	$[-0.59, 2.35]$				$[-4.06, 4.18]$
n=4	$[0.03, 0.93]$	$[-5.52, 14.94]$	$[-2.19\mathrm{e}{+}21, 2.20\mathrm{e}{+}21]$			$[-3.7\mathrm{e}{+}28, 3.7\mathrm{e}{+}28]$
n=6	$[0.18, 0.79]$	$[-6.90, 28.62]$	$[-6.96\mathrm{e}{+}61, 7.08\mathrm{e}{+}61]$	$[-2.5\mathrm{e}{+}91, 2.6\mathrm{e}{+}91]$		$[-4.5\mathrm{e}{+}139, 4.5\mathrm{e}{+}139]$
n=8	$[0.11, 0.75]$	$[-9.77, 48.29]$	$[-1.01\mathrm{e}{+}126, 1.03\mathrm{e}{+}126]$	$[-2.6\mathrm{e}{+}187, 2.6\mathrm{e}{+}187]$	$[-1.6\mathrm{e}{+}301, 1.6\mathrm{e}{+}301]$	$[-\infty, \infty]$
n=10	$[0.09, 0.96]$	$[-16.62, 96.48]$	$[-4.77\mathrm{e}{+}215, 4.88\mathrm{e}{+}215]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-\infty, \infty]$
n=12	$[0.03, 0.97]$	$[-21.06, 139.76]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-\infty, \infty]$

SMT) compute certain ranges of the intermediate variables based on the arithmetic operations. Notwithstanding, the ranges no longer remain the same, if the range of the input matrix changes. In order to investigate this issue, we consider two 2×2 symmetric input matrices given by

$$C = \begin{pmatrix} 0.4427 & 0.1067 \\ 0.1067 & 0.4427 \end{pmatrix} \tag{3}$$

and

$$D = \begin{pmatrix} 33.4834 & 22.2054 \\ 22.2054 & 33.4834 \end{pmatrix}. \tag{4}$$

The ranges obtained using IA and SMT in case of matrix C cannot guarantee to avoid overflow in case of D as shown in Tables V and VI. The fact is also similar for ranges derived using AA. This scenario is handled correctly by the proposed method that produces robust and tight bounds in both the cases C and D. The range estimation using SMT was carried out using the freely available *HySAT* implementation [41].

There is one more common issue with these anaytical methods. We know that, provided with a fixed range of the input stimuli, these analytical (formal) methods successfully produce robust bounds [24]. Even though the range of the input matrix is fixed, the bounds produced by these analytical methods would be robust only for a particular size of the problem or number of iterations. In other words, the bounds obtained will not be independent of the number of iterations. In order to illustrate this, let us consider two random

TABLE V

Var	Simulation	IA	SMT	Proposed
C	$[0, 0.549]$	$[-3.88, 3.92]$	$[-2.0, 3.2]$	$[-1, 1]$
t	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
cs	$[0, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$
sn	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
a	$[0, 0.336]$	$[0, 0.336]$	$[0, 0.336]$	$[0, 1]$
b	$[0, 0.443]$	$[0, 0.443]$	$[0, 0.443]$	$[0, 1]$
c	0	0	0	$[-1, 1]$
X	$[-0.707, 0.707]$	$[-2.88, 2.88]$	$[-1.76, 1.76]$	$[-1, 1]$
λ	$[0.336, 0.549]$	$[-2.29, 3.92]$	$[-1.06, 3.22]$	$[0, 1]$

symmetric positive definite matrices of sizes 3×3 and 5×5 given by

$$Y = \begin{pmatrix} 46.7785 & 28.3501 & 18.8598 \\ 28.3501 & 20.1805 & 13.0975 \\ 18.8598 & 13.0975 & 8.6377 \end{pmatrix} \tag{5}$$

TABLE VI

Var	Simulation	IA	SMT	Proposed
D	$[0, 55.68]$	$[-147.57, 151.96]$	$[-87.2, 103.3]$	$[-1, 1]$
t	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
cs	$[0, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$
sn	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
a	$[0, 11.278]$	$[0, 11.278]$	$[0, 11.278]$	$[0, 1]$
b	$[0, 33.483]$	$[0, 33.483]$	$[0, 33.483]$	$[0, 1]$
c	0	0	0	$[-1, 1]$
X	$[-0.707, 0.707]$	$[-2.36, 2.36]$	$[-1.54, 1.54]$	$[-1, 1]$
λ	$[11.278, 55.68]$	$[-68.5, 151.96]$	$[-12.6, 103.3]$	$[0, 1]$

and

$$Z =$$
$$\begin{pmatrix} 107.6724 & 97.1687 & 107.1030 & 101.8092 & 78.4556 \\ 97.1687 & 118.4738 & 109.0664 & 114.7589 & 101.8092 \\ 107.1030 & 109.0664 & 126.1528 & 109.0664 & 107.1030 \\ 101.8092 & 114.7589 & 109.0664 & 118.4738 & 97.1687 \\ 78.4556 & 101.8092 & 107.1030 & 97.1687 & 107.6724 \end{pmatrix}. \tag{6}$$

The bounds obtained using IA for the input matrices Y and Z are shown in Table VII. The bounds are unnecessarily large compared to the actual bounds produced by the simulation-based approach shown in Table VIII. Now, the input matrices are scaled through the upper bound of their spectral norm to limit their range within -1 and 1. The new matrices $\hat{Y}$ and $\hat{Z}$ whose elements range between -1 and 1 are given by

$$\hat{Y} = \begin{pmatrix} 0.2848 & 0.3945 & 0.3805 \\ 0.3945 & 0.2848 & 0.3945 \\ 0.3805 & 0.0163 & 0.2848 \end{pmatrix} \tag{7}$$

and

$$\hat{Z} = \begin{pmatrix} 0.1160 & 0.2306 & 0.0349 & 0.3036 & 0.0860 \\ 0.2306 & 0.1160 & 0.2306 & 0.0349 & 0.3036 \\ 0.0349 & 0.2306 & 0.1160 & 0.2306 & 0.3435 \\ 0.3036 & 0.0349 & 0.2306 & 0.1160 & 0.2306 \\ 0.0860 & 0.3036 & 0.0349 & 0.2306 & 0.1160 \end{pmatrix}. \tag{8}$$

The ranges obtained for the scaled input matrices are shown in Table IX. Even though after scaling, the range of the variables obtained using IA are large and unbounded compared to the original bounds obtained using simulation-based method (Table X). The difference in unboundedness of the ranges shown in Tables VII and IX is not substantially large. This illustrates that the ranges obtained using IA are not

Variables	IA (Y)	IA (Z)
Y or Z	$[-8.88\text{e}+9, 8.94\text{e}+9]$	$[-4.51\text{e}+71, 4.51\text{e}+71]$
t	$[-1, 1]$	$[-1, 1]$
cs	$[0, 1]$	$[0, 1]$
sn	$[-1, 1]$	$[-1, 1]$
a	$[-9.81\text{e}+8, 9.93\text{e}+8]$	$[-1.80\text{e}+70, 1.81\text{e}+70]$
b	$[-9.81\text{e}+8, 9.93\text{e}+8]$	$[-1.80\text{e}+70, 1.81\text{e}+70]$
c	$[-9.81\text{e}+8, 9.93\text{e}+8]$	$[-1.80\text{e}+70, 1.81\text{e}+70]$
X	$[-9587, 10607]$	$[-4.26\text{e}+34, 4.38\text{e}+34]$
λ	$[-8.88\text{e}+9, 8.94\text{e}+9]$	$[-4.51\text{e}+71, 4.51\text{e}+71]$

independent of the number of iterations. Similar is the case for both AA and SMT. We can observe the phenomenon in Table XI for one of the test hyperspectral data (simulated EnMAP). Inspite of the range of covariance matrix being $[-3.71\text{e}-06, 0.032]$, the bounds estimated using IA and AA exploded compared to the actual bounds obtained using simulation-based approach. These examples comprehend that the bounds derived using the existing analytical methods are not independent of the number of iterations. Given the issues of the existing range estimation methods, our proposed method provides robust and tight bounds for the variables as shown in Tables I, II, V, VI and XI. Moreover, the bounds produced by the proposed method are independent of the size of the problem. The key to all these advantages is the usage of the scaling method and vector, matrix norm properties to derive the ranges.

Variables	Simulation (Y)	Simulation (Z)
Y or Z	$[-0.123, 72.98]$	$[-15.73, 526.54]$
t	$[-1, 1]$	$[-1, 1]$
cs	$[0, 1]$	$[0, 1]$
sn	$[-1, 1]$	$[-1, 1]$
a	$[0, 72.97]$	$[0, 526.54]$
b	$[0, 20.18]$	$[0, 526.54]$
c	$[-0.123, 28.35]$	$[-8.45, 191.52]$
X	$[-0.61, 1]$	$[-0.71, 1]$
λ	$[0.08, 72.98]$	$[6.9\text{e}-3, 526.54]$

Variables	IA ($\hat{Y}$)	IA ($\hat{Z}$)
$\hat{Y}$ or $\hat{Z}$	$[-9.44\text{e}+7, 9.51\text{e}+7]$	$[-8.08\text{e}+68, 8.08\text{e}+68]$
t	$[-1, 1]$	$[-1, 1]$
cs	$[0, 1]$	$[0, 1]$
sn	$[-1, 1]$	$[-1, 1]$
a	$[-1.04\text{e}+7, 1.06\text{e}+7]$	$[-3.23\text{e}+67, 3.23\text{e}+67]$
b	$[-1.04\text{e}+7, 1.06\text{e}+7]$	$[-3.23\text{e}+67, 3.23\text{e}+67]$
c	$[-1.04\text{e}+7, 1.06\text{e}+7]$	$[-3.23\text{e}+67, 3.23\text{e}+67]$
X	$[-9587, 10607]$	$[-4.26\text{e}+34, 4.37\text{e}+34]$
λ	$[-9.44\text{e}+7, 9.51\text{e}+7]$	$[-8.08\text{e}+68, 8.08\text{e}+68]$

Variables	Simulation ($\hat{Y}$)	Simulation ($\hat{Z}$)
$\hat{Y}$ or $\hat{Z}$	$[-1.31\text{e}-3, 0.78]$	$[-0.028, 0.94]$
t	$[-1, 1]$	$[-1, 1]$
cs	$[0, 1]$	$[0, 1]$
sn	$[-1, 1]$	$[-1, 1]$
a	$[0, 0.78]$	$[0, 0.94]$
b	$[0, 0.21]$	$[0, 0.94]$
c	$[-1.31\text{e}-3, 0.31]$	$[-0.015, 0.34]$
X	$[-0.61, 1]$	$[-0.71, 1]$
λ	$[8.59\text{e}-04, 0.78]$	$[1.24\text{e}-5, 0.94]$

Var	Simulation	IA	AA	Proposed
A	$[-0.072, 1.29]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
t	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
cs	$[0.71, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$
sn	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
a	$[0, 1.29]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$
b	$[0, 1.29]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$
c	$[-0.067, 0.174]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
X	$[-0.823, 1]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
λ	$[1.24\text{e}-05, 0.942]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$

III. PROPOSED SOLUTION

Particularizing, there are mainly three issues associated with the existing range estimation methods:

1) incompetence of the simulation-based approach to produce unvarying or robust bounds,
2) bounds produced by existing analytical (formal) methods are not independent of the number of iterations or size of the problem, and
3) overestimated bounds produced by IA and AA.

Taking into account the issues 1 and 2, we propose in this study, an analytical method based on vector and matrix norm properties to derive unvarying or robust bounds for the variables of EVD algorithm. The proof for deriving the bounds make use of the fact that all the eigenvalues of a symmetric semi-positive definite matrix are bounded within the upper bound for the spectral norm of the matrix. Further taking into consideration the issue 3, we demonstrate that if the spectral norm of any matrix is kept within unity, tight ranges for the variables of the EVD algorithm can be derived. It is well-known that the spectral norm of any matrix is bounded by [31], [42]

$$\|A\|_2 \le \sqrt{\|A\|_1 \|A\|_\infty}. \quad (9)$$

For symmetric matrices, the spectral norm $\|A\|_2$ in (9) can be replaced with the spectral radius $\rho(A)$.

Theorem 1: Given the bounds for spectral norm as $\|A\|_2 \le \sqrt{\|A\|_1 \|A\|_\infty}$, the Jacobi EVD algorithm applied

to A has the following bounds for the variables for all i, j, k and l:

- $[A]_{kl} \in [-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$
- $t \in [-1, 1]$
- $cs \in [0, 1]$
- $sn \in [-1, 1]$
- $[X]_{kl} \in [-1, 1]$
- $a \in [0, \sqrt{\|A\|_1\|A\|_\infty}]$
- $b \in [0, \sqrt{\|A\|_1\|A\|_\infty}]$
- $c \in [-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$
- $[\lambda_i]_k \in [0, \sqrt{\|A\|_1\|A\|_\infty}]$

where i, j denote the iteration number and $[]_k$ and $[]_{kl}$ denote the kth component of a vector and kl^{th} component of a matrix respectively.

Proof: Using vector and matrix norm properties the ranges of the variables can be derived. We start by bounding the elements of the input symmetric matrix as

$$\max_{kl} |[A]_{kl}| \leq \|A\|_2 = \rho(A) \leq \sqrt{\|A\|_1\|A\|_\infty}, \quad (10)$$

where (10) follows from [31]. Hence, the elements of A are in the range $[-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$. Line 30 in Algorithm 1 shows the computation of eigenvalues. We know that $\rho(A) \leq \sqrt{\|A\|_1\|A\|_\infty}$, so the upper bound for the eigenvalues is equal to $\sqrt{\|A\|_1\|A\|_\infty}$. In this work, the fixed-point Jacobi EVD algorithm is applied to covariance matrices. Due to the positive semi-definiteness property of covariance matrices, the lower bound for the eigenvalues is equal to zero. Thus, the range of λ_i is $[0, \sqrt{\|A\|_1\|A\|_\infty}]$. The eigenvalues in Line 30 can also be calculated as

$$\lambda_i = \|A(:, i)\|_2. \quad (11)$$

According to vector norm property we can say that

$$\|A(:, i)\|_\infty \leq \|A(:, i)\|_2, \quad (12)$$

where $\|A(:, i)\|_\infty$ is the maximum of the absolute of the elements in $A(:, i)$. From the upper bound of λ_i, (11) and (12) we can say that each element of $A(:, i)$ lie in the range $[-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$. Thus all elements of A lie in the range $[-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$ for all the iterations. Since we have considered symmetric positive semi-definite matrices (unlike the off-diagonal entries the diagonal elements are always positive), the diagonal elements of A are in the range $[0, \sqrt{\|A\|_1\|A\|_\infty}]$. Rest of the elements lie in the range $[-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$. Line 5, 6 and 7 in Algorithm 1 computes a, b and c respectively. Since a and b are the diagonal elements of A, their range is $[0, \sqrt{\|A\|_1\|A\|_\infty}]$. c is the off-diagonal entry of A, therefore its range is $[-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$. Line 8 in Algorithm 1 computes t. Let $t=r/s$ such that

$$r = sign\left(\frac{b-a}{c}\right) \cdot |c| \text{ and } s = \left|\frac{b-a}{2}\right| + \sqrt{c^2 + \left(\frac{b-a}{2}\right)^2}. \quad (13)$$

According to (13), numerator (r) of t lies in the range $[-|c|, |c|]$. $\left|\frac{b-a}{2}\right|$ and $\sqrt{c^2 + \left(\frac{b-a}{2}\right)^2}$ are always positive. The summation s is greater than or equal to $|c|$, because if $b = a$ then s is equal to c or if $b \neq a$ then s is greater than c since $\left|\frac{b-a}{2}\right|$ is greater than or equal to zero and $\sqrt{c^2 + \left(\frac{b-a}{2}\right)^2}$ is greater than $|c|$. From the range of a, b and the denominator of t, we can say that $|c|$ will always be less or equal to $\left|\frac{b-a}{2}\right| + \sqrt{c^2 + \left(\frac{b-a}{2}\right)^2}$. Thus, we can conclude that t lies in the range $[-1, 1]$ and arc tangent of t is limited in the range $[-\frac{\pi}{4}, \frac{\pi}{4}]$. The Jacobi EVD method tries to make the off-diagonal entries of 2×2 submatrix of A zero by overwriting A with $J^T A J$. According to 2×2 symmetric Schur decomposition discussed in [31], cs and sn are cosine and sine trigonometric functions. Thus, the bounds of cs and sn are $[-1, 1]$. Line 9 in Algorithm 1 computes cs which involves square root operation and therefore the range of cs can be modified to $[0, 1]$. As the range of cs and t are $[0, 1]$ and $[-1, 1]$ respectively, using multiplication rule of interval arithmetic [43] the range of sn (Line 10) can be derived as $[-1, 1]$. Next we bound the elements of X. X is the eigenvector matrix each column of which has unity norm (eigenvectors of symmetric matrices are orthogonal). Hence all elements of X are in the range $[-1, 1]$ following (14).

$$\|X(:, i)\|_\infty \leq \|X(:, i)\|_2 = 1. \quad (14)$$

Since the range of A is $[-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$, according to Line 15 of Algorithm 1, the range of tmp can be fixed as $[-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$. $\blacksquare$

The bounds obtained according to Theorem 1 remain unchanged for all the iterations of the algorithm. The bounds are independent of the number of iterations or the size of the input matrix. Thus, the issue 2 has been handled accurately. Now considering the issue 1, the bounds according to Theorem 1 remain same (the pattern remains the same as shown in Theorem 1) for any input matrix, but depend on the factor $\sqrt{\|A\|_1\|A\|_\infty}$. For different input matrices, the magnitude of $\sqrt{\|A\|_1\|A\|_\infty}$ will change and this, in turn, will differ the bounds. The issue 1 has not yet been handled prudently. Hence, we propose that if the input matrix is scaled through $m = \sqrt{\|A\|_1\|A\|_\infty}$ then we can achieve a two-fold advantage: unvarying and tight bounds (solution for issue 3). This will resolve all the issues. If the input matrix is scaled as $\hat{A} = \frac{A}{m}$, the EVD of matrix $\hat{A}$ is given as

$$\hat{A}x = \hat{\lambda}x, \quad (15)$$

where $Ax = \lambda x$ and $\hat{\lambda} = \frac{\lambda}{m}$. x is the eigenvector and λ is the eigenvalue. After scaling through a scalar value, the original eigenvectors do not change. The original eigenvalues change by a factor $\frac{1}{m}$. We need not recover the original eigenvalues because, in PCA, eigenvaues are only used to calculate the required number of PCs. Since, all the eigenvalues are scaled by the same factor, the number of PCs do not change whether the number is fixed using original eigenvalues or scaled ones. In applications, where original eigenvalues are required, the number of IWLs required is $\lceil \log_2(\sqrt{\|A\|_1\|A\|_\infty}) \rceil$ depending on the magnitude of the scaling factor. Only the binary point of the eigenvalues is

required to be adjusted online while for other variables it is fixed irrespective of the property of the input matrix.

Theorem 2: Given the scaling factor as $m = \sqrt{\|A\|_1 \|A\|_\infty}$, the Jacobi EVD algorithm (Algorithm 1) applied to $\hat{A}$ has the following bounds for the variables for all i, j, k and l:

- $[\hat{A}]_{kl} \in [-1, 1]$
- $t \in [-1, 1]$
- $cs \in [0, 1]$
- $sn \in [-1, 1]$
- $[X]_{kl} \in [-1, 1]$
- $a \in [0, 1]$
- $b \in [0, 1]$
- $c \in [-1, 1]$
- $[\hat{\lambda}_i]_k \in [0, 1]$

where i, j denote the iteration number and $[]_k$ and $[]_{kl}$ denote the kth component of a vector and kl^{th} component of a matrix respectively.

Proof: Using vector and matrix norm properties the ranges of the variables can be derived. We start by bounding the elements of the input symmetric matrix as

$$\max_{kl} |[\hat{A}]_{kl}| \leq \|\hat{A}\|_2 = \rho(\hat{A}) \leq 1, \qquad (16)$$

where (16) follows from [31]. Hence, the elements of $\hat{A}$ are in the range $[-1, 1]$. The remaining bounds are derived in the similar fashion as decribed in proof for Therorem 1. ∎
The bounds on the variables of EVD algorithm obtained after scaling remain constant for all the iterations and also do not vary for any input matrix. Besides, the bounds are also tight.

IV. More data sets with results

In this section, we present a few more hyperspectral data sets, and we compare the bounds on variables of Jacobi EVD algorithm produced by the existing range estimation methods and the proposed approach. Tables XII and XIII show the comparison between the bounds on the variables obtained by existing range estimation methods with respect to the proposed approach through the AVIRIS and Landscape data sets. We can observe that the ranges estimated using the simulation for AVIRIS data cannot avoid overflow in case of Landscape. This is quite apparent from the number of integer bits required, shown in Table XIV. As usual, the bounds produced by IA and AA outbursted. However, the proposed method produces robust and tight bounds. The bounds obtained are independent of any range of the input matrix and also the number of iterations.

Signal-to-quantization-noise-ratio (SQNR) is chosen as an error measure to evaluate the accuracy of the proposed method [25], [44]. It is given by

$$SQNR = 10 \log_{10}(\mathrm{E}(|\lambda_{float}|^2))/(\mathrm{E}(|\lambda_{float} - \lambda_{fixed}|^2)), \qquad (17)$$

where λ_{float} and λ_{fixed} are the eigenvalues obtained from double precision floating-point and fixed-point implementations. SQNR of the eigenvalues obtained through the proposed fixed-point design is shown in Table XV. In Table XV,

TABLE XII

COMPARISON BETWEEN THE RANGES COMPUTED BY SIMULATION, INTERVAL ARITHMETIC AND AFFINE ARITHMETIC WITH RESPECT TO THE PROPOSED APPROACH FOR AVIRIS DATA WITH THE RANGE OF THE COVARIANCE MATRIX AS $[-5.01e+05, 1.07e+06]$.

Var	Simulation	IA	AA	Proposed
A	$[-2.66e+6, 2.68e+07]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
t	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
cs	$[0.71, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$
sn	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
a	$[0, 2.68e+07]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$
b	$[0, 9.21e+06]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$
c	$[-2.38e+06, 3.66e+06]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
X	$[-0.939, 1]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
λ	$[15.80, 2.67e+07]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$

TABLE XIII

COMPARISON BETWEEN THE RANGES COMPUTED BY SIMULATION, INTERVAL ARITHMETIC AND AFFINE ARITHMETIC WITH RESPECT TO THE PROPOSED APPROACH FOR LANDSCAPE DATA WITH THE RANGE OF THE COVARIANCE MATRIX AS $[-5.47e+32, 6.81e+32]$.

Var	Simulation	IA	AA	Proposed
A	$[-5.06e+33, 4.32e+34]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
t	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
cs	$[0.71, 1]$	$[0, 1]$	$[0, 1]$	$[0, 1]$
sn	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$	$[-1, 1]$
a	$[0, 4.32e+34]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$
b	$[0, 4.32e+34]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$
c	$[-5.06e+33, 2.12e+33]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
X	$[-0.932, 1]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[-1, 1]$
λ	$[1.0e+19, 4.32e+34]$	$[-\infty, \infty]$	$[-\infty, \infty]$	$[0, 1]$

TABLE XIV

COMPARISON BETWEEN THE INTEGER WORDLENGTHS REQUIRED BASED ON THE RANGES ESTIMATED BY SIMULATION-BASED APPROACH SHOWN IN TABLES XII AND XIII

Var	AVIRIS	Landscape
A	25	116
a	24	116
b	25	116
λ	25	116

TABLE XV

SIGNAL-TO-QUANTIZATION-NOISE-RATIO OF EIGENVALUES OBTAINED IN FIXED-POINT ARITHMETIC (WLs CHOSEN ARE AS A GENERAL BITWIDTH CONSIDERING THE WORST CASE) AFTER DETERMINING RANGES THROUGH PROPOSED APPROACH.

WLs	50 bits	40 bits	32 bits
Hyperion	176.76	106.44	78.03
ROSIS	180.13	134.79	74.96
Landscape	180.65	122.36	77.18
AVIRIS	178.54	110.76	76.43
EnMap	180.67	130.24	78.36

we observe high magnitudes of SQNR which exhibit that the set of ranges obtained according to Theorem 2 are sufficient for avoiding overflow for any input matrix. For data sets like Landscape, where the the range is exorbitant resulting in large IWLs (Table XIV), wordlengths like 50, 40 or 32 bits would never fit. In such cases, with the proposed approach it

TABLE XVI

MEAN-SQUARE-ERROR OF PCS OBTAINED IN FIXED-POINT ARITHMETIC (WLS CHOSEN ARE AS A GENERAL BITWIDTH CONSIDERING THE WORST CASE) AFTER DETERMINING RANGES THROUGH PROPOSED APPROACH.

WLs	Hyperion				ROSIS					Landscape	
	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4	PC5	PC1	PC2
32	8.1e-7	4.9e-7	3.7e-6	4.3e-6	0	1.8e-7	4.5e-6	6.5e-6	2.4e-6	1.2e-9	1.1e-8
40	0	0	8.5e-7	1.9e-7	0	0	0	0	0	1.8e-10	7.4e-11
50	0	0	0	0	0	0	0	0	0	0	0

was possible to fit all the variables within 32 bit wordlength and obtain a high value of SQNR. A common measure to compare two images is mean-square-error (MSE) [45]. MSE between PCA images of fixed-point implementations with various WLs after derving the ranges through proposed approach are shown in Table XVI. The required number of PCs for Hyperion, ROSIS and Landscape are 4, 5 and 2 respectively. The number of PCs explaining 99.0% variance in case of AVIRIS and EnMap are relatively higher. Therefore, the Table XVI only exhibits the results of Hyperion, ROSIS and Landscape. However, similar results were obtained for AVIRIS and EnMap. We observe that the MSE values are negligibly small which signify that the ranges obtained through the proposed approach are absolutely robust. Thus, the error metrics (SQNR and MSE) imply that the number of integer bits derived using the proposed approach is sufficient for avoiding overflow. After deriving the proper ranges through the proposed approach, the fixed-point design is synthesized on Xilinx Virtex 7 XC7VX485 FPGA for different WLs through Vivado high-level synthesis (HLS) design tool [46]. We have used SystemC (mimics hardware description language VHDL and Verilog) to develop the fixed-point code [47]. Using the HLS tool, the SystemC fixed-point code is transformed into a hardware IP (intellectual property) described in Verilog. We compare the resource utilization of simulation approach with respect to the proposed approach (for the same level of accuracy) through the test hyperspectral data sets. The comparative study is illustrated in Table XVII. There is a noteworthy difference in the hardware resources. The hardware resources in case of simulation approach are considerably large compared to the resources used in case of the proposed approach. For the sake of maintaining the same level of accuracy (SQNR, MSE) as the proposed method, the simulation approach uses 50 bit wordlength.

The proposed method also produces robust and tight analytical bounds for variables of singular value decomposition algorithm [48].

V. CONCLUSION

In this paper, we bring out the problem of integer bit-width allocation for the variables of eigenvalue decomposition algorithm. We highlight the issues of the existing range estimation methods in the context of EVD. Integer bit-width allocation is an essential step in fixed-point hardware design. In light of the significance of this step, this paper introduces an analytical method based on vector and matrix

TABLE XVII

COMPARISON BETWEEN HARDWARE COST (%) OF FIXED-POINT JACOBI ALGORITHM AFTER DETERMINING RANGES THROUGH PROPOSED AND SIMULATION APPROACHES.

	Proposed								
WL	Hyperion			ROSIS			AVIRIS		
	FF	LUTs	Power	FF	LUTs	Power	FF	LUTs	Power
32	1.62	6.29	0.413	1.62	6.32	0.42	1.63	6.51	0.45
	Simulation								
WL	Hyperion			ROSIS			AVIRIS		
	FF	LUTs	Power	FF	LUTs	Power	FF	LUTs	Power
50	8	23	2.59	8	23	2.64	8	23	2.64

norm properties together with a scaling procedure to produce robust and tight bounds. Through some hyperspectral data sets, we demonstrate the efficacy of the proposed method in dealing with the issues associated with existing methods. SQNR and MSE values show that the ranges derived using the proposed approach are sufficient for avoiding overflow in case of any input matrix. There are many other numerical linear algebra algorithms which can benefit from the proposed method like QR factorization, power method for finding largest eigenvalue, bisection method for finding eigenvalues of a symmetric tridiagonal matrix, QR iteration, Arnoldi method for transforming a non-symmetric matrix into an upper Hessenberg matrix and LU factorization and Cholesky factorization.

Dealing with the precision problem will be a scope for the future work.

REFERENCES

[1] Z. Nikolić, H. T. Nguyen, and G. Frantz, "Design and implementation of numerical linear algebra algorithms on fixed point dsps," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, 2007.

[2] Y.-L. Chen, C.-Z. Zhan, T.-J. Jheng, and A.-Y. A. Wu, "Reconfigurable adaptive singular value decomposition engine design for high-throughput mimo-ofdm systems," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 21, no. 4, pp. 747–760, 2013.

[3] D. Milford and M. Sandell, "Singular value decomposition using an array of cordic processors," *Signal Processing*, vol. 102, pp. 163–170, 2014.

[4] T. Pradhan, B. Kabi, A. Routray, and G. Anirudh, "Fixed-point hestenes svd algorithm for computing eigen faces."

[5] R. Mohanty, G. Anirudh, T. Pradhan, B. Kabi, and A. Routray, "Design and performance analysis of fixed-point jacobi svd algorithm on reconfigurable system," *IERI Procedia*, vol. 7, pp. 21–27, 2014.

[6] P. M. Szecówka and P. Malinowski, "Cordic and svd implementation in digital hardware," in *Mixed Design of Integrated Circuits and Systems (MIXDES), 2010 Proceedings of the 17th International Conference*. IEEE, 2010, pp. 237–242.

[7] R. C. Grammenos, S. Isam, and I. Darwazeh, "Fpga design of a truncated svd based receiver for the detection of sefdm signals," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on*. IEEE, 2011, pp. 2085–2090.

[8] Y. Wang, K. Cunningham, P. Nagvajara, and J. Johnson, "Singular value decomposition hardware for mimo: State of the art and custom design," in *Reconfigurable Computing and FPGAs (ReConFig), 2010 International Conference on*. IEEE, 2010, pp. 400–405.

[9] Z. Liu, K. Dickson, and J. V. McCanny, "Application-specific instruction set processor for soc implementation of modern signal processing algorithms," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 52, no. 4, pp. 755–765, 2005.

[10] C. Shi and R. W. Brodersen, "Automated fixed-point data-type optimization tool for signal processing and communication systems," in *Design Automation Conference, 2004. Proceedings. 41st*. IEEE, 2004, pp. 478–483.

[11] S. Kim, K.-I. Kum, and W. Sung, "Fixed-point optimization utility for c and c++ based digital signal processing programs," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 45, no. 11, pp. 1455–1464, 1998.

[12] K.-I. Kum, J. Kang, and W. Sung, "Autoscaler for c: An optimizing floating-point to integer c program converter for fixed-point digital signal processors," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 47, no. 9, pp. 840–848, 2000.

[13] D.-U. Lee, A. A. Gaffar, R. C. Cheung, O. Mencer, W. Luk, G. Constantinides *et al.*, "Accuracy-guaranteed bit-width optimization," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 10, pp. 1990–2000, 2006.

[14] J. López, C. Carreras, O. Nieto-Taladriz *et al.*, "Improved interval-based characterization of fixed-point lti systems with feedback loops," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 11, pp. 1923–1933, 2007.

[15] A. B. Kinsman and N. Nicolici, "Bit-width allocation for hardware accelerators for scientific computing using sat-modulo theory," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, no. 3, pp. 405–413, 2010.

[16] S. Vakili, J. P. Langlois, and G. Bois, "Enhanced precision analysis for accuracy-aware bit-width optimization using affine arithmetic," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 12, pp. 1853–1865, 2013.

[17] T. Pradhan, B. Kabi, R. Mohanty, and A. Routray, "Development of numerical linear algebra algorithms in dynamic fixed-point format: a case study of lanczos tridiagonalization," *International Journal of Circuit Theory and Applications*, vol. 44, no. 6, pp. 1222–1262, 2016.

[18] B. Wu, J. Zhu, and F. N. Najm, "Dynamic-range estimation," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 9, pp. 1618–1636, 2006.

[19] G. Constantinides, A. B. Kinsman, and N. Nicolici, "Numerical data representations for fpga-based scientific computing," *IEEE Design & Test of Computers*, no. 4, pp. 8–17, 2011.

[20] A. Banciu, "A stochastic approach for the range evaluation," Ph.D. dissertation, Université Rennes 1, 2012.

[21] J. Carletta, R. Veillette, F. Krach, and Z. Fang, "Determining appropriate precisions for signals in fixed-point iir filters," in *Proceedings of the 40th annual Design Automation Conference*. ACM, 2003, pp. 656–661.

[22] E. Goubault and S. Putot, "A zonotopic framework for functional abstractions," *Formal Methods in System Design*, vol. 47, no. 3, pp. 302–360, 2015.

[23] R. Nehmeh, D. Menard, A. Banciu, T. Michel, and R. Rocher, "Integer word-length optimization for fixed-point systems," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE nternational Conference on*. IEEE, 2014, pp. 8321–8325.

[24] A. B. Kinsman and N. Nicolici, "Automated range and precision bit-width allocation for iterative computations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 30, pp. 1265–1278, 2011.

[25] O. Sarbishei and K. Radecka, "On the fixed-point accuracy analysis and optimization of polynomial specifications," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 32, no. 6, pp. 831–844, 2013.

[26] S. Lopez, T. Vladimirova, C. Gonzalez, J. Resano, D. Mozos, and A. Plaza, "The promise of reconfigurable computing for hyperspectral imaging onboard systems: A review and trends," *Proceedings of the IEEE*, vol. 101, no. 3, pp. 698–722, 2013.

[27] C. Egho, T. Vladimirova, and M. N. Sweeting, "Acceleration of karhunen-loève transform for system-on-chip platforms," in *Adaptive Hardware and Systems (AHS), 2012 NASA/ESA Conference on*. IEEE, 2012, pp. 272–279.

[28] C. Egho and T. Vladimirova, "Adaptive hyperspectral image compression using the klt and integer klt algorithms," in *Adaptive Hardware and Systems (AHS), 2014 NASA/ESA Conference on*. IEEE, 2014, pp. 112–119.

[29] J. Burger and A. Gowen, "Data handling in hyperspectral image analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 108, no. 1, pp. 13–22, 2011.

[30] B. N. Datta, *Numerical linear algebra and applications*. Siam, 2010.

[31] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[32] T. Pradhan, A. Routray, and B. Kabi, "Comparative evaluation of symmetric svd algorithms for real-time face and eye tracking," in *Matrix information geometry*. Springer, 2013, pp. 323–340.

[33] J. Demmel and K. Veselic, "Jacobi's method is more accurate than qr," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 4, pp. 1204–1245, 1992.

[34] C. Egho and T. Vladimirova, "Hardware acceleration of the integer karhunen-loève transform algorithm for satellite image compression," in *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*. IEEE, 2012, pp. 4062–4065.

[35] C. Gonzalez, S. Lopez, D. Mozos, and R. Sarmiento, "A novel fpga-based architecture for the estimation of the virtual dimensionality in remotely sensed hyperspectral images," *Journal of Real-Time Image Processing*, pp. 1–12.

[36] https://hyspiri.jpl.nasa.gov/downloads/2015_Symposium/day2/8_DecompositionAlgorith

[37] B. Kabi, A. S. Sahadevan, R. Mohanty, A. Routray, B. S. Das, and A. Mohanty, "An overflow-free fixed-point singular value decomposition algorithm for dimensionality reduction of hyperspectral images."

[38] http://www.ehu.eus/ccwintco/index.php?title=HyperspectralRemoteSensingScenes.

[39] T. Skauli and J. Farrell, "A collection of hyperspectral images for imaging systems research," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2013, pp. 86 600C–86 600C.

[40] K. Segl, L. Guanter, C. Rogass, T. Kuester, S. Roessner, H. Kaufmann, B. Sang, V. Mogulsky, and S. Hofer, "Eetesthe enmap end-to-end simulation tool," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, vol. 5, no. 2, pp. 522–530, 2012.

[41] *HySAT Download*, University of Oldenburg, Oldenburg, Germany. [Online]. Available: http://hysat.informatik.uni-oldenburg.de/26273.html

[42] N. J. Higham, *Accuracy and stability of numerical algorithms*. Siam, 2002.

[43] R. E. Moore, *Interval analysis*. Prentice-Hall Englewood Cliffs, 1966, vol. 4.

[44] R. Rocher, D. Menard, P. Scalart, and O. Sentieys, "Analytical approach for numerical accuracy estimation of fixed-point systems based on smooth operations," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 59, no. 10, pp. 2326–2339, 2012.

[45] Z. Wang and A. C. Bovik, "Mean squared error: love it or leave it? a new look at signal fidelity measures," *Signal Processing Magazine, IEEE*, vol. 26, no. 1, pp. 98–117, 2009.

[46] C.-b. Design, "High-level synthesis with vivado hls," 2013.

[47] I. S. Association *et al.*, "Ieee std. 1666–2005 open systemc language reference manual," 2005.

[48] https://www.researchgate.net/publication/313036762_Fixed-Point_Singular_Value_Deco