Range Analysis of Matrix Factorization Algorithms for an Overflow Free Fixed-point Design

Article *in* Journal of Signal Processing Systems · July 2019

DOI: 10.1007/s11265-018-1394-3

2 authors:

Bibek Kabi
École Polytechnique
26 PUBLICATIONS **93** CITATIONS
SEE PROFILE

Anand S Sahadevan
Space Application Center, ISRO
8 PUBLICATIONS **72** CITATIONS
SEE PROFILE

Journal of Signal Processing Systems
https://doi.org/10.1007/s11265-018-1394-3

Range Analysis of Matrix Factorization Algorithms for an Overflow Free Fixed-point Design

Bibek Kabi[1] · **Anand S Sahadevan**[2]

Received: 21 December 2017 / Revised: 5 May 2018 / Accepted: 26 June 2018

Abstract

We consider the problem of enabling robust range estimation of matrix factorization algorithms like eigenvalue decomposition (EVD) algorithm and singular value decomposition (SVD) 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 and SVD 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. It was found that by the proposed range estimation approach, all the variables generated during the computation of EVD and SVD algorithms are bounded within ± 1. We also tried to contemplate the effect of different kinds of scaling factors on the bounds of the variables. Some benchmark hyperspectral data sets have been used to evaluate the efficiency of the proposed technique.

Keywords Affine arithmetic · Eigenvalue decomposition · Fixed-point arithmetic · Formal methods · Hyperspectral imaging · Integer bit-width allocation · Interval arithmetic · Overflow · Range analysis · Satisfiability-modulo-theory · Singular value decomposition

1 Introduction

Using the inherent elegance of matrix formulations, matrix factorization algrothms like eigenvalue decomposition (EVD) and singular value decomposition (SVD) have been the key building blocks in signal processing and control applications. The fixed-point development of EVD and SVD algorithms have been extensively studied in the past few years [4, 6, 20, 26, 30, 33, 39, 40, 43, 44, 46–48, 54, 57, 58, 61] 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 [36, 42]. This step has a significant impact on accuracy and hardware resources.

IWLs can be determined either using simulation [27, 31, 43] or by analytical (formal) methods [28, 32, 34, 60]. Existing works on fixed-point EVD and SVD have mainly used simulation-based approach for finding the IWLs

✉ Bibek Kabi
bibek@lix.polytechnique.fr

Anand S Sahadevan
anandss@sac.isro.gov.in

[1] Laboratoire d'Informatique de l'Ecole Polytechnique, CNRS, Ecole Polytechnique, 91128, Palaiseau, France

[2] Space Application Center, Indian Space Research Organisation, Bengaluru, India

Published online: 25 July 2018

[4, 6, 20, 26, 33, 39, 40, 43, 44, 46–48, 54, 57, 58, 61] 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 [63] 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 [7, 34] and therefore, it does not produce absolute bounds [1].

There are several limitations associated with analytical (formal) methods. An analytical method based on L_1 norm and transfer function is described in [3]. This method produces theoretical bounds that guarantee no overflow will occur, but the approach is only limited to linear time-invariant systems [7]. Interval arithmetic (IA) ignores correlation among signals resulting in an overestimation of ranges [32]. Affine arithmetic (AA) is a preferable approach that takes into account the interdependency among the signals [19], but ranges determined through AA explode during division if the range of divisor includes zero [28, 60]. IA also suffers from the same problem. Both IA and AA are pessimistic approaches leading to higher implementation cost [42]. Satisfiability modulo theory (SMT) produces tight ranges compared to IA and AA [28, 29]. However, it is computationally expensive and much slower as compared to IA and AA [52, 60]. 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.

This work is an extension of the recent published work [25] presented at DASIP 2017. In the conference paper, we proposed an analytical method for designing an overflow free fixed-point EVD algorithm. We evaluated the efficacy of the proposed approach through a case study of dimensionality reduction in hyperspectral images. As a extension of the conference work, we include in the current work the range analysis of SVD algorithm through the proposed approach. It is well-known that the magnitude of the singular values are bounded by (i) the square root of the product of matrix 1-norm and infinity-norm i.e. $\sqrt{\|A\|_1\|A\|_\infty}$ and (ii) the matrix Frobenius norm. The bounds on the intermediate variables of Hestenes SVD algorithm are derived analytically using (i) and (ii). We show that tighter bounds can be obtained using $\|A\|_F$ as the scaling factor as compared to $\sqrt{\|A\|_1\|A\|_\infty}$.The remainder of the paper is organised as follows. In section 2, the problem definition is discussed. Section 3 presents the analytical approach for deriving bounds for the variables of fixed-point EVD algorithm. The proof for the derivation of the bounds is also discussed. Section 4 shows the results obtained using fixed-point EVD. Section V and VI discusses how the proposed method can also be used to derive tight bounds for SVD algorithm. The results obtained using fixed-point SVD algorithm are illustrated in section VII.

2 EVD

Eigenvalue decomposition of a symmetric matrix A is given by

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

where Λ is a diagonal matrix of eigenvalues and the columns of the orthogonal matrix X are the eigenvectors of A. There are several algorithms developed in the literature for EVD of symmetric matrices [8, 17, 45]. Among all, two-sided Jacobi algorithm is most accurate and numerically stable [8, 9]. Most of the work attempted for dimensionality reduction via EVD uses two-sided Jacobi algorithm [11, 12, 18]. Apart from the accuracy and stability of Jacobi algorithm, it also has high degree potential for parallelism, and hence can be implemented on FPGA [40, 58]. In [20, 33, 39, 40, 46, 54, 58, 61] 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 1).

Algorithm 1

INITIALIZE $X = A$;
for $l = 1$ to n **do**
  for $i = 1$ to n **do**
    for $j = i + 1$ to n **do**
      $a = A(i, i)$;
      $b = A(j, j)$;
      $c = A(i, j) = A(j, i)$;
      $t = \frac{sign(\frac{b-a}{c}) \cdot |c|}{|\frac{b-a}{2}| + \sqrt{c^2 + (\frac{b-a}{2})^2}}$;
      $cs = 1/\sqrt{1 + t^2}$;
      $sn = cs \cdot t$;
      $A(i, i) = a - c \cdot t$;
      $A(j, j) = b + c \cdot t$;
      $A(i, j) = A(j, i) = 0$;
      for $k = 1$ to n **do**
        $tmp = A(i, k)$;
        $A(i, k) = cs \cdot tmp - sn \cdot A(j, k)$;
        $A(j, k) = sn \cdot tmp + cs \cdot A(j, k)$;
        $A(k, i) = A(i, k)$;
        $A(k, j) = A(j, k)$;
      end for
      for $k = 1$ to n **do**
        $tmp = X(k, i)$;
        $X(k, i) = cs \cdot tmp - sn \cdot X(k, j)$;
        $X(k, j) = sn \cdot tmp + cs \cdot X(k, j)$;
      end for
    end for
  end for
end for
for $i = 1$ to n **do**
  $\lambda_i = A(i, i)$;
end for

Jacobi method computes EVD of a symmetric matrix A by producing a sequence of orthogonally similar matrices, which eventually converges to a diagonal matrix [8] given by

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

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. Further in this section, we illustrate the issues associated with the existing range estimation methods through a case study i.e., dimensionality reduction of hyperspectral images using fixed-point EVD). The two-sided Jacobi algorithm is used for computing the EVD of the covariance matrices of hyperspectral images.

The most widely used algorithm for dimensionality reduction is principal component analysis (PCA). PCA requires computation of eigenvalues (λ) and eigenvectors (x). The eigenvectors are called principal axes or principal directions of the data. Projections of the data on the principal axes are called principal components. X is a new coordinate basis for the image. 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.

Dimensionality reduction has been a major topic of research in Hyperspectral imaging (HSI) analysis [35]. Conventionally, dimensionality reduction algorithms are developed in floating-point arithmetic to achieve high accuracy. Unfortunately, floating-point arithmetic increases the resources and power consumption. In HSI, the large number of spectral bands results in an excessively large volume of imaging data. Therefore, increased memory resources are needed to store imagery on board the satellite and a significant downlink transmission power will be required to transmit the raw images to the ground station. Both power and memory resources are major design constraints in spacecraft embedded applications and an efficient on-board hyperspectral image compression is essential to ensure that the design is within the power and memory budget of the mission. Converting the floating-point algorithms into fixed-point is an effective way to address these limitations [13, 16, 35]. Fixed-point architectures exhibit certain properties like no requirement of mantissa alignment, smaller memory and bus widths [35, 37, 38, 51]. Hence, the resource utilization and power consumption are reduced in fixed-point arithmetic. However, 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 [13, 35]. 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. [12] 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. [35] reported

Table 1 Comparison between the ranges computed by simulation, interval arithmetic and affine arithmetic with respect to the proposed approach for hyperion (Chilika) 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]

overflow issues while computing EVD in fixed-point arithmetic. Burger et al. [2] 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. 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, AVIRIS and HYDICE), handheld sensors (Landscape) and Synthetic (simulated EnMap). There are two datas acquired by Hyperion sensor. One of them is the Chilika Lake (latitude: 19.63 N - 19.68 N, longitude: 85.13 E -85.18 E) and its catchment areas [24] and the second one is a sequence of data over the Okavango Delta, Botswana. These are two scenes (Pavia University and Centre) acquired by the ROSIS sensor during a flight campaign over Pavia, northern Italy. These are three scenes (Indian Pines, Salinas and Kennedy Space Center (KSC)) gathered by the AVIRIS sensor. Landscape data is obtained from the database available from Stanford University [56]. The simulated EnMap image subset contains the Maktesh Ramon, Israel (30.57 N, 34.83 E) [53]. HYDICE sensor data is a scene of the Washington DC Mall. The sizes of the covariance matrix for the images are 120×120 for Hyperion (Chilika), 145×145 for Hyperion (Botswana), 103×103 for ROSIS (University), 102×102 for ROSIS (Centre), 200×200 for AVIRIS (Pines), 204×204 for AVIRIS (Salinas), 176×176 for AVIRIS (KSC), 148×148 for Landscape, 244×244 for simulated EnMap, 191×191 for HYDICE. Out of 120, 103, 200, 148, 244 and 191 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 1 and 2 shows the ranges obtained for Hyperion (Chilika) and ROSIS (University) 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 3) estimated using the simulation-based approach for Hyperion (Chilika) cannot avoid overflow in case of ROSIS (University). In other words, based on the ranges obtained using the simulation of Hyperion (Chilika) data one would allocate 24 bits to the integer part, but these number of

Table 2 Comparison between the ranges computed by simulation, interval arithmetic and affine arithmetic with respect to the proposed approach for ROSIS (University) 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 3 Comparison between the integer wordlengths required based on the ranges estimated by simulation-based approach shown in Tables 1 and 2.

Var	Hyperion (Chilika)	ROSIS (University)
A	24	25
a	22	22
b	24	25
λ	24	25

bits cannot avoid overflow in case of ROSIS (University). 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 1 and 2. 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 4 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 4 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 1 and 2. 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 1 and 2.

In order to combat this, 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 SMT)

Table 4 Range explosion of A while computing range using interval arithmetic and affine arithmetic.

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]	[−2.19e+21, 2.20e+21]			[−3.7e+28, 3.7e+28]
$n=6$	[0.18, 0.79]	[−6.90, 28.62]	[−6.96e+61, 7.08e+61]	[−2.5e+91, 2.6e+91]		[−4.5e+139, 4.5e+139]
$n=8$	[0.11, 0.75]	[−9.77, 48.29]	[−1.01e+126, 1.03e+126]	[−2.6e+187, 2.6e+187]	[−1.6e+301, 1.6e+301]	[−∞, ∞]
$n=10$	[0.09, 0.96]	[−16.62, 96.48]	[−4.77e+215, 4.88e+215]	[−∞, ∞]	[−∞, ∞]	[−∞, ∞]
$n=12$	[0.03, 0.97]	[−21.06, 139.76]	[−∞, ∞]	[−∞, ∞]	[−∞, ∞]	[−∞, ∞]

Table 5 Comparison between the ranges computed by simulation, interval arithmetic and satisfiability-modulo-theory with respect to the proposed approach for input matrix C.

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]

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

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

and

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

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 5 and 6. 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 [22].

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 [29]. 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 symmetric positive definite matrices of sizes 3×3 and 5×5 given by

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

and

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

The bounds obtained using IA for the input matrices Y and Z are shown in Table 7. The bounds are unnecessarily large compared to the actual bounds produced by the simulation-based approach shown in Table 8. Now, the input matrices are scaled through the upper bound of their spectral norm to

Table 6 Comparison between the ranges computed by simulation, interval arithmetic and satisfiability-modulo-theory with respect to the proposed approach for input matrix D.

Var	Simulation	IA	SMT	Proposed
D	[0, 55.68]	[−147.5, 151.9]	[−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	[−0.70, 0.70]	[−2.36, 2.36]	[−1.54, 1.54]	[−1, 1]
λ	[11.27, 55.68]	[−68.5, 151.96]	[−12.6, 103.3]	[0, 1]

Table 7 Ranges computed by interval arithmetic for input matrices Y and Z.

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

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

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

and

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

The ranges obtained for the scaled input matrices are shown in Table 9. 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 10). The difference in unboundedness of the ranges shown in Tables 7 and 9 is not substantially large. This illustrates that the ranges obtained using IA are not independent of the number of iterations. Similar is the case for both AA and SMT. We can observe the phenomenon

Table 8 Ranges computed by simulation-based method for input matrices Y and Z.

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]	[6.9e−3, 526.54]

Table 9 Ranges computed by interval arithmetic for input matrices $\hat{Y}$ And $\hat{Z}$.

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

in Table 11 for one of the test hyperspectral data (simulated EnMAP). Inspite of the range of covariance matrix being [−3.71e−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 1, 2, 5, 6 and 11. 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.

3 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,

Table 10 Ranges computed by simulation-based method for input matrices $\hat{Y}$ And $\hat{Z}$.

Variables	Simulation ($\hat{Y}$)	Simulation ($\hat{Z}$)
$\hat{Y}$ or $\hat{Z}$	[−1.31e−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	[−1.31e−3, 0.31]	[−0.015, 0.34]
X	[−0.61, 1]	[−0.71, 1]
λ	[8.59e−04, 0.78]	[1.24e−5, 0.94]

Table 11 Comparison between the ranges computed by simulation, interval arithmetic and affine arithmetic with respect to the proposed approach for simulated EnMAP data with the range of the covariance matrix as [−3.71e−06, 0.032].

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	[1.24e−05, 0.942]	$[-\infty, \infty]$	$[-\infty, \infty]$	[0, 1]

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 [17, 23]

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}. \tag{9}$$

For symmetric matrices, the spectral norm $\|A\|_2$ in (34) can be replaced with the spectral radius $\rho(A)$.

Theorem 1 *Given the bounds for spectral norm as* $\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}$*, 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* k^{th} *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

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

where (10) follows from [17]. 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

$$\lambda_i = \|A(:, i)\|_2. \tag{11}$$

According to vector norm property we can say that

$$\|A(:, i)\|_\infty \leq \|A(:, i)\|_2, \tag{12}$$

where $\|A(:, i)\|_\infty$ is the maximum of the absolute of the elements in $A(:, i)$. From the upper bound of λ_i, (22) and (23) 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

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

According to (28), numerator (r) of t lies in the range $[-|c|, |c|]$. $|\frac{b-a}{2}|$ and $\sqrt{c^2 + (\frac{b-a}{2})^2}$ 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 $\sqrt{c^2 + (\frac{b-a}{2})^2}$ 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 $|\frac{b-a}{2}| + \sqrt{c^2 + (\frac{b-a}{2})^2}$. 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 $J^T AJ$. According to 2×2 symmetric Schur decomposition discussed in [17], 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 [41] 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 (33).

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

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 $[-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$. □

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

$$\hat{A}x = \hat{\lambda}x, \tag{15}$$

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 $\left\lceil \log_2(\sqrt{\|A\|_1\|A\|_\infty}) \right\rceil$ 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* k^{th} *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

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

where (16) follows from [17]. 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.

The matrix Frobenius norm is also an upper bound for the spectral norm of any matrix [17, 23]. The relationship for spectral norm and Frobenius norm is given by

$$\|A\|_2 \leq \|A\|_F \tag{17}$$

Uisng $m = \|A\|_F$ produces the same bounds as the scaling factor $m = \sqrt{\|A\|_1\|A\|_\infty}$ for Jacobi EVD algorithm.

4 Results Obtained Using Fixed-point EVD

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 12 and 13 show the comparison between the bounds on the variables obtained by existing range estimation methods with respect

Table 12 Comparison between the ranges computed by simulation, interval arithmetic and affine arithmetic with respect to the proposed approach for AVIRIS (Pines) 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]

to the proposed approach through the AVIRIS (Pines) and Landscape data sets. We can observe that the ranges estimated using the simulation for AVIRIS (Pines) data cannot avoid overflow in case of Landscape. This is quite apparent from the number of integer bits required, shown in Table 14. 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 [51, 52]. It is given by

$$\text{SQNR} = 10\log_{10}(\text{E}(|\lambda_{float}|^2))/(\text{E}(|\lambda_{float} - \lambda_{fixed}|^2)) \tag{18}$$

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 15. In Table 15, 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 14), wordlengths like 50, 40 or 32 bits would never fit. In such cases, with the proposed approach it 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) [62]. MSE between PCA images of fixed-point implementations with various WLs after derving the ranges through proposed approach are shown in Table 16. The required number of PCs for Hyperion (Chilika), ROSIS (University) and Landscape are 4, 5 and 2 respectively. The number of PCs explaining 99.0% variance in case of AVIRIS (Pines) and EnMap are relatively higher. Therefore, the Table 16 only exhibits the results of Hyperion (Chilika), ROSIS (University) and Landscape. Similar results were obtained for Hyperion (Botswana), AVIRIS (Pines), AVIRIS (Salinas), AVIRIS (KSC), EnMap and HYDICE. 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 [14]. We have used SystemC (mimics hardware description language VHDL and Verilog) to develop the fixed-point code. Using the HLS tool, the SystemC fixed-point code is transformed into a hardware IP (intellectual property) described in Verilog. The fixed-point IP generator of Jacobi algorithm takes array as

Table 13 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 14 Comparison between the integer wordlengths required based on the ranges estimated by simulation-based approach shown in Tables 12 And 13.

Var	AVIRIS (Pines)	Landscape
A	25	116
a	24	116
b	25	116
λ	25	116

input argument. Hence, this argument is transformed into a Block RAM (BRAM) interface with appropriate address, enable and write signal to access the Xilinx memory. We compare the resource utilization (in terms of flip flops (FFs), look-up-tables (LUTs) and power) of simulation approach with respect to the proposed approach (for the same level of accuracy) through the test hyperspectral data sets. The resource utilization for the fixed-point design has been generated through post-synthesis simulation in Vivado design suite. The comparative study is illustrated in Table 17. 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.

In the following section, we illustrate how the proposed method also produces robust and tight analytical bounds for variables of singular value decomposition algorithm.

5 SVD

Similar to EVD, there exists another significant matrix factorization method known as singular value decomposition (SVD). A SVD of a real $m \times n$ matrix A (data matrix) is its factorization into the product of three matrices :

$$A = U\Sigma V^{\mathrm{T}} \tag{19}$$

where $U(m \times m)$ and $V(n \times n)$ are orthogonal matrices and $\Sigma(m \times n)$ is a rectangular diagonal matrix. It is absolutely well-known that EVD can only be applied to square matrices for example covariance matrices ($A^{\mathrm{T}}A$, if $m > n$) or (AA^{T}, if $m < n$) whereas SVD can be applied directly on the data matrix (A). Speaking of which, SVD is numerically more accurate than EVD [10, 21]. This is because, if the condition number of the data matrix (A) is in the order of 10^{-5}, the condition number of the covariance matrix will be the square of the condition number of A and computing its EVD will result in loss of precision compared to direct SVD. This makes SVD more general and it covers a wide range of applications from signal processing [49], bioinformatics [15], robotics [59]. Shi et al. [54] implemented fixed-point SVD algorithm where the ranges of the variables were estimated using simulation based method. Liu et al. [33] developed application-specific instruction set processor for system-on-chip (SoC) implementation of modern signal processing algorithms. IWLs were estimated using the statistics (dynamic range, the mean, standard deviation and the distribution histogram) obtained from floating-point simulation. Nikolić et al. [43] implemented fixed-point Jacobi SVD algorithm using the same simulation-based profiling. Pradhan et al. [46] developed fixed-point Hestenes SVD algorithm for computing eigen faces. Integer wordlengths were estimated using the statistics obtained from floating-point simulations. Nearly all the work on fixed-point SVD uses simulation based profiling to decide the interger wordlengths [4, 20, 55]. Like EVD, SVD based PCA is also one of the most widely used algorithm for dimensionality reduction in hyperspectral imaging [5, 49, 50]. The effectiveness of the fixed-point SVD algorithm after obtaining the ranges through the proposed approach has been evaluated using some hyperspectral datasets.

Table 15 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 (Chilika)	176.76	106.44	78.03
Hyperion (Botswana)	178.45	103.21	79.31
ROSIS (University)	180.13	134.79	74.96
ROSIS (Centre)	183.15	134.91	76.81
Landscape	180.65	122.36	77.18
AVIRIS (Pines)	178.54	110.76	76.43
AVIRIS (Salinas)	181.27	114.89	77.54
AVIRIS (KSC)	178.71	110.87	78.84
EnMap	180.67	130.24	78.36
HYDICE	168.35	147.11	78.67

One of the most prominent algorithms to compute SVD is Hestenes algorithm. Hestenes algorithm (Algorithm 2) is based on Jacobi method implicitly applied for the symmetric eigenvalue problem to $A^{\mathrm{T}}A$ [9, 10]. The algorithm uses a Jacobi rotation to make the off-diagonal entries of 2×2 submatrix of $A^{\mathrm{T}}A$ zero. Line 1 of Algorithm 2 shows that the matrix V is initially an identity matrix, but gradually it is the accumulation of all column rotations. Once all columns of AV are orthogonal, we get the relation $AV = U\Sigma = U \cdot diag(\sigma_1, \sigma_2, ..., \sigma_n)$ where U, V are orthogonal matrices containing the left and right singular vectors and Σ is a diagonal matrix containing the singular values. Thus, $A = U\Sigma V^T$. It is well-known that the magnitude of the singular values are bounded by (i) the square root

Table 16 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 (Chilika)				ROSIS (University)					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

of the product of matrix 1-norm and infinity-norm i.e. $\sqrt{\|A\|_1\|A\|_\infty}$ and (ii) the matrix Frobenius norm. The bounds on the intermediate variables of Hestenes SVD algorithm are derived analytically using (i) and (ii). In case of input matrices, where $\sqrt{\|A\|_1\|A\|_\infty}$ or $\|A\|_F$ is high, larger integer wordlengths (IWLs) are required leading to implementation over-cost. In such cases, tight analytical bounds for the variables can be derived by using either (i) or (ii) as the scaling factor. We show that tighter bounds can be obtained using $\|A\|_F$ as the scaling factor as compared to $\sqrt{\|A\|_1\|A\|_\infty}$.

Algorithm 2

1: INITIALIZE $V = I$;
2: **for** $l = 1$ to n **do**
3: **for** $i = 1$ to n **do**
4: **for** $j = i + 1$ to n **do**
5: $a = A(:, i)^{\mathrm{T}} A(:, i)$;
6: $b = A(:, j)^{\mathrm{T}} A(:, j)$;
7: $c = A(:, i)^{\mathrm{T}} A(:, j)$;
8: $\zeta = (b - a)/(2c)$;
9: $t = sign(\zeta)/(|\zeta| + \sqrt{1 + \zeta^2})$;
10: $cs = 1/\sqrt{1 + t^2}$;
11: $sn = cs \cdot t$;
12: **for** $k = 1$ to n **do**
13: $tmp = A(k, i)$;
14: $A(k, i) = cs \cdot tmp - sn \cdot A(k, j)$;
15: $A(k, j) = sn \cdot tmp + cs \cdot A(k, j)$;
16: **end for**
17: **for** $k = 1$ to n **do**
18: $tmp = V(k, i)$;
19: $V(k, i) = cs \cdot tmp - sn \cdot V(k, j)$;
20: $V(k, j) = sn \cdot tmp + cs \cdot V(k, j)$;
21: **end for**
22: **end for**
23: **end for**
24: **end for**
25: **for** $i = 1$ to n **do**
26: $\sigma_i = \|A(:, i)\|_2$;
27: **end for**
28: **for** $i = 1$ to n **do**
29: $U(:, i) = A(:, i)/\sigma_i$;
30: **end for**

6 Bounds on Variables

In this section we show that given a size of the spectral norm of the input matrix, the bounds for variables can be derived analytically.

6.1 Bounds on Variables for $\|A\|_2 \leq \sqrt{\|A\|_1\|A\|_\infty}$

The upper bound for the spectral norm of any matrix is given by [17, 23]:

$$\|A\|_2 \leq \sqrt{\|A\|_1\|A\|_\infty} \tag{20}$$

Thus the size of the spectral norm for input matrices can be represented as $\|A\|_2 \in [0, \sqrt{\|A\|_1\|A\|_\infty}]$.

Theorem 3 *Given the bounds for spectral norm of input matrix as* $[0, \sqrt{\|A\|_1\|A\|_\infty}]$, *the Hestenes SVD algorithm applied to A has the following bounds for the variables for all i, j, x and y:*

- $[A]_{xy} \in [-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$
- $t \in [-1, 1]$
- $cs \in [0, 1]$
- $sn \in [-1, 1]$
- $[U]_{xy} \in [-1, 1]$
- $[V]_{xy} \in [-1, 1]$
- $a \in [0, r\|A\|_1\|A\|_\infty]$
- $b \in [0, r\|A\|_1\|A\|_\infty]$
- $c \in [-r\|A\|_1\|A\|_\infty, r\|A\|_1\|A\|_\infty]$
- $[\sigma_i]_x \in [0, \sqrt{\|A\|_1\|A\|_\infty}]$,

where i, j denotes the iteration number and $[]_x$ *and* $[]_{xy}$ *denote the* x^{th} *component of a vector and* xy^{th} *component of a matrix respectively.*

Proof We start by bounding the elements of the input matrix as

$$\max_{xy} |[A]_{xy}| \leq \|A\|_2 \leq \sqrt{\|A\|_1\|A\|_\infty}, \tag{21}$$

where (21) follows from [17]. Hence, the elements of A are in the range $[-\sqrt{\|A\|_1\|A\|_\infty}, \sqrt{\|A\|_1\|A\|_\infty}]$. Line 5, 6 and 7 in Algorithm 2 computes the (i, j) submatrix of $A^{\mathrm{T}}A$. According to Line 5 in Algorithm 2, a is computed

Table 17 Comparison between hardware cost (%) Of fixed-point jacobi algorithm after determining ranges through proposed and simulation approaches.

WL	Hyperion (Chilika)			ROSIS (University)			AVIRIS (Pines)		
Proposed									
	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									
	FF	LUTs	Power	FF	LUTs	Power	FF	LUTs	Power
50	8	23	2.59	8	23	2.64	8	23	2.64

from the square of the entries of each column of matrix A ($a = A(:,i)^{\mathrm{T}}A(:,i)$). Hence, we can say that the lower bound of $A(:,i)^{\mathrm{T}}A(:,i)$ is 0. If we can find the bound of $\sqrt{A(:,i)^{\mathrm{T}}A(:,i)}$ or $\|A(:,i)\|_2$, then the upper bound of $A(:,i)^{\mathrm{T}}A(:,i)$ can be fixed. Since $\|A(:,i)\|_2$ is the Euclidean norm of column i and $\|A\|_F$ is the Euclidean norm of the whole matrix, we can say that

$$\|A(:,i)\|_2 \leq \|A\|_F \tag{22}$$

The relationship of Frobenius norm and 2-norm of a matrix [17] is

$$\|A\|_F \leq \sqrt{r}\|A\|_2 \tag{23}$$

where r is the rank of the matrix. From (21), (22) and (23) we can derive the bounds for $\|A(:,i)\|_2$ as

$$\|A(:,i)\|_2 \leq \|A\|_F \leq \sqrt{r}\|A\|_2 \leq \sqrt{r}\sqrt{\|A\|_1\|A\|_\infty} \tag{24}$$

Thus, the range of a is $[0, r\|A\|_1\|A\|_\infty]$. Similarly, the range of b (Line 6 of Algorithm 2) is also $[0, r\|A\|_1\|A\|_\infty]$. The off-diagonal entry c of 2×2 submatrix is formed from multiplication of $A(:,i)^{\mathrm{T}}$ and $A(:,j)$ as shown in Line 7 of Algorithm 2. Let us assume $i = 1$, $j = 2$ and $A(:,1) = \mathbf{m}$, $A(:,2) = \mathbf{n}$ respectively. Thus a is $\mathbf{m}^{\mathrm{T}}\mathbf{m}$, b is $\mathbf{n}^{\mathrm{T}}\mathbf{n}$ and c is $\mathbf{m}^{\mathrm{T}}\mathbf{n}$. We know that $\mathbf{(m\text{-}n)}^{\mathrm{T}}\mathbf{(m\text{-}n)} \geq 0$. Since $\mathbf{m}$ and $\mathbf{n}$ are column vectors, we can say that

$$\mathbf{m}^{\mathrm{T}}\mathbf{m} - \mathbf{m}^{\mathrm{T}}\mathbf{n} - \mathbf{n}^{\mathrm{T}}\mathbf{m} + \mathbf{n}^{\mathrm{T}}\mathbf{n} \geq 0 \tag{25}$$

Thus, the relationship among a, b and c is given by

$$c \leq \frac{a+b}{2} \tag{26}$$

According to (26) the range of c is $[-r\|A\|_1\|A\|_\infty, r\|A\|_1\|A\|_\infty]$ [64]. Line 9 in Algorithm 2 computes t which can be expressed as

$$t = \frac{sign(\frac{b-a}{c}) \cdot |c|}{|\frac{b-a}{2}| + \sqrt{c^2 + (\frac{b-a}{2})^2}} \tag{27}$$

Let $t=p/q$ such that

$$p = sign\left(\frac{b-a}{c}\right) \cdot \frac{|c|}{2} \text{ and}$$
$$q = \frac{1}{2}\left(\left|\frac{b-a}{2}\right| + \sqrt{c^2 + \left(\frac{b-a}{2}\right)^2}\right) \tag{28}$$

Based on the ranges of a, b and c and according to (26) it can be concluded that

$$0 \leq \left|\frac{b-a}{2}\right| \leq r\|A\|_1\|A\|_\infty \tag{29}$$

and also

$$c^2 + \left(\frac{b-a}{2}\right)^2 \leq \left(\frac{b+a}{2}\right)^2 + \left(\frac{b-a}{2}\right)^2 = \frac{a^2+b^2}{2} \leq r^2\|A\|_1{}^2\|A\|_\infty{}^2 \tag{30}$$

From (29), (30) and addition rule of interval arithmetic [41], we can say that

$$|\frac{b-a}{2}| + \sqrt{c^2 + (\frac{b-a}{2})^2} \leq 2r\|A\|_1\|A\|_\infty \tag{31}$$

$|\frac{b-a}{2}| + \sqrt{c^2 + (\frac{b-a}{2})^2}$ lie in the range $[0, 2r\|A\|_1\|A\|_\infty]$. Thus, an upper bound for $|c|$ can be fixed as

$$|c| \leq \left|\frac{b-a}{2}\right| + \sqrt{c^2 + \left(\frac{b-a}{2}\right)^2} \tag{32}$$

Thus, from (28) and (32) we can conclude that t lies in the range $[-1, 1]$ [64].

Hestenes SVD method makes the off-diagonal entries of 2×2 submatrix of $A^{\mathrm{T}}A$ zero by overwriting $A^{\mathrm{T}}A$ with $(AJ)^{\mathrm{T}}(AJ)$, where J is the Jacobi rotation. According to 2×2 symmetric schur decomposition discussed in [17], cs and sn are cosine and sine trigonometric functions. Thus, the bounds for cs and sn are $[-1, 1]$. Line 10 in Algorithm 2 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 [41] the range of sn can be fixed as $[-1, 1]$.

Line 26 in Algorithm 2 shows the computation of singular values. We know that $\|A\|_2 \leq \sqrt{\|A\|_1\|A\|_\infty}$. So, the upper bound for the singular values is $\sqrt{\|A\|_1\|A\|_\infty}$. Since σ_i is formed as a result of norm operation the range of σ_i is $[0, \sqrt{\|A\|_1\|A\|_\infty}]$. Line 29 in Algorithm 2 shows the formation of left singular vectors U from the final A. As a result of the normalization step, the left singular vectors have unity norm, and hence all elements of U are in the range $[-1, 1]$ following (33).

$$\|U(:, i)\|_\infty \leq \|U(:, i)\|_2 = 1 \tag{33}$$

Similarly like left singular vectors, the right singular vectors also have unity norm, and hence all elements of V are in the range $[-1, 1]$. □

6.2 Bounds on Variables for $\|A\|_2 \leq \|A\|_F$

The matrix Frobenius norm is also an upper bound for the spectral norm of any matrix [17, 23]. The relationship for spectral norm and Frobenius norm is given by

$$\|A\|_2 \leq \|A\|_F \tag{34}$$

Thus the size of the spectral norm for input matrices can be represented as $\|A\|_2 \in [0, \|A\|_F]$.

Theorem 4 *Given the bounds for spectral norm of input matrix as* $[0, \|A\|_F]$*, the Hestenes SVD algorithm applied to A has the following bounds for the variables for all i, j, x and y:*

- $[A]_{xy} \in [-\|A\|_F, \|A\|_F]$
- $t \in [-1, 1]$
- $cs \in [0, 1]$
- $sn \in [-1, 1]$
- $[U]_{xy} \in [-1, 1]$
- $[V]_{xy} \in [-1, 1]$
- $a \in [0, \|A\|^2{}_F]$
- $b \in [0, \|A\|^2{}_F]$
- $c \in [-\|A\|^2{}_F, \|A\|^2{}_F]$
- $[\sigma_i]_x \in [0, \|A\|_F]$,

where i, j denotes the iteration number and $[]_x$ *and* $[]_{xy}$ *denote the* x^{th} *component of a vector and* xy^{th} *component of a matrix respectively.*

Proof The bounds for the variables in Theorem 2 can be derived similarly as shown in the proof of Theorem 1. The bounds on a, b and c does not involve any rank of the input matrix. As discussed earlier according to (22), $\|A(:, i)\|_2 \leq \|A\|_F$. In Theorem 2, since Frobenius norm is considered as the upper bound for the largest singular value, the range of a is $[0, \|A\|^2{}_F]$. Similarly the range of b and c are $[0, \|A\|^2{}_F]$ and $[-\|A\|^2{}_F, \|A\|^2{}_F]$ respectively. □

In case of input matrices, where $\sqrt{\|A\|_1\|A\|_\infty}$ or $\|A\|_F$ is high, larger IWLs are required leading to implementation over-cost. In such cases, $\sqrt{\|A\|_1\|A\|_\infty}$ or $\|A\|_F$ can be used as scaling factor to obtain tight analytical bounds for the variables.

6.3 Bounds on Variables if $\|A\|_2 \leq 1$

Kishore et al. [30] discussed that by choosing small values for the input matrix, the values of the intermediate variables of SVD algorithm at all subsequent iterations can be kept sufficiently small to avoid overflow. Thus we scale the input matrix through an upper bound ($m = \sqrt{\|A\|_1\|A\|_\infty}$ or $m = \|A\|_F$) of the spectral norm to keep $\|\hat{A}\|_2 \leq 1$, where $\hat{A}$ is the matrix obtained after scaling. If the input matrix is scaled as $\hat{A} = \frac{A}{m}$, the SVD of matrix $\hat{A}$ is

$$\hat{A} = U\hat{\Sigma}V^{\mathrm{T}}, \tag{35}$$

where $A = U\Sigma V^{\mathrm{T}}$ and $\hat{\Sigma} = \frac{\Sigma}{m}$. U and V are orthogonal matrices each column of which contain the left and right singular vectors and Σ is a diagonal matrix containing the singular values. After scaling through a scalar value, the original singular vectors do not change. The original singular values can be recovered as $\Sigma = m\hat{\Sigma}$. Two significant points in this complete process are that scaling of the input symmetric matrix and recovery of the original singular values is carried out in floating-point arithmetic to ensure higher accuracy.

Theorem 5 *Given the scaling factor as* $m = \sqrt{\|A\|_1\|A\|_\infty}$*, the Hestenes SVD algorithm applied to* $\hat{A}$ *has the following bounds for the variables for all i, j, x and y:*

- $[\hat{A}]_{xy} \in [-1, 1]$
- $t \in [-1, 1]$
- $cs \in [0, 1]$
- $sn \in [-1, 1]$
- $[U]_{xy} \in [-1, 1]$
- $[V]_{xy} \in [-1, 1]$
- $a \in [0, r]$
- $b \in [0, r]$
- $c \in [-r, r]$
- $[\sigma_i]_x \in [0, 1]$,

Table 18 Signal-to-quantization-noise-ratio Of singular values 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 (Chilika)	175.81	106.49	79.31
Hyperion (Botswana)	176.23	102.45	77.98
ROSIS (University)	182.33	136.81	77.65
ROSIS (Centre)	183.15	136.13	75.32
Landscape	180.61	123.34	78.28
AVIRIS (Pines)	177.64	111.78	78.43
AVIRIS (Salinas)	183.29	116.89	79.74
AVIRIS (KSC)	176.75	110.88	78.83
EnMap	181.67	133.24	79.24
HYDICE	167.98	145.27	74.21

where i, j denotes the iteration number and $[]_x$ and $[]_{xy}$ denote the x^{th} component of a vector and xy^{th} component of a matrix respectively.

Proof In Theorem 3, $\sqrt{\|A\|_1\|A\|_\infty}$ is considered as the upper bound on the spectral norm before scaling the input matrix. Thus, the bounds of Theorem 1 apply for variables before scaling. The bounds for the variables in Theorem 3 can be derived similarly as shown in the proof of Theorem 1. □

Theorem 6 *Given the scaling factor as $m = \|A\|_F$, the Hestenes SVD algorithm applied to $\hat{A}$ has the following bounds for the variables for all i, j, x and y:*

- $[\hat{A}]_{xy} \in [-1, 1]$
- $t \in [-1, 1]$
- $cs \in [0, 1]$
- $sn \in [-1, 1]$
- $[U]_{xy} \in [-1, 1]$
- $[V]_{xy} \in [-1, 1]$
- $a \in [0, 1]$
- $b \in [0, 1]$
- $c \in [-1, 1]$
- $[\sigma_i]_x \in [0, 1]$,

where i, j denotes the iteration number and $[]_x$ and $[]_{xy}$ denote the x^{th} component of a vector and xy^{th} component of a matrix respectively.

Proof In Theorem 4, $\|A\|_F$ is considered as the upper bound on the spectral norm before scaling the input matrix. Thus, the bounds of Theorem 2 apply for variables before scaling. The bounds for the variables in Theorem 4 can be derived similarly as shown in the proof of Theorem 1. The bounds of a, b and c does not involve any rank of the input matrix as in Theorem 2. Much tighter bounds are obtained for a, b and c while using $m = \|A\|_F$ as the scaling factor as compared to $m = \sqrt{\|A\|_1\|A\|_\infty}$. □

7 Results Obtained Using Fixed-point SVD

In this section, we present a few hyperspectral data sets, and we compare the bounds on variables of Hestenes SVD algorithm produced by the existing range estimation methods and the proposed approach. SQNR of the singular values obtained through the proposed fixed-point design is shown in Table 18. MSE between PCA images of fixed-point implementations with various WLs after derving the ranges through proposed approach are shown in Table 19. Table 18 shows high magnitudes of SQNR and Table 19 shows negligibly smaller values of MSE which exhibit that the set of ranges obtained based on our proposed method are sufficient for avoiding overflow for any input matrix. 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 20. There is a noteworthy difference in the hardware resources. The hardware resources in case of simulation approach are

Table 19 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 (Chilika)				ROSIS (University)					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

Table 20 Comparison between hardware cost (%) Of fixed-point hestenes algorithm after determining ranges through proposed and simulation approaches.

WL	Hyperion (Chilika)			ROSIS (University)			AVIRIS (Pines)		
Proposed									
	FF	LUTs	Power	FF	LUTs	Power	FF	LUTs	Power
32	1.61	6.17	0.411	1.61	6.29	0.41	1.62	6.44	0.43
Simulation									
	FF	LUTs	Power	FF	LUTs	Power	FF	LUTs	Power
50	7	21	2.49	7	21	2.61	7	22	2.63

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.

8 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 matrix factorization algorithms like EVD and SVD. 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 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.

Acknowledgements We would like to thank the editorial board of DASIP 2017 for considering our work in Special Issue of the Journal of Signal Processing Systems.

References

1. Banciu, A. (2012). A stochastic approach for the range evaluation. Ph.D. thesis, Université, Rennes 1.
2. Burger, J., & Gowen, A. (2011). Data handling in hyperspectral image analysis. *Chemometrics and Intelligent Laboratory Systems*, *108*(1), 13–22.
3. Carletta, J., Veillette, R., Krach, F., Fang, Z. (2003). Determining appropriate precisions for signals in fixed-point iir filters. In Proceedings of the 40th annual design automation conference (pp. 656–661). ACM.
4. Cesear, T., & Uribe, R. (2005). Exploration of least-squares solutions of linear systems of equations with fixed-point arithmetic hardware. In Software defined radio technical conference,(SDR'05).
5. Chaudhari, A.J., Darvas, F., Bading, J.R., Moats, R.A., Conti, P.S., Smith, D.J., Cherry, S.R., Leahy, R.M. (2005). Hyperspectral and multispectral bioluminescence optical tomography for small animal imaging. *Physics in Medicine & Biology*, *50*(23), 5421.
6. Chen, Y.L., Zhan, C.Z., Jheng, T.J., Wu, A.Y.A. (2013). Reconfigurable adaptive singular value decomposition engine design for high-throughput mimo-ofdm systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *21*(4), 747–760.
7. Constantinides, G., Kinsman, A.B., Nicolici, N. (2011). Numerical data representations for fpga-based scientific computing. *IEEE Design & Test of Computers*, *28*(4), 8–17.
8. Datta, B.N. (2010). Numerical linear algebra and applications. Siam.
9. Demmel, J., & Veselic, K. (1992). Jacobi's method is more accurate than qr. *SIAM Journal on Matrix Analysis and Applications*, *13*(4), 1204–1245.
10. Demmel, J.W. (1997). Applied numerical linear algebra. Siam.
11. Egho, C., & Vladimirova, T. (2012). Hardware acceleration of the integer karhunen-loève transform algorithm for satellite image compression. In Geoscience and remote sensing symposium (IGARSS), 2012 IEEE international (pp. 4062–4065). IEEE.
12. Egho, C., & Vladimirova, T. (2014). Adaptive hyperspectral image compression using the klt and integer klt algorithms. In 2014 NASA/ESA conference on Adaptive hardware and systems (AHS) (pp. 112–119). IEEE.
13. Egho, C., Vladimirova, T., Sweeting, M.N. (2012). Acceleration of karhunen-loève transform for system-on-chip platforms. In 2012 NASA/ESA conference on adaptive hardware and systems (AHS) (pp. 272–279). IEEE.
14. Feist, T. (2012). Vivado design suite. White Paper 5.
15. Feng, C.W., Hu, T.K., Chang, J.C., Fang, W.C. (2014). A reliable brain computer interface implemented on an fpga for a mobile dialing system. In 2014 IEEE international symposium on circuits and systems (ISCAS) (pp. 654–657). IEEE.

16. Fry, T.W., & Hauck, S. (2002). Hyperspectral image compression on reconfigurable platforms. In 2002. Proceedings. 10th annual IEEE symposium on field-programmable custom computing machines (pp. 251–260). IEEE.
17. Golub, G.H., & Van Loan, C.F. (2012). *Matrix computations* Vol. 3. Baltimore: JHU Press.
18. Gonzalez, C., Lopez, S., Mozos, D., Sarmiento, R. 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.
19. Goubault, E., & Putot, S. (2015). A zonotopic framework for functional abstractions. *Formal Methods in System Design*, *47*(3), 302–360.
20. Grammenos, R.C., Isam, S., Darwazeh, I. (2011). Fpga design of a truncated svd based receiver for the detection of sefdm signals. In 2011 IEEE 22nd international symposium on Personal indoor and mobile radio communications (PIMRC) (pp. 2085–2090). IEEE.
21. Hall, P., Marshall, D., Martin, R. (2000). Merging and splitting eigenspace models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(9), 1042–1049.
22. Herde, C., Eggers, A., Fränzle, M., Teige, T. (2008). Analysis of hybrid systems using hysat. In 2008. ICONS 08. Third international conference on Systems (pp. 196–201). IEEE.
23. Higham, N.J. (2002). Accuracy and stability of numerical algorithms. Siam.
24. Kabi, B., Sahadevan, A.S., Mohanty, R., Routray, A., Das, B.S., Mohanty, A. An overflow-free fixed-point singular value decomposition algorithm for dimensionality reduction of hyperspectral images.
25. Kabi, B., Sahadevan, A.S., Pradhan, T. (2017). An overflow free fixed-point eigenvalue decomposition algorithm: Case study of dimensionality reduction in hyperspectral images. In 2017 Conference on design and architectures for signal and image processing (DASIP) (pp. 1–9). IEEE.
26. Kasap, S., & Redif, S. (2014). Novel field-programmable gate array architecture for computing the eigenvalue decomposition of para-hermitian polynomial matrices. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *22*(3), 522–536.
27. Kim, S., Kum, K.I., Sung, W. (1998). Fixed-point optimization utility for c and c++ based digital signal processing programs. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *45*(11), 1455–1464.
28. Kinsman, A.B., & Nicolici, N. (2010). Bit-width allocation for hardware accelerators for scientific computing using sat-modulo theory. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, *29*(3), 405–413.
29. Kinsman, A.B., & Nicolici, N. (2011). Automated range and precision bit-width allocation for iterative computations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *9*(30), 1265–1278.
30. Kota, K., & Cavallaro, J.R. (1993). Numerical accuracy and hardware tradeoffs for cordic arithmetic for special-purpose processors. *IEEE Transactions on Computers*, *42*(7), 769–779.
31. Kum, K.I., Kang, J., Sung, W. (2000). Autoscaler for c: An optimizing floating-point to integer c program converter for fixed-point digital signal processors. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *47*(9), 840–848.
32. Lee, D.U., Gaffar, A.A., Cheung, R.C., Mencer, O., Luk, W., Constantinides, G., et al. (2006). Accuracy-guaranteed bit-width optimization. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, *25*(10), 1990–2000.
33. Liu, Z., Dickson, K., McCanny, J.V. (2005). Application-specific instruction set processor for soc implementation of modern signal processing algorithms. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *52*(4), 755–765.
34. López, J., Carreras, C., Nieto-Taladriz, O., et al. (2007). Improved interval-based characterization of fixed-point lti systems with feedback loops. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, *26*(11), 1923–1933.
35. Lopez, S., Vladimirova, T., Gonzalez, C., Resano, J., Mozos, D., Plaza, A. (2013). The promise of reconfigurable computing for hyperspectral imaging onboard systems: a review and trends. *Proceedings of the IEEE*, *101*(3), 698–722.
36. Martel, M., Najahi, A., Revy, G. (2014). Toward the synthesis of fixed-point code for matrix inversion based on cholesky decomposition. In 2014 conference on Design and architectures for signal and image processing (DASIP) (pp. 1–8). IEEE.
37. Menard, D., Chillet, D., Sentieys, O. (2006). Floating-to-fixed-point conversion for digital signal processors. *EURASIP Journal on Applied Signal Processing*, *2006*, 77–77.
38. Menard, D., Rocher, R., Sentieys, O. (2008). Analytical fixed-point accuracy evaluation in linear time-invariant systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *55*(10), 3197–3208.
39. Milford, D., & Sandell, M. (2014). Singular value decomposition using an array of cordic processors. *Signal Processing*, *102*, 163–170.
40. Mohanty, R., Anirudh, G., Pradhan, T., Kabi, B., Routray, A. (2014). Design and performance analysis of fixed-point jacobi svd algorithm on reconfigurable system. *IERI Procedia*, *7*, 21–27.
41. Moore, R.E. (1966). *Interval analysis* Vol. 4. Englewood Cliffs: Prentice-Hall.
42. Nehmeh, R., Menard, D., Banciu, A., Michel, T., Rocher, R. (2014). Integer word-length optimization for fixed-point systems. In 2014 IEEE nternational conference on Acoustics, speech and signal processing (ICASSP) (pp. 8321–8325). IEEE.
43. Nikolić, Z., Nguyen, H.T., Frantz, G. (2007). Design and implementation of numerical linear algebra algorithms on fixed point dsps. EURASIP Journal on Advances in Signal Processing 2007.
44. Pradhan, T., Kabi, B., Mohanty, R., Routray, A. (2016). Development of numerical linear algebra algorithms in dynamic fixed-point format: a case study of lanczos tridiagonalization. *International Journal of Circuit Theory and Applications*, *44*(6), 1222–1262.
45. Pradhan, T., Routray, A., Kabi, B. (2013). Comparative evaluation of symmetric svd algorithms for real-time face and eye tracking. In Matrix information geometry (pp. 323–340). Springer.
46. Pradhan, T., Routray, A., Kabi, B. (2013). Fixed-point hestenes svd algorithm for computing eigen faces. *International Journal OF Circuits Systems and Signal Processing*, *7*(6), 312–321.
47. Rahmati, M., Sadri, M.S., Naeini, M.A. (2008). Fpga based singular value decomposition for image processing applications. In ASAP 2008. International conference on application-specific systems, architectures and processors, 2008 (pp. 185–190). IEEE.
48. Reddy, K., & Herron, T. (2001). Computing the eigen decomposition of a symmetric matrix in fixed-point arithmetic. In 10Th annual symposium on multimedia communications and signal processing, IEEE bangalore section, bangalore, india.
49. Ren, Y., Liao, L., Maybank, S., Zhang, Y., Liu, X. (2017). Hyperspectral image spectral-spatial feature extraction via tensor principal component analysis. IEEE Geoscience and Remote Sensing Letters.
50. Reshma, R., Sowmya, V., Soman, K. (2017). Effect of legendre–fenchel denoising and svd-based dimensionality reduction algorithm on hyperspectral image classification. Neural Computing and Applications pp. 1–10.
51. Rocher, R., Menard, D., Scalart, P., Sentieys, O. (2012). Analytical approach for numerical accuracy estimation of fixed-point systems based on smooth operations. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *59*(10), 2326–2339.

52. Sarbishei, O., & Radecka, K. (2013). On the fixed-point accuracy analysis and optimization of polynomial specifications. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, *32*(6), 831–844.
53. Segl, K., Guanter, L., Rogass, C., Kuester, T., Roessner, S., Kaufmann, H., Sang, B., Mogulsky, V., Hofer, S. (2012). Eetes—the enmap end-to-end simulation tool. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *5*(2), 522–530.
54. Shi, C., & Brodersen, R.W. (2004). Automated fixed-point data-type optimization tool for signal processing and communication systems. In Design automation conference, 2004. Proceedings. 41st (pp. 478–483). IEEE.
55. Shih, W.Y., Liao, J.C., Huang, K.J., Fang, W.C., Cauwenberghs, G., Jung, T.P. (2013). An efficient vlsi implementation of on-line recursive ica processor for real-time multi-channel eeg signal separation. In Engineering in medicine and biology society (EMBC), 2013 35th annual international conference of the IEEE (pp. 6808–6811). IEEE.
56. Skauli, T., & Farrell, J. (2013). A collection of hyperspectral images for imaging systems research. In ISAndamp;t/SPIE electronic imaging (pp. 86,600c–86,600c). International society for optics and photonics.
57. Studer, C., Blosch, P., Friedli, P., Burg, A. (2007). Matrix decomposition architecture for mimo systems: Design and implementation trade-offs. In Conference record of the forty-first asilomar conference on Signals, systems and computers, 2007. ACSSC 2007 (pp. 1986–1990). IEEE.
58. Szecówka, P.M., & Malinowski, P. (2010). Cordic and svd implementation in digital hardware. In Mixed design of integrated circuits and systems (MIXDES), 2010 Proceedings of the 17th International Conference (pp. 237–242). IEEE.
59. Thieffry, M., Kruszewski, A., Goury, O., Guerra, T.M., Duriez, C. (2017). Dynamic control of soft robots. In IFAC World congress.
60. Vakili, S., Langlois, J.P., Bois, G. (2013). Enhanced precision analysis for accuracy-aware bit-width optimization using affine arithmetic. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, *32*(12), 1853–1865.
61. Wang, Y., Cunningham, K., Nagvajara, P., Johnson, J. (2010). Singular value decomposition hardware for mimo: State of the art and custom design. In 2010 international conference on reconfigurable computing and FPGAs (reconfig) (pp. 400–405). IEEE.
62. Wang, Z., & Bovik, A.C. (2009). Mean squared error: love it or leave it? a new look at signal fidelity measures. *Signal Processing Magazine, IEEE*, *26*(1), 98–117.
63. Wu, B., Zhu, J., Najm, F.N. (2006). Dynamic-range estimation. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*, *25*(9), 1618–1636.
64. Zheng, P., & Gao, X. (2013). Fixed-point cca algorithm applied to ssvep based bci system. In 2013 IEEE symposium on computational intelligence, cognitive algorithms, mind, and brain (CCMB) (pp. 107–114). IEEE.

Bibek Kabi is pursuing his PhD from LIX, Ecole Polytechnique, CNRS, Université Paris-Saclay, 91128 Palaiseau, France. His research interests include floating-point and fixed-point arithmetic, optimization of fixed-point formats in numerical programs, fixed-point numerical linear algebra, verification of cyber-physical systems and data mining.

Anand S Sahadevan received his PhD from Indian Institute of Technology Kharagpur, India in 2016. He is currently working as a scientist in Hyperspectral Technique Development Division of Space Applications Centre, Indian Space research Organization (ISRO). His research interests include hyperspectral image processing, machine learning, and chemometrics.