Exploration of Heterogeneous FPGAs for Mapping Linear Projection Designs

Christos-S. Bouganis, *Member, IEEE*, Iosifina Pournara, and Peter Y. K. Cheung, *Senior Member, IEEE*

***Abstract*—In many applications, a reduction of the amount of the original data or a representation of the original data by a small set of variables is often required. Among many techniques, the linear projection is often chosen due to its computational attractiveness and good performance. For applications where real-time performance and flexibility to accommodate new data are required, the linear projection is implemented in field-programmable gate arrays (FPGAs) due to their fine-grain parallelism and reconfigurability properties. Currently, the optimization of such a design is considered as a separate problem from the basis calculation leading to suboptimal solutions. In this paper, we propose a novel approach that couples the calculation of the linear projection basis, the area optimization problem, and the heterogeneity exploration of modern FPGAs. The power of the proposed framework is based on the flexibility to insert information regarding the implementation requirements of the linear basis by assigning a proper prior distribution to the basis matrix. Results from real-life examples on modern FPGA devices demonstrate the effectiveness of our approach, where up to 48% reduction in the required area is achieved compared to the current approach, without any loss in the accuracy or throughput of the design.**

***Index Terms*—Computer vision, factor analysis, field-programmable gate array (FPGA), heterogeneous, linear projection.**

I. Introduction

In many scientific fields, it is required to represent a set of data using a small number of variables. This problem is usually referred to as *dimensionality reduction* or *feature extraction*. Examples can be found in the face-recognition/detection problem [1], [2] where images of people are mapped into a space with fewer dimensions than the original one capturing the main characteristics of the faces, in optical character recognition [3], in image compression [4], and others. In these examples, the objective is to capture the main modes of variation of the original data that have large impact on the overall performance of the application and not to maintain the original information.

An example of dimensionality reduction for a face-recognition application is shown in Fig. 1. The original space of the images has 2000 dimensions. The data are projected to a smaller space with 40 dimensions and then retrieved again in the original space for display purposes. The figure shows that most of the information is well captured in the new space and the faces are well recognized, achieving, at the same time, a 50 times compression.

Fig. 1. Example of projection to a smaller space for a face-recognition application. The top row shows the images in the original space with 2000 dimensions (50×40 pixels), whereas the bottom row shows the images after their projection to a smaller space with 40 dimensions and backprojection again to the original space for display purposes.

The dimensionality-reduction methods are classified into two groups, linear and nonlinear methods, depending on the type of the mapping function to the new space. The main representative of the linear methods is the principal component analysis (PCA) [5] method that maximizes the approximation of the original data under a linear mapping. Factor analysis [6] is similar to PCA, except that it allows the modeling of different noise levels in the dimensions of the original space. In nonlinear methods, the objective is to maximize the approximation of the original data by employing nonlinear mappings. Kernel PCA [7] and principal curves [8] are two representatives of this group.

This paper focuses on the linear projection methods. Let us denote by $x \in R^P$ the original data vector with P elements, by Λ the orthogonal basis of the new space with dimensions $P \times K$, and by $f \in R^K$ the factor vector, which is the result of projecting the original data to the new space. Note that K is usually much smaller than P. Then, the original data can be recovered from the subspace as in

$$x = \Lambda f. \tag{1}$$

Note that the equality holds only in the case where the original data can be perfectly embedded to a smaller space. If this is not the case, the original data cannot be fully retrieved. Many applications require the Λ matrix to have the following property: $\Lambda^{\mathrm{T}}\Lambda = I$, where I denotes the identity matrix. The orthogonality between the columns of Λ implies that the factor vector f corresponding to data x can be calculated by using the same matrix Λ as shown in the following, thus reducing the number of coefficients that need to be stored in the system:

Manuscript received March 10, 2008; revised August 14, 2008, November 23, 2008, and December 18, 2008. This work was supported by the U.K. Research Council under the Basic Technology Research Programme "Reverse Engineering Human Visual Processes" GR/R87642/02.

C.-S. Bouganis and P. Y. K. Cheung are with the Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ London, U.K. (e-mail: christos-savvas.bouganis@imperial.ac.uk).

I. Pournara is with the School of Crystallography, Birkbeck College, University of London, WC1E 7HX London, U.K.

Digital Object Identifier 10.1109/TVLSI.2009.2012510

$$f = \Lambda^{\mathrm{T}} x. \tag{2}$$

In many applications, the large dimensionality of matrix Λ, i.e., 500×40 for face detection applications, and the real-time performance requirements of the algorithm lead to hardware-based solutions. Field-programmable gate arrays (FPGAs) are often used to achieve this goal due to their fine grain parallelism and reconfigurability. This paper addresses the case where maximum performance for the evaluation of (1) or (2)[1] in terms of speed and area is required; thus, only pipelined designs using fixed-point arithmetic are considered.

Several systems can be found in the literature that target an FPGA device and employ a linear projection algorithm, usually PCA, to achieve dimensionality reduction. In [9], Ngo *et al.* propose a face-recognition system for an FPGA that employs a PCA algorithm for dimensionality reduction, whereas in [10], Shams *et al.* propose an FPGA architecture for a face-recognition application that is based on Daubechies wavelets and uses PCA and independent component analysis for dimensionality reduction. In [11], Nguyen *et al.* propose an FPGA architecture for network-intrusion-detection systems. The authors employ PCA to perform dimensionality reduction and to predict malicious connections in a workload.

The problem under consideration is the calculation of an orthogonal basis matrix Λ such that the following are true.

1) The required area for the implementation of (1) or (2)[1] in an FPGA is minimized.
2) An efficient allocation of the heterogeneous components, i.e., embedded multipliers and memory blocks that exist in modern FPGAs, is performed.
3) A specific error in the approximation of the original data that is specified by the user is achieved.

In general, current techniques for mapping the aforementioned process into FPGAs treat the problem as a three-step process. First, the appropriate subspace is calculated in the floating-point domain as the one that provides the best approximation of the data [minimization of the mean-square error (MSE)], resulting in the construction of the basis matrix Λ. Then, the elements of the Λ matrix are quantized for mapping into hardware. In order to optimize the design in terms of the required area, the elements of the basis matrix are often encoded using canonic signed digit (CSD) recoding [12] or subexpression elimination [13], [14].

Finally, the allocation of the available embedded multipliers and memory blocks is performed, usually by assigning the most area-hungry constant coefficient multipliers to the available embedded multipliers of the device or by using a memory block as a lookup table (LUT). The main drawback of the current approach is that the design process for the basis matrix calculation does not take into account the hardware restrictions, leading to suboptimal solutions.

In this paper, we propose a novel framework where the steps of subspace estimation and hardware implementation are considered simultaneously, allowing us to target area-optimized designs that efficiently explore the heterogeneity properties of modern FPGAs. This is achieved by formulating the problem of the subspace calculation in a Bayesian framework, where the cost of the implementation of the necessary components in the Λ matrix is inserted into the system as a prior information. Experiments using real data from computer vision applications demonstrate the effectiveness of the proposed framework.

In summary, this paper presents the following:

1) the coupling of the subspace estimation problem and the area optimization problem for hardware realization under a uniform Bayesian framework;
2) an investigation of how to leverage the dedicated multipliers and embedded RAMs in modern FPGAs to improve the hardware implementations of the subspace estimation problem;
3) the exploration of a family of functions for incorporating the implementation cost of the system into a Bayesian framework.

Part of the work has been published in [15] and [16]. This paper extends the published work by including the efficient allocation of the available blocks during the optimization process, describes in more detail the underlying ideas, provides a more thorough investigation of the performance of the proposed framework when modern FPGA devices are targeted, investigates the accuracy of the high-level area models used in the optimization phase, and finally, focuses the performance evaluation of the proposed framework on examples from real-life applications.

This paper is structured as follows. Section II gives a description of the current work in the hardware design field on the dimensionality-reduction problem and the motivation behind this work. Section III describes the proposed Bayesian factor analysis framework, whereas Section IV discusses how information regarding resource requirements for the hardware implementation of the dimensionality-reduction system is inserted to the Bayesian factor analysis framework. Sections V and VI discuss the heterogeneity exploration of modern FPGAs for the dimensionality-reduction problem. The main assumptions behind the proposed work are discussed in Section VII, whereas Section VIII discusses the high-level area models used in the presented work. Section IX discusses the scalability issues of the proposed framework, whereas Section X provides a summary of the proposed framework. Section XI presents results regarding the performance evaluation of the proposed methodology, whereas Section XII concludes the paper.

II. Background

The problem of dimensionality reduction can be formulated as follows. Given a set of N data $x^i \in R^P$, with $X = [x^1 \quad x^2 \quad \cdots \quad x^N]$, the goal is to find a subspace Λ with dimensions $P \times K$ such that the original data can be expressed as in (3), such that $\|E\|_F$ is minimized, where E denotes the error in the approximation and $\|\cdot\|_F$ denotes the Frobenius norm

$$X = \Lambda F + E. \tag{3}$$

[1]In the case of an orthogonal basis matrix Λ.

[2]In the case of an orthogonal basis matrix Λ.

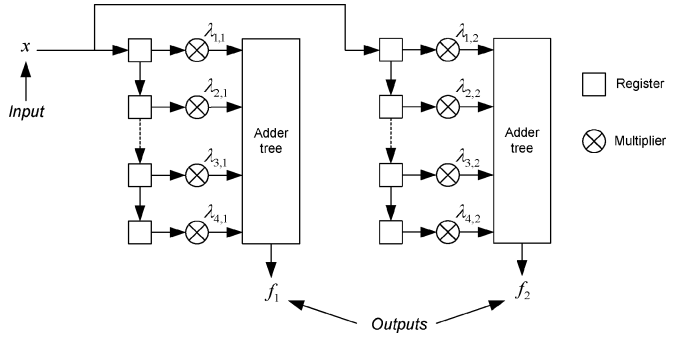

Fig. 2. High-level overview of the system that implements $f = \Lambda^{\mathrm{T}} x$. The system maps the input data from Z^4 space to Z^2 space.

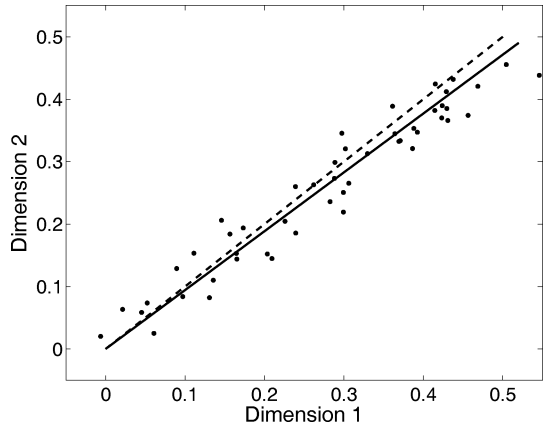

Fig. 3. Projection of 2-D data into 1-D space. The KLT algorithm finds a basis $\Lambda = [0.52\ 0.49]$ that corresponds to the solid line and best explains the data in the floating point domain. However, in a fixed-point domain, the basis $\Lambda = [0.5\ 0.5]$, which corresponds to the dashed line, leads to a design that requires less area than the KLT basis.

The matrix Λ is called *basis matrix* or *factor loadings*, where the matrix F is called *factor matrix* and has dimensions $K \times N$. Out of all possible decompositions, we are seeking the most compact orthogonal matrix Λ, in terms of the number of columns, for a given target mean-square reconstruction error.

The current methodology applies the Karhunen–Loeve transform (KLT) [17], or otherwise known as PCA, for the calculation of the Λ matrix. It has been shown that the KLT produces the most compact representation of the original data, assuming that the error has the same power in all dimensions. In more detail, the KLT transform works as follows. Assuming that the data are centralized, i.e., having zero mean, the columns of the Λ matrix are calculated by iterating between

$$\lambda_i = \arg \max_{\|\lambda_i\|=1} \mathcal{E}\left\{\left(\lambda_i^{\mathrm{T}} X_{i-1}\right)^2\right\} \tag{4}$$

$$X_i = X - \sum_{k=1}^{i-1} \lambda_k \lambda_k^{\mathrm{T}} X \tag{5}$$

where $X = [\,x^1 \quad x^2 \quad \cdots \quad x^N\,]$, $X_0 = X$, and $\mathcal{E}\{\cdot\}$ denotes the expectation.

Note that (5) ensures the orthogonality of the resulting Λ matrix. Having calculated matrix Λ, the coefficients are quantized such that the required approximation for the reconstruction of the data is achieved. Moreover, area-related optimizations can be applied such as CSD recoding [12] and subexpression elimination [13], [14].

A top-level description of the system corresponding to $f = \Lambda^{\mathrm{T}} x$ is shown in Fig. 2. The illustrated system contains two basis vectors, each one with four dimensions. The design produces the projection of the input data from Z^4 space to Z^2 subspace defined by the basis vectors in every clock cycle.

The motivation behind this work is demonstrated by the following example shown in Fig. 3. The 2-D data can be expressed using a 1-D space, achieving a small error in their approximation. The current methodology that is based on the KLT algorithm finds that the best basis to describe the data as $\Lambda = [0.52\,0.49]$, without taking into account the implementation cost associated with such a basis. However, the basis $\Lambda = [0.5\,0.5]$ requires considerably less area by introducing a small error in the data approximation.

Thus, the problem under consideration is to find a basis matrix Λ that produces the best approximation of the original data, in an MSE sense, minimizing, at the same time, the required hardware resources. One possible solution to the problem would be to explore the space of the KLT transformations by optimizing the number of bits that are required for the representation of each element in the KLT basis. The problem can be formulated as in (6), where the entries in the basis matrix Λ and in the factor matrix F take integer values, due to the fixed-point representation in hardware

$$\min_{\Lambda} \sum \sum (\Lambda F - X)^2. \tag{6}$$

This is an ill-conditioned problem because both Λ and F matrices have to be calculated, and thus, further constraints are required. Moreover, it is an integer nonlinear problem; thus, a heuristic method should be applied in order to explore the solution space, which may lead to suboptimal solutions. Also, an extra term should be added that penalizes solutions with large area requirements. Finally, the allocation of the embedded multipliers and memory blocks of modern FPGAs is considered as a separate problem which leads again to suboptimal solutions.

In this paper, we propose a novel framework that is based on a Bayesian formulation of a factor analysis model for dimensionality reduction where the cost of the hardware implementation of the elements in the Λ matrix is minimized. The proposed approach addresses the problem of dimensionality reduction in FPGAs in a unified framework allowing 1) an automatic and more efficient exploration of the design space regarding the approximation of the data versus the cost of the implementation and 2) the efficient allocation of the embedded multipliers and memories in modern FPGAs.

The proposed framework is based on a Bayesian formulation of the factor analysis model instead of the KLT algorithm. This allows the insertion of information regarding the required hardware cost of the implementation of the basis matrix Λ in its calculation process, as it will be demonstrated in the subsequent sections of this paper. Moreover, under the factor analysis model, the error in each dimension of the original space is assumed to be independent from the error in the other dimensions. This provides a larger flexibility than the KLT transform, where

the power of the error is assumed to be the same in all dimensions. Advantage of this feature can be taken in cases where the dimensions of the original space have been corrupted under different noise levels. The proposed methodology can explore the variability of the noise in the original space, leading to the production of efficient designs.

III. Bayesian Factor Analysis Model

Let us assume that we have a random observed vector x with dimensions $P \times 1$. An instance of this vector is denoted as x^i, and we assume that we have N such instances x^i where $i = 1, \ldots, N$. We denote as $f = [\, f_1 \quad \cdots \quad f_K \,]^{\mathrm{T}}$ a vector of K variables that are known as *factors*. f^i denotes an instance of this vector. Note that the number K of factors is always smaller or equal to the number P of observed variables. The factor analysis model states that the observed variables are a linear combination of the factors plus a mean and an error term. For an instance i, that is

$$x^i = \mu + \Lambda f^i + \epsilon^i \tag{7}$$

where $\mu = [\, \mu_1 \quad \cdots \quad \mu_P \,]^{\mathrm{T}}$ and $\epsilon^i = [\, \epsilon_1^i \quad \cdots \quad \epsilon_P^i \,]^{\mathrm{T}}$ are both column vectors of P dimensions with each element corresponding to the mean and the error term of each observed dimension, respectively. The vector μ is the same for all cases i. In our case, we centralize the data, which implies that the mean vector is zero, and thus, it will be discarded in the rest of this paper. Λ is the unobserved basis matrix or sometimes referred to as the *factor loading matrix*. This matrix corresponds to the orthogonal basis matrix of the KLT, but without the orthogonality constraint to be imposed between the columns of the matrix. The factor loading matrix has $P \times K$ dimensions. That is, each column corresponds to a factor, and each row corresponds to an observed variable. The entries of the basis matrix indicate the strength of the dependence of each observed variable on each factor. For example, if λ_{pk} is zero, then variable x_p is independent of factor f_k.

In a matrix form and assuming that the data have been centralized, (7) is written as

$$X = \Lambda F + E \tag{8}$$

where $X = [\, x^1 \quad \cdots \quad x^N \,]$, $F = [\, f^1 \quad \cdots \quad f^N \,]$, and $E = [\, \epsilon^1 \quad \cdots \quad \epsilon^N \,]$.

Factor analysis models assume that the error terms ϵ^i are independent and multivariate normally distributed with mean zero and covariance matrix Ψ as

$$\epsilon^i \sim \mathcal{N}(0, \Psi) \tag{9}$$

where $\Psi = \mathrm{diag}(\psi_1^2, \ldots, \psi_P^2)$.

Thus, the probability distribution of x for each observed case i has a multivariate normal density given by

$$\begin{aligned} p(x^i | f^i, \Lambda, \Psi) &= \mathcal{N}(x^i | \Lambda f^i, \Psi) \\ &= (2\pi)^{-P/2} |\Psi|^{-1/2} \\ &\quad \times \exp\left(-\frac{1}{2} \epsilon^{i\mathrm{T}} \Psi^{-1} \epsilon^i \right) \end{aligned} \tag{10}$$

where $\epsilon^i = x^i - \Lambda f^i$. ϵ^i is the error in the approximation of data x^i by Λf^i. In a matrix notation, the aforementioned equation is written as

$$\begin{aligned} p(X | F, \Lambda, \Psi) &= \mathcal{N}(X | \Lambda F, \Psi) \\ &= (2\pi)^{-N/2} |\Psi|^{-1/2} \\ &\quad \times \exp\left(-\frac{1}{2} \mathrm{tr}[E^{\mathrm{T}} \Psi^{-1} E] \right) \end{aligned} \tag{11}$$

where $E = X - \Lambda F$ and $\mathrm{tr}[\cdot]$ stands for the *trace* operator. In the following subsections, we discuss the prior and posterior probabilities of the parameters F, Λ, and Ψ.

A. Factors

The factors are assumed to be normally distributed with mean zero and covariance matrix Σ_F. That is

$$f^i \sim \mathcal{N}(0, \Sigma_F).$$

The posterior probability of the factors is now derived as

$$p(f^i | x^i, \Lambda, \Psi) \propto p(f^i) p(x^i | f^i, \Lambda, \Psi) = \mathcal{N}\left(f^i | m_F^*, \Sigma_F^* \right) \tag{12}$$

where the posterior mean and variance are given by

$$\begin{aligned} \Sigma_F^* &= (\Sigma_F + \Lambda^{\mathrm{T}} \Psi^{-1} \Lambda)^{-1} \\ m_F^* &= \Sigma_F^* \Lambda^{\mathrm{T}} \Psi^{-1} x^i. \end{aligned}$$

We can now integrate F out of (11) to get the complete density of the data as

$$\begin{aligned} p(X | \Lambda, \Psi) &= \mathcal{N}(X | \Lambda \Sigma_F \Lambda^{\mathrm{T}} + \Psi) \\ &= (2\pi)^{-N/2} |\Lambda \Sigma_F \Lambda^{\mathrm{T}} + \Psi|^{-1/2} \\ &\quad \times \exp\left(-\frac{1}{2} \mathrm{tr}\left[X^{\mathrm{T}} (\Lambda \Sigma_F \Lambda^{\mathrm{T}} + \Psi)^{-1} X \right] \right). \end{aligned} \tag{13}$$

As shown in (13), the complete density of the data is given by a normal distribution with covariance matrix $\Lambda \Sigma_F \Lambda^{\mathrm{T}} + \Psi$. There is a scale identifiability problem associated with Λ and Σ_F. In order to avoid this problem, we can either restrict the columns of Λ to unit vectors or set Σ_F to the identity matrix. The second approach is often used in factor analysis, and it is also adopted here. The impact of this decision to the hardware design is that the dynamic range of the factors is restricted, which is beneficial if the application of interest is required to store or manipulate these parameters.

B. Basis Matrix Λ

The main advantage of the proposed framework lies on the flexibility in selecting the prior distribution of the basis matrix Λ.

We aim to identify a basis matrix Λ that can represent faithfully the data in the high-dimensional space, but, at the same time, provides an optimized hardware design in terms of area usage. The suggested prior is a function of the area that is required for implementing a LUT-based multiplier in an FPGA.

In order to reduce the computational complexity of the algorithm, we assume that the variables in the Λ matrix are independent. This holds when the back-end synthesis tool from the FPGA vendor does not perform any further optimization during the hardware implementation of the derived matrix Λ in the place-and-route stage. This assumption allows us to express the probability distribution of the Λ matrix as the product of the probabilities of the individual elements as in

$$p(\Lambda) = \prod_{p=1}^{P} \prod_{k=1}^{K} p(\lambda_{pk}). \tag{14}$$

In the current work, we select the prior probability distribution $p(\lambda_{pk})$ to be inversely proportional to the area that is required for the realization of a multiplication by λ_{pk} using LUTs. The function $g(\cdot)$ relates the area $A(\lambda_{pk})$ required by the constant coefficient multiplier to the prior probability distribution as in (15). The selection of the function g is discussed in Section IV.

$$p(\lambda_{pk}) = g\left(A(\lambda_{pk})\right). \tag{15}$$

The posterior probability of each element λ_{pk} of Λ is given by

$$p(\lambda_{pk}|X, F, \Psi) \propto p(X|F, \Lambda, \Psi) \prod_{p=1}^{P} \prod_{k=1}^{K} p(\lambda_{pk}). \tag{16}$$

The aforementioned distribution does not have a known form; thus, we have to calculate (16) for all possible values of λ_{pk} and then use the uniform probability distribution to sample the new value of λ_{pk}.

C. Noise Covariance Matrix Ψ

A convenient conjugate prior is assigned to the inverse of the noise covariance matrix Ψ so that its posterior distribution has a known form. Thus, the prior on each ψ_p^{-2} is a Gamma distribution given by

$$\begin{aligned} p\left(\psi_p^{-2}|\alpha_\Psi, \beta_\Psi\right) &= \mathcal{G}\left(\psi_p^{-2}|\alpha_\Psi, \beta_\Psi\right) \\ &\propto \left(\psi_p^{-2}\right)^{\alpha_\Psi - 1} \exp\left(-\psi_p^{-2}\beta_\Psi\right) \end{aligned}$$

where α_Ψ and β_Ψ are the shape and scale parameters of the Gamma distribution, respectively.

The Gamma posterior distribution of ψ^{-2} is given by

$$\begin{aligned} p\left(\psi_p^{-2}|X, F, \Lambda\right) &\propto p\left(\psi_p^{-2}|\alpha_\Psi, \beta_\Psi\right) p(X|F, \Lambda, \Psi) \\ &= \mathcal{G}\left(\psi_p^{-2}|\alpha_\Psi + \frac{1}{2}P, \beta_\Psi + \frac{1}{2}S_{pp}\right) \end{aligned}$$

where

$$S_{pp} = \sum_{i=1}^{N} \sum_{p=1}^{P} \left(x_p^i - \sum_{k=1}^{K} \lambda_{pk} f_k^i\right)^2. \tag{17}$$

In the current work, we suggest to use a common variance ψ^{-2} for all dimensions P; however, the model can be extended to allow the estimation of different variance ψ_p^{-2} in each dimension p.

D. Orthogonality

The aforementioned statistical framework does not necessarily produce an orthogonal basis of the new space. However, in the computer vision field, which is our targeted domain, this condition is often required. Under the proposed framework, this requirement is enforced by finding first the direction that mostly explains the data, which is the direction with the maximum variance, and then projecting the data to the obtained space and retrieving them back. The direction with the maximum variance in the original space is the one that contains data points that deviate the most from the average value, and thus, it is the direction that should be explained first since the data along this direction exhibit the worst approximation by using the average value. This process is repeated in order to calculate each vector in the new space. The advantage of this approach is twofold. First, it produces an orthogonal basis that describes the new space, and second, it inserts the error due to the quantization of the data in the hardware implementation back to the remaining data. By doing that, the next vector of the new subspace minimizes the error due to quantization of the basis matrix Λ and factors F, as well as explaining the data. In our earlier work [18], [19], we have proposed a similar methodology for 2-D filter design exploration.

Summarizing, Section III demonstrates how we can express the estimation of matrix Λ under a Bayesian framework. The framework treats the Λ matrix as a random matrix and constructs a probability density function for it. Due to the complexity of the distribution, the final matrix Λ is estimated using sampling techniques. The Bayesian framework allows prior information regarding instances of Λ to be inserted, steering the posterior distribution to certain modes. This prior distribution allows us to penalize instances of the Λ matrix that require many hardware resources by assigning low probabilities to these realizations, such that we favor realizations of Λ that do not require many hardware resources.

IV. Mapping Implementation Cost to Prior Distribution

The prior distribution for the Λ matrix has to be a valid distribution, that is: $p(\lambda_{pk}) \geq 0$, $\forall \lambda_{pk}$ and $\sum_{\lambda_{pk}} p(\lambda_{pk}) = 1$. Thus, we are seeking functions that map the space of the area cost to the space of valid distributions. These functions should be monotonically decreasing and nonnegative and should sum to one. In the rest of this paper, we use the family of functions shown in (18) to map the area required by a constant coefficient multiplier to a valid distribution

$$g\left(A(\lambda_{pk})\right) = c\left(A(\lambda_{pk})\right)^{-a}, \quad a, c > 0. \tag{18}$$

c is a constant and ensures that $\sum_{\lambda_{pk}} g(A(\lambda_{pk})) = 1$. Fig. 4 shows possible mappings of the original cost distribution to valid distributions. The figure shows that constant coefficient multipliers that use less area are assigned to have large probabilities, whereas constant coefficient multipliers that require large area have small probabilities. The smaller the value a becomes, the more uniform the distribution gets, with $a = 0$ resulting to a noninformative prior to the system. This means that all coefficients can be selected with the same probability, implying that the respective constant coefficient multipliers require the same

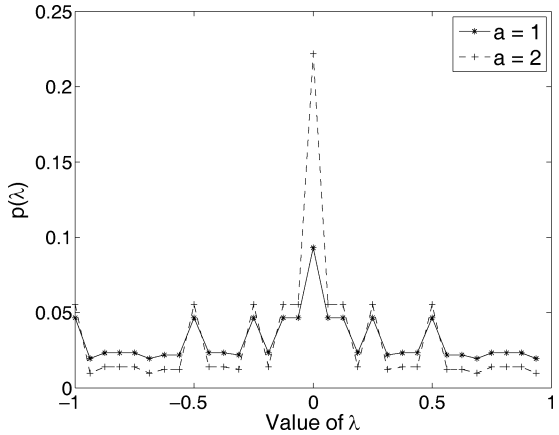

Fig. 4. Mapping of area information to a probability distribution for $a = 1$ and $a = 2$ [see (18)].

hardware resources. In this case, the proposed framework resembles the KLT algorithm.

A. LUT-Based Multiplier Cost Model

It should be noted that the proposed framework is independent of the encoding scheme for the multipliers. In the current work, the two's complement representation for the coefficients and the CSD recoding are used. The corresponding cost for the implementation in the first case is calculated using the COREGEN tool from Xilinx [20] in order to use accurate resource usage information for the constant coefficient multiplier, but estimation models can also be used. Other encoding schemes can be easily accommodated by the framework. The only information needed by the proposed framework is the area that is required for each constant coefficient multiplier.

V. Allocation of the Embedded Multipliers

The previous section considered the problem of creating the prior distribution of matrix Λ when all coefficients are mapped to logic-based multipliers. In this section, the efficient allocation of the embedded multipliers in modern FPGAs for the problem of dimensionality reduction is also targeted.

A possible solution to the problem can be the allocation of the embedded multipliers to the most area-hungry multiplications. However, this would lead to suboptimum solutions since the calculation of the basis matrix Λ does not take into account the extra precision that an embedded multiplier provides.

Instead, a method is proposed that couples the calculation of the Λ matrix with the efficient allocation of the available embedded multipliers. This is achieved by the introduction of an indicator matrix Z, which indicates the coefficients of the Λ matrix that are mapped to the embedded multipliers. The indicator matrix Z has the same dimensions as the Λ matrix, where each element z_{pk} can take only two values. A $z_{pk} = 1$ indicates that the λ_{pk} coefficient is mapped to an embedded multiplier, where $z_{pk} = 0$ indicates a mapping to a multiplier that is implemented through reconfigurable logic (LUTs). The possible values of the indicator matrix Z are constrained by the fact that the number of entries z_{pk} that have a value of one should be equal to the number of the embedded multipliers that are available to the user.

The posterior probability of each element λ_{pk} of Λ(16) is now augmented by the indicator matrix Z as in

$$p(\lambda_{pk}, z_{pk}|X, F, \Psi) \propto p(X|F, \Lambda, \Psi, Z) \prod_{p=1}^{P} \prod_{k=1}^{K} p(\lambda_{pk}, z_{pk}). \tag{19}$$

The previous distribution does not have a known form; thus, we have to calculate (19) for all possible values of λ_{pk} and z_{pk}. However, due to the large number of possible combinations of λ_{pk} and z_{pk}, this is prohibited. The approach that has been adopted in this paper is to sample the indicator variable z_{pk} through a uniform distribution, imposing, at the same time, the constraint $\sum z_{pk} = N_{\text{em}}$, where N_{em} is the number of the available embedded multipliers. It should also be noted that the likelihood of the data $p(X|F, \Lambda, \Psi, Z)$ has been now augmented by the indicator matrix Z.

The prior probability distribution $p(\lambda_{pk}, z_{pk})$ has two forms depending on the value of the indicator variable z_{pk}; $p(\lambda_{pk}, z_{pk} = 1)$ has a uniform distribution in the range of values that are allowed by the precision imposed by the embedded multipliers in the target device, whereas $p(\lambda_{pk}, z_{pk} = 0)$ follows the same distribution as in (18), since the coefficient $p(\lambda_{pk})$ is mapped to reconfigurable logic.

In the case where an orthogonal basis for the new space is required, an alternation of the aforementioned algorithm for the allocation of the embedded multipliers is required, since at each iteration, only one vector of the new basis is calculated (see Section III-D). In the proposed framework, the approach that has been adopted is to preallocate a fraction of the total number of available multipliers to each vector (dimension) of the new space. Note that the embedded multipliers are not preallocated to a specific coefficient of matrix Λ but are assigned to be allocated to one of the coefficients of a row of matrix Λ during the optimization phase. This preallocation is based on the assumption that a more important dimension, in terms of its impact to the output error, should be allocated with a larger number of embedded multipliers than a less important dimension. The estimation of the importance of each dimension is performed by applying the KLT and using the eigenvalues, after a proper normalization, as a measurement of the importance of each dimension. It should be noted that the quantization effects and the complexity for each vector in the calculated basis matrix Λ have not been taken into account in the preallocation of the embedded multipliers; thus, it is expected that the framework will produce a suboptimal design. However, this preallocation is necessary for not increasing the computational complexity of the algorithm.

VI. Allocation of the Embedded Memories

Modern FPGA devices contain a large number of embedded memory blocks. For the problem under consideration, the available memory blocks can be used as LUTs of different widths that realize the multiplication of input data with a constant coefficient. The proposed framework optimizes the allocation of the available memory blocks during the calculation of the basis

matrix Λ. This is achieved by allowing the entries in the indication matrix Z to take three values, indicating a LUT-based multiplier, a multiplication using an embedded multiplier, and a multiplication using a memory block.

Thus, the prior probability distribution $p(\lambda_{pk}, z_{pk})$ now takes three forms depending on the value of the indicator variable z_{pk}; $p(\lambda_{pk}, z_{pk} = 0)$ and $p(\lambda_{pk}, z_{pk} = 1)$ are the same as before, whereas $p(\lambda_{pk}, z_{pk} = 2)$ has a uniform distribution in the range of values that are allowed by the precision imposed by the width of the available memory blocks in the target device.

In the case where an orthogonal basis for the new space is required, a similar preallocation technique to the embedded multiplier case is followed.

VII. Summary of Assumptions

This section summarizes the main assumptions that are used in the presented work. These are the following.

1) Normal prior distribution of the factors f. This assumption allows the posterior distribution to be of known form, making the sampling of these variables simple.
2) Independence of the Λ matrix elements. This assumption allows the decomposition of the joint probability function as the product of the probability of each element in the basis matrix Λ. This permits the sampling of each element of the Λ matrix independently from the others. Thus, the computational complexity of the sampling process is a linear function of the number of elements in the matrix and not an exponential that would be otherwise.
3) Gamma distribution for the noise covariance Ψ. This allows the posterior distribution of Ψ to be of a known form and restricts the values of the samples to be always positive.
4) Uniform sampling of the indicator matrix Z. With the introduction of the indicator matrix, the complexity of sampling from $p(\lambda_{pk}, z_{pk})$ increases exponentially, since all the possible instances of the indicator matrix have to be constructed. This prohibits the application of the proposed framework to large problems. Thus, it is necessary to consider only a subset of the instances of the indicator matrix in each iteration of the algorithm.
5) Preallocation of the embedded multipliers and memory blocks. This step is necessary when an orthogonal basis matrix Λ is targeted. The preallocation is based on the KLT algorithm which provides a high-level estimate of the importance of each dimension regarding its impact to the final error at the output of the system. This may produce suboptimal designs, since the quantization effects are not taken into account during this process, but it provides our best estimate for the preallocation of the embedded multipliers and memory blocks, and it is necessary in order to couple the allocation of these blocks with the calculation of the basis matrix Λ.

VIII. Area Models

The aim of the framework is to calculate a basis matrix Λ that achieves a certain error in the data approximation and, at the same time, minimizes the required resources. A cost model has been constructed for a Xilinx Virtex-II FPGA to predict the cost of the different components that are used by the framework [19].

The current high-level model predicts the resource usage within 3% of the actual cost when the component is synthesized and placed and routed on the device [21]. However, when the whole design is placed and routed, the error between the predicted resource usage and the actual one increases. This is due to further optimizations that are applied by the back-end tool, which is out of the scope of the used high-level model. It should be noted that, in the current work, the cost of the adder trees and the cost of the constant coefficient multipliers using CSD are estimated, where the cost of the constant coefficient multipliers using a two's complement representation is calculated using the COREGEN tool from Xilinx [20].

A more detailed evaluation of the accuracy of the area models in the proposed framework for two different devices is performed in Section XI.

IX. Scalability

The proposed framework utilizes a Gibbs sampling algorithm [22] in order to draw samples from the posterior distribution of the variables. The initial few samples do not correspond to the true posterior distribution and are discarded. This period is called the *burn-in period*. After that point, the samples are kept, and the final values are estimated. The prior and posterior distributions of the noise covariance matrix Ψ and of the factors f have well-known expressions and are easy to sample from, and their complexity scales linearly with the problem size. Due to the discrete nature of the coefficients in the multipliers, the prior and posterior distributions of λ_{pk} are discrete and do not map to a distribution of a known form. This implies that, in each iteration, the posterior distribution of Λ has to be calculated for every discrete value of each λ_{pk}. However, the complexity of the system scales linearly to the number of constant coefficient multipliers that are available to the design, making the proposed approach applicable in real-life scenarios.

In the case where an efficient allocation of the embedded multipliers and memory blocks is also targeted, the complexity of the calculation of the prior and posterior distributions of λ_{pk} would scale exponentially with respect to the number of embedded blocks. However, a suboptimum solution is targeted by uniformly sampling the indicator matrix Z. Thus, the proposed framework is still applicable to real-life problems.

X. Summary of the Proposed Framework

The proposed Bayesian formulation for dimensionality reduction gives the flexibility of inserting any prior knowledge regarding the resource requirements of the system under consideration through the use of prior distributions. The proposed framework explores this feature by inserting *a priori* information in basis matrix Λ regarding the hardware-related cost for the implementation of the required constant coefficient multipliers using LUTs and the number of available embedded blocks. Thus, the Bayesian model aims to find a basis matrix that represents faithfully the data, while, at the same time, information about the implementation cost of the required multipliers is taken into account. The objective is to reduce the hardware resources required for the implementation of the

Algorithm: Bayesian factor analysis for K factors
Set $X_0 = X$, where X denotes the original centralized data.
Set $F_0 = [\ ]$.
Initialize Λ_0.
FOR $k = 1 : K$
  Calculate vector λ_k (Figure 6)
  Calculate the factors using
  $f_k = (\lambda_k^T \lambda_k)^{-1} \lambda_k^T X_{k-1}$
  and quantize them to the user's specific number of bits, $f_k \leftarrow quant(f_k)$.
  Set $X_k = X - \sum_{j=1}^{k} \lambda_j f_j$.
  Set $\Lambda_k = [\Lambda_{k-1} \lambda_k]$ and $F_k = [F_{k-1} f_k]$.
END

Fig. 5. Algorithm for Bayesian factor analysis for K factors.

Algorithm: Calculate vector λ_k
FOR $iter = 1 : maxIter$
  For each data x^i, sample f^i from $p(f^i|x^i, \Lambda, \Psi)$
  Set $F = [f_1\ f_2\ \ldots\ f_N]$
  For each element of matrix Λ, sample λ_{pk} from $p(\Lambda|X, F, \Psi)$.
  Sample Ψ from $p(\psi_p^{-2}|X, F, \Lambda)$
  If $iter >$ *burn-in period*
    Store Λ
  endif
END
return most_often_elements(Λ)
most_often_elements() operator is applied
component-wise

Fig. 6. Algorithm for calculating vector λ_k.

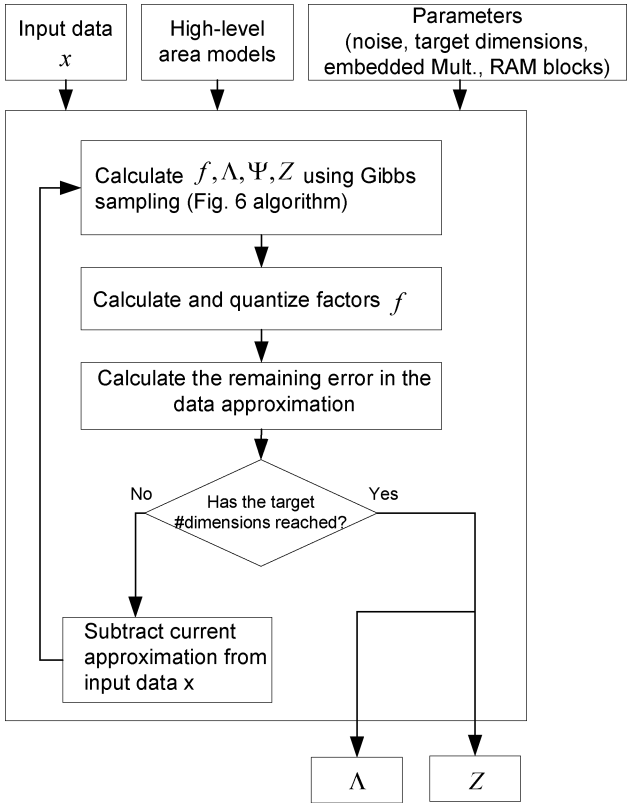

Fig. 7. High-level block diagram of the proposed framework.

linear projection stages, maximizing, at the same time, the data approximation. The proposed framework is shown in Figs. 5 and 6. It should be noted that the calculation of the factors is performed using (20), which provides a solution that minimizes the MSE of the approximation. λ_k denotes the kth column of the Λ matrix

$$f_k = \left(\lambda_k^{\mathrm{T}} \lambda_k\right)^{-1} \lambda_k^{\mathrm{T}} X_{k-1}. \tag{20}$$

Due to the orthogonality requirements, it should hold that $\Lambda^{\mathrm{T}}\Lambda = I$, where I denotes the identity matrix. However, due to quantization error, this is not the case, and the factors should be calculated using (20) to achieve an optimum performance. The calculation of the vector λ_k that best explains the data is outlined in Fig. 6.

In the case where an efficient allocation of the embedded multipliers and memory blocks is also targeted, the algorithm in Fig. 6 is altered. Just before the point of sampling the elements of Λ matrix, the indicator matrix Z is sampled, which indicates the mapping of the coefficients λ_{pk} to LUTs, embedded multipliers, and memory blocks. The code is omitted from the figure for reasons of clarity.

The proposed algorithm returns the coefficients of the basis matrix Λ including information regarding their actual mapping into the hardware, i.e., LUT-based multiplier, embedded multiplier, or memory block, that best approximate the original data.

Fig. 7 shows a high-level overview of the proposed framework, where its main steps are depicted. The inputs of the framework are the set of data X for approximation, the high-level area models that provide an estimate of the area requirements for the implementation of a constant coefficient multiplier, the power of the noise in the data, the target dimensionality of the reduced space, and the number of embedded multipliers and block RAMs that are available. The output of the framework is the basis matrix Λ and the indicator matrix Z. As the figure shows, the framework estimates the basis matrix Λ, the noise covariance Ψ, the indicator matrix Z, and the factors f using Gibbs sampling. The details of this step are shown in Fig. 6. In the next step, the algorithm calculates the factors f that best approximate the original data X given the current basis matrix Λ using (20), and the obtained values are quantized for hardware mapping.

When the orthogonality constraint is imposed, the aforementioned process is repeated for each dimension of the target subspace until the number of targeted dimensions has been reached. In this case, the remaining error in the data approximation is calculated, and if the number of target dimensions has not been reached, the current approximation of the data is subtracted from the original data set X, and the process is repeated using as input the remainder of the approximation.

In the case where such orthogonality constraint is not imposed, all the columns (dimensions) of the basis matrix Λ are calculated in one pass of the algorithm.

XI. Performance Evaluation

The proposed framework is evaluated under two different scenarios. The first scenario focuses on the evaluation of the framework when the objective is to find a basis of a new subspace having the number of dimensions of the subspace fixed. In the second scenario, the aforementioned constraint is lifted, and the proposed algorithm searches to find the best basis that describes the original data X with the minimum MSE, without fixing the

number of dimensions of the new space. In both cases, the target is to minimize the design's implementation cost and to allocate efficiently the embedded multipliers and memory blocks, if any, of the device. Face recognition/detection [2], [1], optical character recognition [3], and image compression [4] are a few of the applications where the aforementioned two problems are encountered.

We have compared our proposed framework with the currently available approach that is based on the KLT transform and subsequent quantization of the Λ and F matrices. In addition, the reference algorithm has been extended to search the space of possible bases by varying the wordlength of the elements in the Λ matrix and imposing a common wordlength for all the elements. The factors F are quantized to 8 bits in both the proposed framework and the reference algorithm. In all the cases, it is assumed that the input data has an 8-bit wordlength, which is common for image-processing applications. The error in the final approximation is calculated by projecting the data back to the original space after having calculated and quantized the factors F. In the cases where embedded multipliers and memory blocks are used in the design, their precision, i.e., wordlength of the multiplier and width of the memory block, are reported.

It should be noted that both algorithms use CSD recoding for the implementation of the constant coefficient multipliers. This enhances the performance of the reference algorithm, making it "fixed-point" aware. Moreover, both algorithms are based on the same hardware architecture (see Fig. 2), producing one result per clock cycle. The algorithms differ in the way that they estimate the basis matrix Λ. The proposed algorithm takes into account the required resources for the realization of the multipliers during the calculation of the basis matrix Λ, where the reference algorithm does not.

A. Area Model Evaluation

The proposed framework applies high-level area models of the required multipliers and adders in order to provide a fast estimation of the required resources for a given design. The accuracy of these models for the case of CSD recoding representation is evaluated by targeting two Xilinx devices. The focus of this section is on the high-level area models of the CSD recoding rather than on the COREGEN generated multipliers, since the area usage is not known in the former case. It should be noted that there is an uncertainty of the estimated area usage even for the latter case, due to the area usage estimation of the required adder trees. These devices are the Xilinx Virtex II 6000 with speed grade 6 and the Xilinx Virtex4LX80 with speed grade 12.

Due to the structure of the hardware system, the evaluation of the total area estimation can be broken down to the evaluation of the area estimation of each hardware block that performs the projection to one dimension of the new space. Due to the large number of possible configurations that this hardware block can have, the objective is to sample instances of possible configurations of this block in order to investigate the accuracy of the proposed high-level area modeling. An example that maps a 6-D space to spaces with dimensions that vary between one and eight is considered. It should be noted that any example case can be used as long as a sufficient sampling of

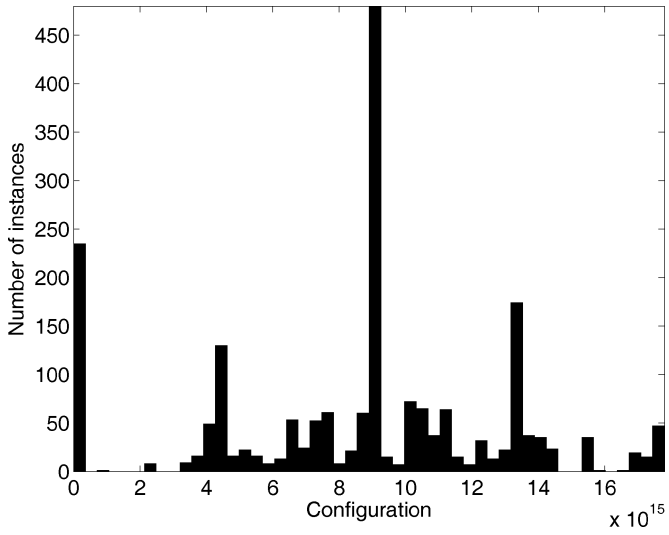

Fig. 8. Histogram of the generated hardware block configurations used for testing the accuracy of the high-level resource usage model.

the possible configurations of the hardware block that implements the projection is achieved. The generated designs that perform the aforementioned projection and are used for the evaluation of the area models have between one and eight dimensions, where the number of bits allocated to each of the coefficients of the constant coefficient multipliers vary between one and eight. In total, 448 designs were synthesized and placed and routed. Fig. 8 shows the histogram of the number of generated hardware blocks used for the linear projection. In order to be able to visualize this histogram, each 6-D configuration of the hardware block has been mapped to a scalar using a one-to-one mapping function, i.e., each possible configuration of the 6-D space is mapped to a point on a line segment, and vice versa; each point on the line segment is mapped to a single point in the 6-D space. The figure demonstrates that the 448 used designs provide hardware block configurations that cover most of the space of possible configurations.

Fig. 9 shows a graph of the predicted resource utilization (in slices) and the actual resource utilization after placing and routing the designs in the Xilinx Virtex II device using ISE 9.1 from Xilinx. It can be seen that there is a linear relationship between the predicted cost and the actual cost reported by the Xilinx tools. The high value of 0.9801 of the correlation coefficient [23] between the estimated and actual resource usage supports the validity of the used high-level area models. The p-value [23], for testing the hypothesis of no correlation, is close to zero, which further supports the validity of the predicted results. The same experiment is performed again but targeting the Xilinx Virtex 4 device. The graph of the predicted resource utilization and the actual resource utilization when the Virtex 4 device is targeted is similar to that in Fig. 9. Again, the results have a strong correlation with a value of 0.9786 and a p-value of almost zero. The aforementioned results demonstrate that the high-level area models used in the proposed framework provide highly accurate predictions of the actual resource usage. It should be noted that there is a trend of the high-level area model to overestimate the required resources as the size of the

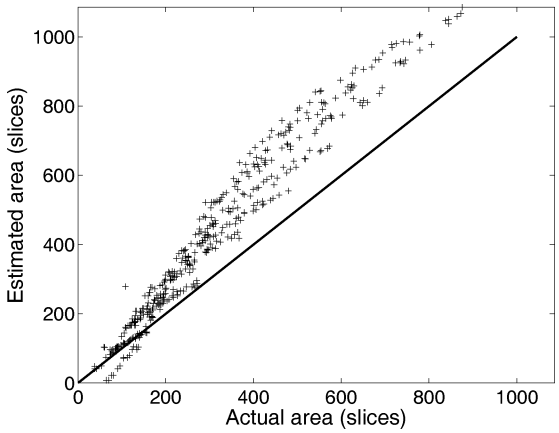

Fig. 9. Predicted resource utilization (in slices) and actual resource utilization after placing and routing the designs in a Xilinx Virtex II device. The solid line corresponds to the function $y = x$.

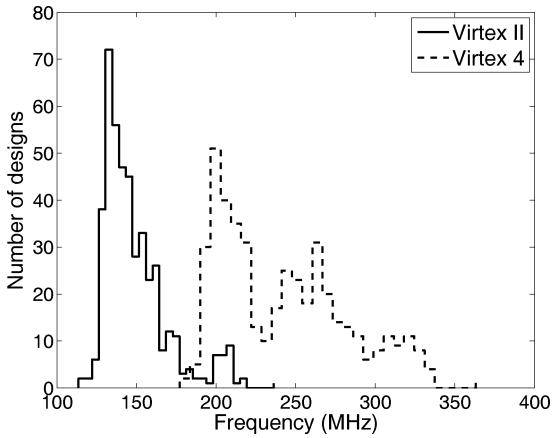

Fig. 10. Histogram of the achieved maximum frequency for the Xilinx Virtex II and Virtex 4 devices.

design increases. This is expected as the back-end tools perform low-level optimizations to the design that are not modeled in the high-level model.

Moreover, the two graphs have very small differences, indicating that the area usage of the specific architecture used in this framework (see Fig. 2) is not affected by the architectural differences of these two devices.

Furthermore, the aforementioned set of designs is used to investigate the achieved clock frequency after the place-and-route stage. Fig. 10 shows a histogram of the achieved maximum frequency of the designs when the two devices are targeted. When the Xilinx Virtex II device is targeted, the figure shows that the maximum frequency varies between 115 and 240 MHz. The reason for this variation lies on the varying complexity of the constant coefficient multipliers in the designs. When the Xilinx Virtex 4 device is targeted, the maximum achieved frequency varies between 175 and 360 MHz. The improved results are only due to the use of a different device. A closer investigation of the obtained results shows that the distribution ordering is almost the same across the two devices but not identical. This is expected due to undeterministic results of place-and-route tools. It can be concluded that the proposed framework produces, in all the cases, highly optimized designs. It should be clarified that the large variation on the achieved frequency of the aforementioned designs shown in Fig. 10 is because the plotted frequencies are from designs that target a range of MSE approximations and resource usages, which imply a large variation of the wordlength of the used multipliers.

B. Dimensionality Reduction Targeting a Specific Number of Dimensions

First, the proposed framework is tested for its performance when the number of dimensions of the target space is fixed. This scenario can arise in applications where the number of dimensions of the factors has to be restricted, e.g., image compression [4]. From the hardware perspective, this can also be enforced due to the available memory bandwidth in the system where the factors F are stored.

Fig. 11 shows the performance of the proposed Bayesian framework and the reference algorithm for mapping data from the R^3 space to Z^2 subspace. The data belong to R^2 space and have been embedded to the R^3 space by adding Gaussian noise. Note that, even if the data are inherently 2-D, projecting them to Z^2 does not imply that the approximation error will be zero. This is due to the quantization process of the basis matrix Λ and to the error that has been added to the data. The figure shows the estimated number of slices required using the high-level area models for the implementation of the system as a function of the achieved MSE approximation of the data. The staircase like shape of the plots is due to the discrete nature of the design space. It should be noted that the few number of design points in this figure and in the subsequent figures is due to the use of Pareto curves when the design points are plotted. Please note that it is not always necessary for the Pareto front to be well covered. For the problem of interest, i.e., mapping linear projection designs to hardware, it is of interest to find a design that minimizes the MSE in the data approximation, given the designer's resource budget, and not to be able to produce designs for all possible levels of MSE. However, a well-covered front would provide a designer more flexibility to select a design as close as possible to his/her resource budget. The proposed framework can be adjusted to increase the coverage of the Pareto front by using a more refined set of the parameter values as the number of bits that are available for number presentation in the constant coefficient multipliers and the values of the a parameter in (18) that maps the area requirements to a probability distribution during the search for the basis matrix Λ. However, due to the discrete nature of the design space and the nature of the proposed framework, a well-covered Pareto curve cannot be guaranteed. The same drawback also holds for the existing techniques. The results demonstrate that the proposed framework can reduce the required area by up to a factor of two, achieving, at the same time, the same MSE in the data approximation with the reference algorithm. It should be noted that the reduction in the area that is achieved by applying the proposed framework depends on the data to be approximated. An upper bound on the area reduction factor cannot be calculated analytically. However, given that enough iterations are performed in the Gibbs sampling, the proposed methodology will never perform worse than the

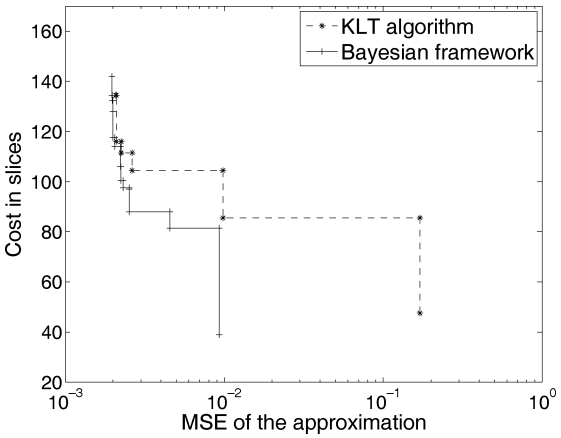

Fig. 11. Required area for mapping data from R^3 space to Z^2 space versus the MSE of the data approximation.

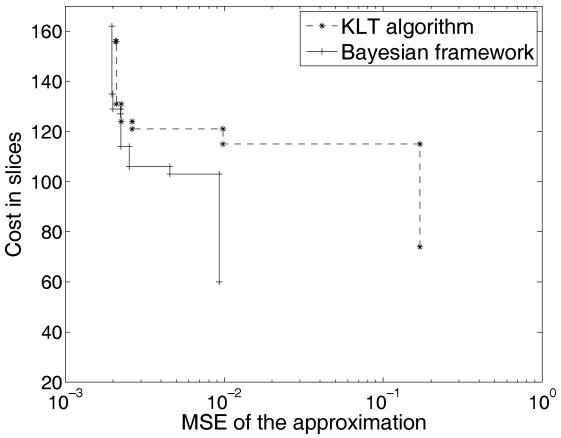

Fig. 12. Required area for mapping data from R^3 space to Z^2 space versus the MSE of the data approximation. The graph presents placed and routed designs using the Xilinx tools.

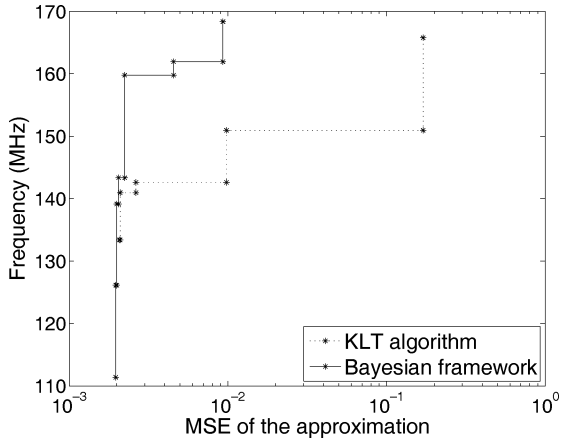

Fig. 13. Achieved clock frequency of designs mapping data from R^3 space to Z^2 space versus the MSE of the data approximation. The results are acquired using placed and routed designs using the Xilinx tools.

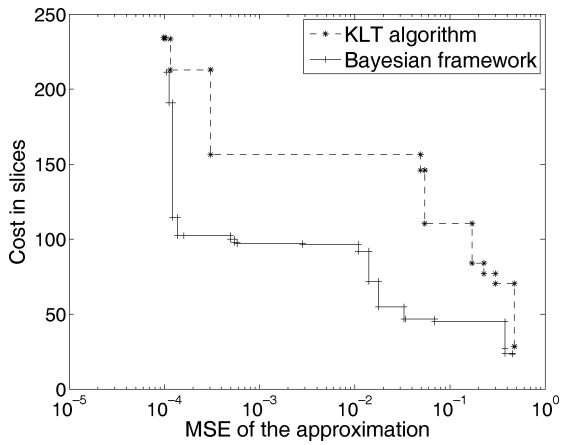

Fig. 14. Required area for mapping data from R^4 space to Z^2 space versus the MSE of the data approximation. Two embedded multipliers and two memory blocks are available for allocation.

reference algorithm. Fig. 12 shows the same results, but now, the acquired designs have been placed and routed, targeting a Xilinx Virtex-II device using the Xilinx tools. The figure shows that the achieved gain in the area remains almost the same with the predicted gain using the high-level area models. Comparing these results with Fig. 11, there is a small shift of the plots, but the general shape of the plots remains the same.

Fig. 13 shows the achieved clock frequency of the aforementioned designs as it has been reported by the Xilinx tools. The figure shows that the proposed framework produces designs that not only require less resources than the designs produced by the current methodology achieving the same MSE approximation of the original data but also can be clocked with a higher frequency. This is mainly attributed to the fact that the produced designs employ less complex constant coefficient multipliers, coefficients with a fewer number of nonzero bits than the reference algorithm, allowing the designs to be clocked with a higher frequency clock than the corresponding designs produced using the reference algorithm for the same MSE. It should be noted that no embedded multipliers or memory blocks were used.

Fig. 14 shows the performance of the proposed framework in the case of mapping data from the R^4 space to Z^2 subspace where two embedded multipliers and two memory blocks are available for allocation. The framework explores the design space and estimates a projection basis that takes advantage of the availability of the embedded blocks and the arithmetic precision that each one offers. In this case, a precision of 10 bits (coefficient wordlength) is assumed for both embedded blocks. In the reference algorithm, the embedded blocks are allocated to the coefficients that introduce the largest error in the approximation when they are quantized to a specific number of bits. The figure demonstrates that the proposed framework outperforms the existing methodology across the whole range of targeted MSE approximation.

The proposed framework has also been evaluated using data from a real application. Fig. 15 shows the results obtained by the proposed framework and the reference algorithm for a face-recognition application. In this experiment, the YALE Face Database B is used [24]. The original space is Z^{500} and is mapped to a Z^{40} space. In all the cases, the proposed framework outperforms the reference algorithm, achieving designs that require less area and, at the same time, have the same MSE error in the data approximation. Fig. 16 shows the percentage gain in slices for various target values of the error in the original data approximation. The figure shows that the gain

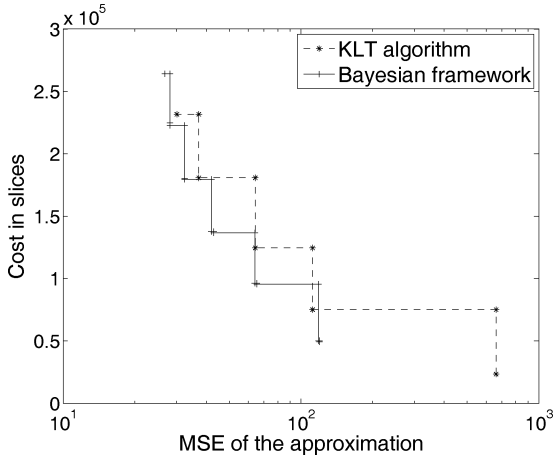

Fig. 15. Required area for a face-recognition application versus the MSE of the data approximation. The algorithm maps the data from Z^{500} space to Z^{40} space.

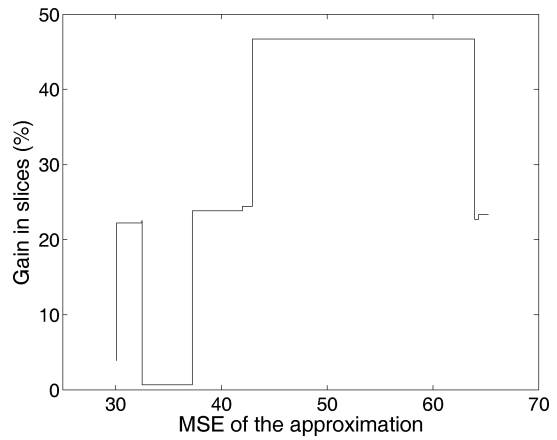

Fig. 16. Percentage gain in the area (in slices) for various values of the target MSE of the data approximation versus the required area for a face-recognition application.

in slices can reach up to 48% for a given range of the acceptable approximation error. The area comparison has been performed considering designs from both methodologies that achieve the same MSE at the output of the system.

Moreover, Fig. 17 shows the approximation of a set of images from the face database when 30 dimensions are used to represent the original data. The first column depicts the original images. The second column corresponds to the reconstruction of the same data using a design derived by the reference algorithm, whereas the third column corresponds to data reconstruction using a design derived by the proposed framework. The image quality is almost the same between the second and the third column; however, the proposed framework produces a design that requires 70% of the area of the design that is derived using the reference algorithm. Fig. 18 shows the approximation of a set of images when similar size designs from the reference algorithm and the proposed algorithm are employed. It is clear that the design that is derived using the proposed algorithm produces a considerably better approximation of the data than the design produced by the reference algorithm.

Fig. 17. First column corresponds to the original images. The second column corresponds to the reconstruction of the same data using a design derived by the reference algorithm (KLT and fixed point), whereas the third column corresponds to data reconstruction using a design derived by the proposed framework. The proposed framework produces a design that requires 70% of the area of the design that is derived using the reference algorithm, achieving, at the same time, a similar level of approximation of the data.

Fig. 18. Approximation of the original data when a similar size in area designs are employed by (second column) the reference algorithm (KLT and fixed point) and (third column) the proposed algorithm. The results demonstrate the superior data approximation achieved by the proposed algorithm compared with that of the reference algorithm, where the used designs have similar sizes.

Fig. 19 shows the achieved MSE in the approximation of each person in the database for the reference algorithm (KLT using fixed point) and the proposed framework. The proposed framework achieves similar quality results using only 70% of the area of the reference design. Fig. 20 shows the achieved MSE in the approximation of each person in the database for the case where designs from the reference algorithm and the proposed framework with similar area requirements are used. It is clear from the figure that the proposed framework produces designs that achieve better quality results than the designs produced by the reference algorithm, having, at the same time, the same area requirements.

The proposed framework has been also evaluated for an object detection application using the CODID–CVAP Object Detection Image Database [25]. In this experiment, seven images of a ball are used for capturing its appearance under different illumination conditions. The original space is Z^{2500} and is mapped to a Z^3 space. Fig. 21 shows two of the original images (first column) and the results obtained by the reference algorithm (second column) and the proposed framework (third column) after projecting the images back to the original space. The selected designs for comparison achieve very similar average MSEs: 67 for the reference algorithm and 76 for the proposed framework. However, the proposed algorithm provides a design that requires 81% of the area of the design that is derived using the reference algorithm.

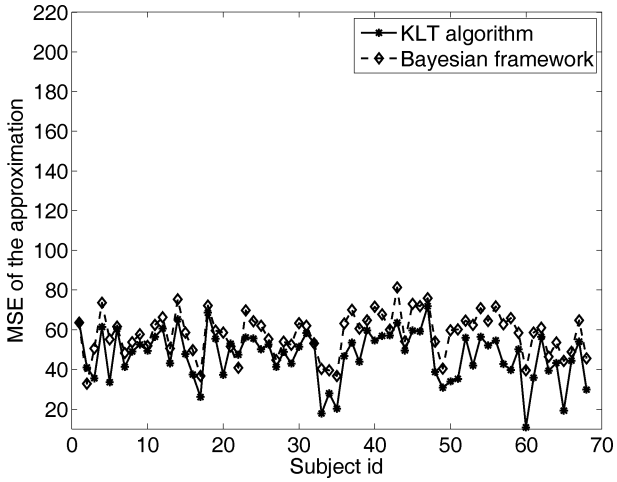

Fig. 19. MSE error approximation across all the subjects in the database. The produced design by the Bayesian framework requires 70% of the area of the design that is derived by the reference algorithm.

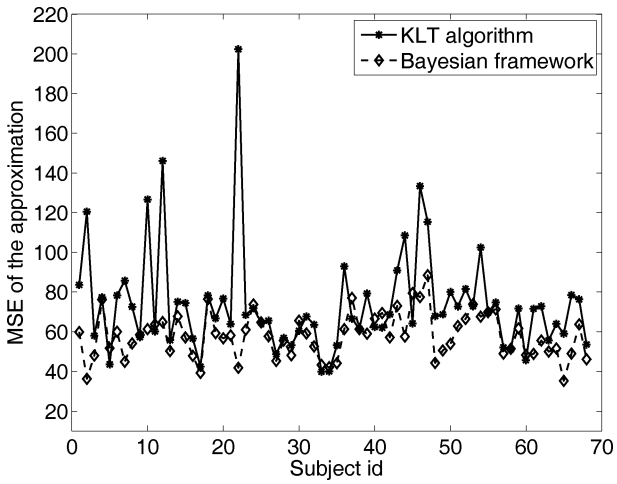

Fig. 20. MSE error approximation across all the subjects in the database. The produced design by the Bayesian framework and the design that is derived by the reference algorithm require the same area.

Fig. 21. Approximation of the original data using designs derived by (second column) the reference algorithm (KLT and fixed point) and (third column) the proposed algorithm. The proposed framework produces a design that requires 81% of the area of the design that is derived using the reference algorithm, achieving, at the same time, a similar level of approximation of the data.

C. Dimensionality Reduction

The proposed framework is evaluated also for a general reduction problem where there are no constraints about the number of dimensions in the new space. Face recognition/detection [1], [2] and optical character recognition [3] are few of the applications where there is no constraint about the number of dimensions in the new space.

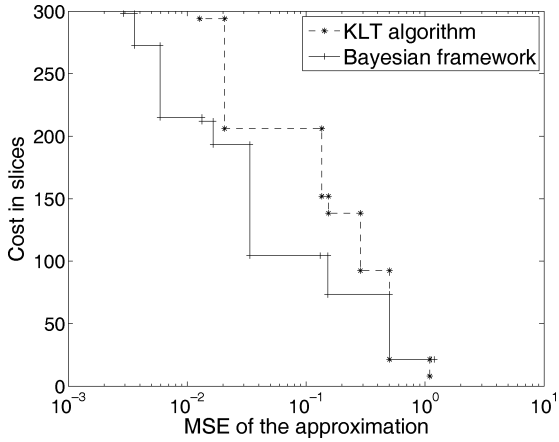

Fig. 22. Required area for data that lie into R^6 space versus the MSE of the data approximation. No constraint on the dimensionality of the new space is imposed.

Fig. 22 shows the MSE approximation versus the required area for data that belong to R^6, when the proposed framework and the reference algorithm are not constrained by the dimensionality of the new space. The proposed approach outperforms the reference algorithm, producing designs that require up to half the hardware resources than the designs produced by the reference algorithm, achieving, at the same time, the same mean-square approximation error.

D. Run-Time Investigation

The scalability of the proposed framework is discussed in Section IX. Here, results regarding the run-time requirements of the proposed framework are presented. Fig. 23 shows the required time for the proposed framework as a function of the number of original dimensions (P) and the dimensions of the targeted space (K), when coefficients are represented by 8 bits. A PC using an Intel Core 2 CPU running at 1.86 MHz with 2 GB of RAM was used, and the programming environment was MATLAB. The figure shows that the required time scales linearly with these variables. Using these data, a linear function that provides an estimate of the required run time is fitted

$$Time\ (\text{in seconds}) = 72P + 294K - 1612. \qquad (21)$$

Furthermore, experiments have shown that the run time scales exponentially with the number of bits used for representing the coefficients of the constant coefficient multipliers. The run time for the reference algorithm is always in the range of seconds. However, it should be noted that the process of finding the basis matrix Λ is usually performed only once in the design cycle, and it may be repeated when new data become available. Thus, considering the potential area savings of the proposed framework, the extra required run time that is needed for the proposed framework compared to the reference algorithm is not of importance. Moreover, the run time scalability of the proposed framework allows its application to real-life problems as has already been demonstrated.

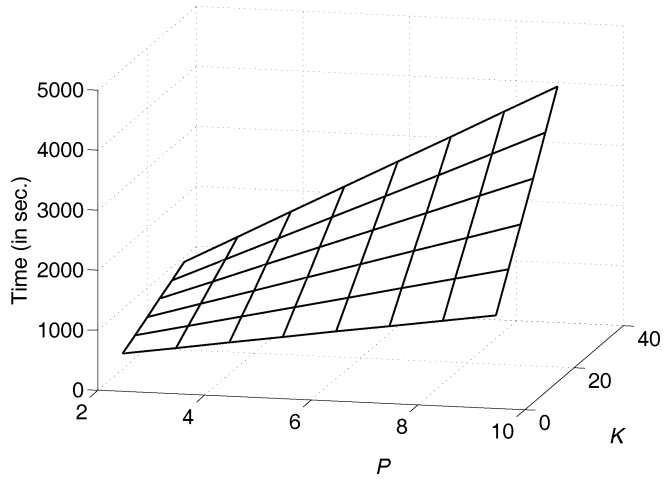

Fig. 23. Required run time of the proposed framework as a function of the number of dimensions (P) in the original space and the dimensions of the targeted space (K). The coefficients are represented by 8 bits.

XII. Conclusion

This paper proposes a novel Bayesian factor analysis framework for dimensionality-reduction implementation designs in FPGAs. The proposed approach couples the problem of data approximation using a small set of variables, the problem of area design optimization, and the problem of the heterogeneity exploration of modern FPGAs under a unified framework. It has been demonstrated that, by injecting information to the system regarding the resource requirements for the implementation of the constant coefficient multipliers using a prior distribution, we are able to target designs that have a significant reduction in the resource usage when they are compared against current techniques, achieving, at the same time, the same MSE in the approximation of the data. Future work will involve the extension of the framework to exploit the heterogeneous components of more recent FPGA devices, e.g., Stratix III from Altera,[3] where the heterogeneous components can be also configured, adding an extra dimension to the problem of mapping dimensionality-reduction designs into FPGAs.

References

[1] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 72–86, 1991.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[3] H.-C. Kim, D. Kim, and S. Y. Bang, "A numeral character recognition using the PCA mixture model," *Pattern Recog. Lett.*, vol. 23, no. 1-3, pp. 103–111, Jan. 2002.

[4] J. Taur and C. Tao, "Medical image compression using principal component analysis," in *Proc. Int. Conf. Image Process.*, 1996, vol. 2, pp. 903–906.

[5] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philos. Mag.*, vol. 2, no. 6, pp. 559–572, 1901.

[6] C. E. Spearman, "'General intelligence' objectively determined and measured," *Amer. J. Psychol.*, vol. 15, pp. 201–293, 1904.

[7] B. Schölkopf, A. Smola, and K.-R. Muller, *Advances in Kernel Methods—Support Vector Learning*. Cambridge, MA: MIT Press, 1999.

[8] T. Hastie and W. Stuetzle, "Principal curves," *J. Amer. Stat. Assoc.*, vol. 84, no. 406, pp. 502–516, 1989.

[9] H. Ngo, R. Gottumukkal, and V. Asari, "A flexible and efficient hardware architecture for real-time face recognition based on eigenface," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, May 2005, pp. 280–281.

[10] N. Shams, I. Hosseini, M. Sadri, and E. Azarnasab, "Low cost FPGA-based highly accurate face recognition system using combined wavelets with subspace methods," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 2077–2080.

[11] D. T. Nguyen, G. Memik, and A. Choudhary, "A reconfigurable architecture for network intrusion detection using principal component analysis," in *Proc. ACM/SIGDA 14th Int. Symp. FPGA*, New York, 2006, p. 235.

[12] I. Koren, *Computer Arithmetic Algorithms*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[13] A. Dempster and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 42, no. 9, pp. 569–577, Sep. 1995.

[14] R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, and D. Durackova, "A new algorithm for elimination of common subexpressions," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 1, pp. 58–68, Jan. 1999.

[15] C.-S. Bouganis, I. Pournara, and P. Y. K. Cheung, "A statistical framework for dimensionality reduction implementation in FPGAs," in *Proc. IEEE Int. Conf. Field Program. Technol.*, 2006, pp. 365–368.

[16] C.-S. Bouganis, I. Pournara, and P. Y. K. Cheung, "Efficient mapping of dimensionality reduction designs onto heterogeneous FPGAs," in *Proc. IEEE Symp. Field-Program. Custom Comput. Mach.*, 2007, pp. 141–150.

[17] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Edu. Psychol.*, vol. 24, pp. 417–441, 1933.

[18] C.-S. Bouganis, G. A. Constantinides, and P. Y. K. Cheung, "A novel 2D filter design methodology for heterogeneous devices," in *Proc. Field-Program. Custom Comput. Mach.*, 2005, pp. 13–22.

[19] C.-S. Bouganis, P. Y. K. Cheung, and G. A. Constantinides, "Heterogeneity exploration for multiple 2D filter designs," in *Proc. Field Program. Logic Appl.*, 2005, pp. 263–268.

[20] [Online]. Available: http://www.xilinx.com

[21] C.-S. Bouganis, S.-B. Park, G. A. Constantinides, and P. Y. K. Cheung, "Synthesis and optimization of 2D filter designs for heterogeneous FPGAs," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 1, no. 4, p. 24, Jan. 2008.

[22] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-12, no. 6, pp. 609–628, Nov. 1984.

[23] W. M. I. Dennis, D. Wackerly, and R. L. Scheaffer, *Mathematical Statistics with Applications*. Pacific Grove, MA: Duxbury, 2002.

[24] P. Belhumeur and D. Kriegman, "What is the set of images of an object under all possible lighting conditions?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 1996, pp. 270–277.

[25] S. Ekvall, "CODID – CVAP object detection image data base," 2008. [Online]. Available: http://www.nada.kth.se/~ekvall/codid.html

[3] [Online]. Available: http://www.altera.com

Christos-S. Bouganis (S'01–M'04) received the M.Eng. (Honors) degree in computer engineering and informatics from University of Patras, Patras, Greece, and the M.Sc. and Ph.D. degrees from Imperial College London, London, U.K., in 1998, 1999, and 2004, respectively.

In 2007, he joined the faculty at Imperial College London. His research interests include reconfigurable architectures for signal processing, computer vision and machine learning algorithms targeting reconfigurable hardware.

Iosifina Pournara received the Ph.D. degree from Birkbeck College, University of London, London, U.K., in 2004.

Her research interests include analysis of high throughput life science data using machine learning algorithms and acceleration of machine learning algorithms using FPGAs.

Peter Y. K. Cheung (M'85–SM'04) received the B.S. degree with first class honors from Imperial College of Science and Technology, University of London, London, U.K., in 1973.

Since 1980, he has been with the Department of Electrical Electronic Engineering, Imperial College, where he is currently a Professor of digital systems and head of the department. His research interests include VLSI architectures for signal processing, asynchronous systems, reconfigurable computing using FPGAs, and architectural synthesis.