Efficient Mapping of Dimensionality Reduction Designs onto Heterogeneous FPGAs

Christos-S. Bouganis [#1], Iosifina Pournara [*2], Peter Y.K. Cheung [#3]

[#] *Department of Electrical and Electronic Engineering*
Imperial College London
South Kensington campus, London SW7 2AZ, UK
[1] ccb98@imperial.ac.uk
[3] p.cheung@imperial.ac.uk

[*] *School of Crystallography*
Birkbeck College, University of London
Malet Street, London WC1E 7HX, UK
[2] i.pournara@cryst.bbk.ac.uk

Abstract

Dimensionality reduction or feature extraction has been widely used in applications that require to reduce the amount of original data, like in image compression, or to represent the original data by a small set of variables that capture the main modes of data variation, as in face recognition and detection applications. A linear projection is often chosen due to its computational attractiveness. The calculation of the linear basis that best explains the data is usually addressed using the Karhunen-Loeve Transform (KLT). Moreover, for applications where real-time performance and flexibility to accommodate new data are required, the linear projection is implemented in FPGAs due to their fine-grain parallelism and reconfigurability properties. Currently, the optimization of such a design, in terms of area usage and efficient allocation of the embedded multipliers that exist in modern FPGAs, is considered as a separate problem to the basis calculation. 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 under a probabilistic Bayesian framework. 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. Results using real-life examples demonstrate the effectiveness of our approach.

1 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 as *dimensionality reduction* or *feature extraction*. Examples can be found in the face recognition/detection problem [11, 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 [7], in image compression [10], and others.

An example of dimensionality reduction for a face recognition application is illustrated in Figure 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 demonstrates that most of the information is well captured in the new space, the faces are well recognized, achieving at the same time a 50 times compression.

The dimensionality reduction is often achieved by the linear projection of the original data to a smaller space than the original. The basis of the new space is constructed using a set of data targeting a certain error in the data approximation. The data can be recovered from the subspace as in (1), where $x \in Z^P$ denotes the original data vector with P elements, Λ denotes an orthogonal basis with dimensions $P \times K$, and $f \in Z^K$ denotes the factors vector. Note that K is usually much smaller than P.

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

Many applications require the Λ matrix to be orthogonal,

0-7695-2940-2/07 $25.00 © 2007 IEEE
DOI 10.1109/FCCM.2007.50

Figure 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, where the bottom row shows the images after their projection to a smaller space with 40 dimensions, and back-projection again to the original space for display purposes.

that is $\Lambda^T\Lambda = I$ and $\Lambda^{-1} = \Lambda^T$, where I denotes the identity matrix. The orthogonality of Λ implies that the factors vector f corresponding to data x can be calculated by using the same matrix Λ as shown in (2), thus reducing the amount of coefficients that need to be stored in the system.

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

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. FPGAs are often used to achieve this goal due to their fine grain parallelism and reconfigurability. The current work addresses the case where maximum performance for the evaluation of (1) in terms of speed is required, thus only pipelined designs are considered.

The problem under consideration is the calculation of an orthogonal basis matrix Λ such that the required area for implementation of (1) in an FPGA is minimized, an efficient allocation of the heterogeneous components, i.e. embedded multipliers that exist in modern FPGAs, is performed, and a specific error in the approximation of the original data that is specified by the user is achieved.

Current techniques for mapping the above process into FPGAs treat the problem as a three step process. Firstly, the appropriate subspace is calculated in the floating-point domain as the one that provides the best approximation of the data. 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 are often encoded using Canonic Signed Digit recoding [8] or subexpression elimination [5, 9]. Finally, the allocation of the available embedded multipliers is performed, usually by assigning the area hungry constant coefficient multipliers to the embedded multipliers of the device. The main drawback of the current approach is that the design process for the basis 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 efficient 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 Λ matrix is inserted to the system as a prior information. Experiments using real data from computer vision applications demonstrate the effectiveness of the proposed framework.

In summary, the original contributions of this paper are:

- the combination of the subspace estimation problem and the area optimization problem for hardware realization under a uniform Bayesian framework,
- the exploration of the heterogeneity properties of modern FPGAs for the subspace estimation problem,
- the exploration of a family of functions for incorporating the implementation cost of the system into a Bayesian framework.

The paper is structured as follows. Section 2 gives a description of current work in the field. Section 3 describes the proposed Bayesian factor analysis framework, where Section 4 discusses the requirements of the functions that map the area implementation information to a probability distribution. Section 5 discusses the heterogeneity exploration of modern FPGAs for the subspace estimation problem. Section 7 discusses the scalability issues of the proposed framework, where Section 8 provides a summary of the framework. Finally, Section 9 presents the evaluation results of the proposed algorithm, where Section 10 concludes the paper.

2 Background

The problem of dimensionality reduction can be formulated as follows. Given a set of N data $x^i \in R^P$, with $X = [x^1, x^2, \cdots, x^N]$, 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 $\mathcal{E}\{E^2\}$ is minimized, where E denotes the error in the approximation and $\mathcal{E}\{\cdot\}$ denotes the mean operator.

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

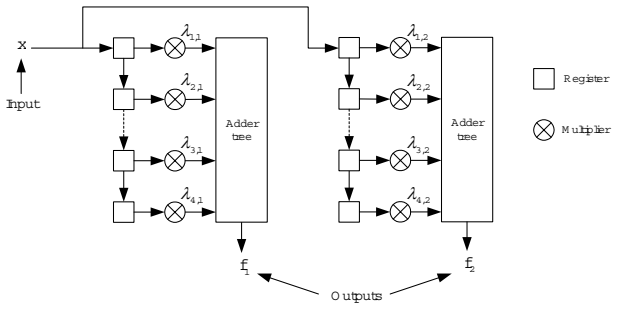

Figure 2. A high level overview of the system that implements $f = \Lambda^T x$. The system maps the input data from Z^4 space to Z^2 space.

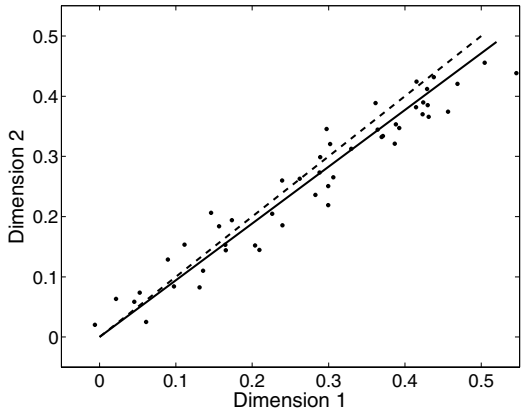

Figure 3. Projection of 2D data into 1D space. The KLT algorithm finds a basis $\Lambda = [0.52, 0.49]$ that corresponds to the solid line and best explains the data in the floating point domain. However, in a fixed-point domain, the basis $\Lambda = [0.5, 0.5]$, which corresponds to the dashed line,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

Current methodology applies the Karhunen-Loeve Transform (KLT) [6] for the calculation of the Λ matrix. It has been shown that the KLT transform produces the most compact representation of the original data assuming that the error has the same power in all dimensions. In more details, the KLT transform works as follows. Assuming that the data are centralized, that is having zero mean, the columns of the Λ matrix are calculated by iterating between equations (4) and (5).

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

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

where $X = [x^1, x^2, \cdots, x^N]$ and $X_0 = X$.

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 Canonic Signed Digit recoding [8] and subexpression elimination [5, 9].

A top-level description of the system corresponding to $f = \Lambda^T x$ is illustrated in Figure 2. The illustrated system contains two basis vectors, each one with dimension four. 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 depicted in Figure 3. The two dimensional data can be expressed using a one-dimension space achieving small error in their approximation. The current methodology that is based on the KLT algorithm, finds that the best basis to describe the data is $\Lambda = [0.52; 0.49]$, without taking into account the implementation cost associated with such a basis. However, the basis $\Lambda = [0.5; 0.5]$ requires considerable less area and at the same time achieves a smaller error in the approximation of the original data than the required.

One possible solution to the problem under consideration 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 non-linear problem, thus a heuristic method should be applied in order to explore the solution space, which may lead to suboptimal solutions. Finally, the allocation of the embedded multipliers 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 im-

plementation 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 (a) a better exploration of the design space regarding the approximation of the data versus the cost of the implementation, and (b) the efficient allocation of the embedded multipliers in modern FPGAs. Moreover, 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.

3 Bayesian Factor Analysis Model

Let's 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 $f = (f_1, \ldots, f_K)'$ a vector of K variables, known as *factors*, where 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 $\mu = (\mu_1, \ldots, \mu_P)'$ and $\epsilon^i = (\epsilon^i_1, \ldots, \epsilon^i_P)'$ 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 it will be discarded in the rest of the paper. Λ is the unobserved basis matrix or more often referred to as the *factor loadings matrix*. The factor loadings 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 factor loadings 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, (7) is written as

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

where $X = (x^1, \ldots, x^N)$, $F = (f^1, \ldots, f^N)$, and $E = (\epsilon^1, \ldots, \epsilon^N)$.

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

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

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

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

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

In a matrix notation, the above equation is written as

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

where $tr[\cdot]$ stands for *trace* operator. In the following subsections, we discuss the prior and posterior probabilities of the parameters F, Λ and Ψ.

3.1 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

$$p(f^i|x^i, \Lambda, \Psi) \propto p(f^i)p(x^i|f^i, \Lambda, \Psi) = \mathcal{N}(f^i|m^*_F, \Sigma^*_F) \tag{12}$$

where the posterior mean and variance are given by

$$\begin{aligned} \Sigma^*_F &= (\Sigma_F + \Lambda'\Psi^{-1}\Lambda)^{-1} \\ m^*_F &= \Sigma^*_F\Lambda'\Psi^{-1}x^i \end{aligned}$$

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

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

As shown in (13), the complete density of the data is given by a normal distribution with covariance matrix $\Lambda\Sigma_F\Lambda' + \Psi$. 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.

3.2 Factor loadings matrix Λ

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

We aim to identify a factor loadings matrix that can represent faithfully the data in the high dimension 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 synthesis tool does not perform any further optimization in the derived matrix Λ during the mapping process. 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 (14).

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

In the current work, we select the prior probability distribution $p(\lambda_{pk})$ to be inverse 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 4.

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

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 above 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}.

3.3 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

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

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

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

where

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

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.

3.4 Orthogonality

The above statistical framework does not necessarily produces an orthogonal basis of the new space. However, in computer vision field, which is our target domain, this condition is often required. Under the proposed framework, this requirement is enforced by finding first the direction that mostly explains the data, that is the direction with the maximum variance, and then by projecting the data to the obtained space and retrieving them back. This process is repeated in order to calculate each vector in the new space. The advantage of this approach is twofold. Firstly, it produces an orthogonal basis that describes the new space, and secondly it inserts the error due to the quantization of the data in the hardware implementation back to the remaining space. By doing that, the next vector in the new sub-space minimizes the error due to quantization of the factor loadings Λ and factors F, as well as explaining the data. Bouganis *et al.* [4, 3] have proposed a similar methodology for 2D filter design exploration.

4 Mapping implementation cost to prior distribution

The prior distribution for the Λ matrix has to be a valid distribution, that is:

$$p(\lambda_{pk}) \geq 0 \ ,\forall \lambda_{pk} \tag{18}$$

and

$$\sum_{\lambda_{pk}} p(\lambda_{pk}) = 1 \tag{19}$$

Thus, we are seeking for functions that map the space of the area cost to the space of valid distributions. These functions should be:

- monotonically decreasing
- no negative

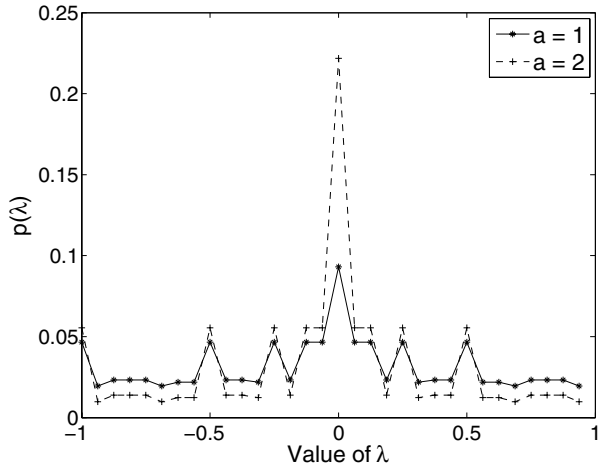

Figure 4. Mapping of area information to a probability distribution for $a = 1$ and $a = 2$ (see (20)).

- sum to one

In the rest of the paper we use the family of functions shown in (20) to map the area required by a constant coefficient multiplier to a valid distribution.

$$g(A(\lambda_{pk})) = c\,(A(\lambda_{pk}))^{-a} \quad , \quad a, c > 0 \qquad (20)$$

,where c is a constant and ensures that $\sum_{\lambda_{pk}} g(A(\lambda_{pk})) = 1$. Figure 4 demonstrates possible mappings of the original cost distribution to valid distributions. The figure shows that constant coefficients multipliers that use small area are assigned with large probability, where constant coefficient multipliers that require large area, have small probability. The smaller the value a becomes, the more uniform the distribution gets, with $a = 0$ resulting to a non-informative prior to the system. In this case, the proposed framework resembles the KLT algorithm.

4.1 LUT based multipliers cost model

It should be noted that the proposed framework is independent of the encoding scheme for the multipliers. In the current work we use a two's complement representation for the coefficients. The corresponding cost for the implementation is calculated using the COREGEN tool from Xilinx [1]. Other encoding schemes like the Canonic Signed Digit recoding 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.

5 Allocation of the embedded multipliers

In this section, the efficient allocation of the embedded multipliers in modern FPGAs for the problem of dimensionality reduction is targeted. 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 value 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 (21).

$$p(\lambda_{pk}, z_{pk}|X, F, \Psi) \propto p(X|F, \Lambda, \Psi, Z) \prod_{p=1}^{P} \prod_{k=1}^{K} p(\lambda_{pk}, z_{pk}) \qquad (21)$$

The above distribution does not have a known form, thus we have to calculate (21) for all possible values of λ_{pk} and z_{pk}. However, due to the large number of possible combination of λ_{pk} and z_{pk} this is prohibited. The approach that has been adopted in this paper is to sample the indication variable z_{pk} through a uniform distribution, imposing at the same time the constraint $\sum z_{pk} = N_{em}$, where N_{em} is the number of the available embedded multipliers. It should be also 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, where $p(\lambda_{pk}, z_{pk} = 0)$ follows the same distribution as (20), since the coefficient $p(\lambda_{pk})$ is mapped to reconfigurable logic.

6 Area models

The aim of the framework is to calculate a factor loadings matrix Λ that achieves a certain error in the data approximation and at the same time to minimize 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 [3].

The current high-level model predicts the resource usage within 3% of the actual cost when the component is synthesized and placed on the device. 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 only the cost of the adder trees are estimated, since the cost of the constant coefficient multipliers are given by the COREGEN tool from Xilinx [1].

7 Scalability

The proposed framework utilizes a Gibbs sampling algorithm 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 *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, are easy to sample from them, 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 known form distribution. 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 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 is also targeted, the complexity of the calculation of the prior and posterior distributions of λ_{pk} scales exponentially with respect to the number of coefficients. However, a sub-optimum solution is found by uniformly sampling the indicator matrix Z. Thus, the proposed framework is still applicable to real-life problems.

8 Summary of the proposed framework

The proposed Bayesian formulation for dimensionality reduction gives the flexibility of inserting any prior knowledge regarding the system under consideration through the use of prior distributions. The proposed framework explores this feature by inserting a priori information in Λ regarding the hardware related cost for the implementation of the required constant coefficient multipliers. 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 proposed framework is summarized in Figures 5 and 6. It should be noted that the calculation of the factors is performed using (22), which provides a solution

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 $X_k = X - \sum_{j=1}^{i} \lambda_j f_j$.
  Set $\Lambda_k = [\Lambda_{k-1} \lambda_k]$ and $F_k = [F_{k-1} f_k]$.
END

Figure 5. Algorithm for Bayesian factor analysis for K factors

that minimizes the mean square error of the approximation. λ_k denotes the k^{th} column of the Λ matrix.

$$f_k = (\lambda_k^T \lambda_k)^{-1} \lambda_k^T X_{k-1} \tag{22}$$

Due to the orthogonality requirements it should hold that $\Lambda^T \Lambda = I$, where I denotes the identity matrix. However, due to quantization error this is not the case, and the factors should be calculated using (22), to achieve an optimum performance. The calculation of the vector λ_k that best explains the data is outlined in Figure 6.

In the case where an efficient allocation of the embedded multipliers is also targeted, the algorithm in Figure 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 embedded multipliers. The code is omitted from the figure for reasons of clarity.

9 Performance Evaluation

The proposed framework is evaluated under two different scenarios. The first scenario focuses on the evaluation of the framework when the target is to find a basis of a new subspace having the number of dimensions of the subspace fixed. In the second scenario, the above constraint is lifted, and the proposed algorithm searches to find the best basis that describes the original data X with the minimum mean square error. In both cases, the target is to minimize the design's implementation cost and to efficient allocate the embedded multipliers, if any, of the device. Face recognition/detection, optical character recognition, and image

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 $F = [f_1, f_2, \cdots, f_N]$
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

Figure 6. Algorithm for calculating vector λ_k

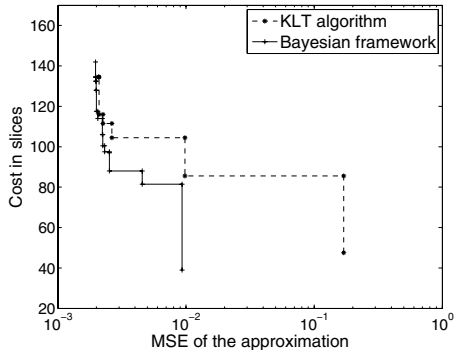

Figure 7. Required area for mapping data from R^3 space to Z^2 space versus the mean square error of the data approximation.

compression are few of the applications where the above two problems are encountered.

We have compared our proposed framework with the current available approach which 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 basis 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 have 8 bits 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 are used in the design, their precision, *i.e.* wordlength of the multiplier, is reported.

9.1 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 [10]. From the hardware perspective, this can also be enforced due to the available memory bandwidth in the system where the factors F are stored.

Figure 7 illustrates 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 error. Note that even the data are inherent two dimensional, 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 plots the achieved mean square error approximation of the data as a function of the number of slices that are required for the implementation of the system. The staircase like shape of the plots is due to the discrete nature of the design space. The proposed framework can reduce the required area up to a factor of two, achieving at the same time the same mean square error 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. Figure 8 illustrates 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 depicts 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 Figure 7 there is a small shift of the plots, but the general shape of the plots remains the same.

Figure 9 demonstrates the performance of the proposed framework and the current methodology in the case of mapping data from the R^3 space to Z^2 subspace where one embedded multiplier is available for allocation. The framework explores the design space and estimates a projection basis that takes advantage of the availability of an embedded multiplier and the arithmetic precision that it offers. In this case, a precision of 12 bits (input wordlength) is assumed for the embedded multiplier. In the current methodology, the embedded multiplier is allocated to the coefficient that introduces the largest error in the approximation when it is quantized to a specific number of bits. The figure demonstrates that in the range of an acceptable approximation, that is design with less than 0.1 mean square error, the proposed framework outperforms the existing methodology.

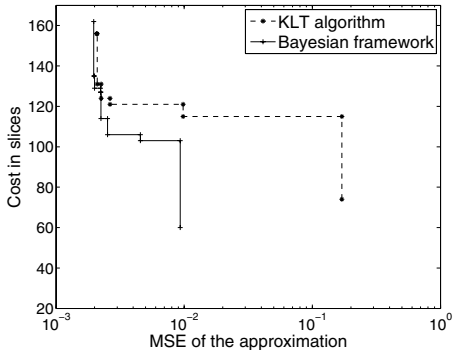

Figure 8. Required area for mapping data from R^3 space to Z^2 space versus the mean square error of the data approximation. The graph presents placed and routed designs using the Xilinx tools.

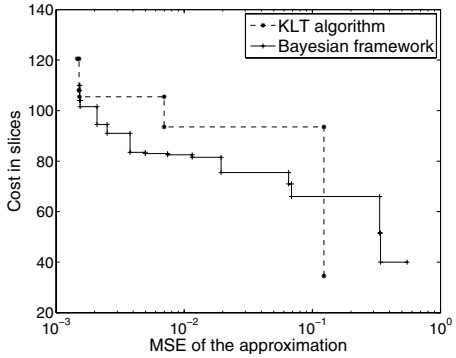

Figure 9. Required area for mapping data from R^3 space to Z^2 space versus the mean square error of the data approximation. One embedded multiplier is available for allocation.

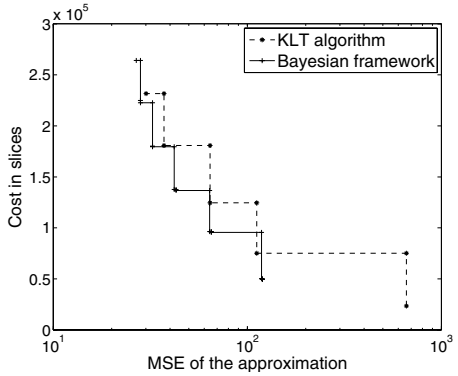

Figure 10. Required area for a face recognition application versus the mean square error of the data approximation. The algorithm maps the data from Z^{500} space to Z^{40} space.

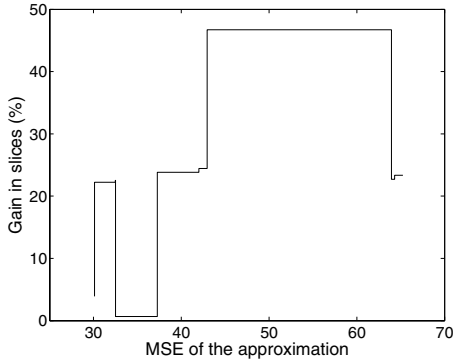

Figure 11. Percentage gain in the area (slices) for various values of the target mean square erroror of the data approximation versus the required area for a face recognition application.

The proposed framework has also been evaluated using data from a real application. Figure 10 illustrates the results obtained by the proposed framework and the reference algorithm for a face recognition application. 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 error in the data approximation. Figure 11 illustrates the percentage gain in slices for various target values of the error in the original data approximation. The figure shows that the gain in slices can reach up to 48% for a given range of the acceptable approximation error.

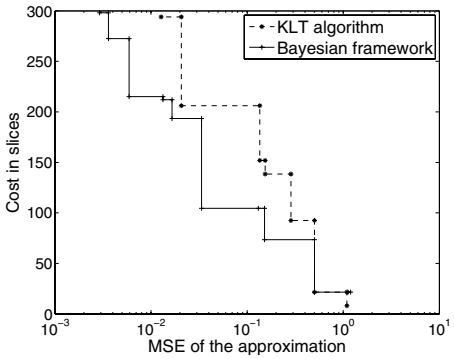

Figure 12. Required area for data lie into R^6 space versus the mean square error of the data approximation. No constrain to the number of vectors for the new space is imposed.

9.2 Dimensionality reduction

The proposed framework is evaluated also for a general reduction problem where there is no constraints about the number of dimensions in the new space. Face recognition/detection [11, 2] and optical character recognition [7] are few of the applications where there is no constraint about the number of dimensions in the new space.

Figure 12 shows the mean square error approximation versus the required area for data that belong to the 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 half the hardware resources than the designs produced by the reference algorithm, achieving at the same time the same mean square approximation error.

10 Conclusions

This paper proposes a novel Bayesian factor analysis framework for dimensionality reduction implementation 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 heterogeneity exploration of modern FPGAs under a unified framework. It has been demonstrated that by injecting information to the system regarding the area 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 area requirement when they are compared against current techniques, achieving at the same time the same error in the approximation of the data. Moreover, the proposed framework can be easily adapted to accommodate other optimization directions as power or speed, by incorporating the relevant information through the prior distribution of the basis Λ. It should be noted that the amount of the achieved gain using the proposed framework depends on the input data. However, under a uniform prior distribution for the coefficients (setting $a = 0$ in (20)), the proposed framework will perform no worse than the conventional techniques. 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, where the heterogeneous components can be also configured, adding an extra dimension to the problem of mapping dimensionality reduction designs into FPGAs.

Acknowledgement

This work was funded by the UK Research Council under the Basic Technology Research Programme "Reverse Engineering Human Visual Processes" GR/R87642/02.

References

[1] http://www.xilinx.com.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.

[3] C.-S. Bouganis, P. Y. K. Cheung, and G. A. Constantinides. Heterogeneity Exploration for Multiple 2D Filter Designs. In *Proc. Field Programmable Logic and Applications*, pages 263–268, 2005.

[4] C.-S. Bouganis, G. A. Constantinides, and P. Y. K. Cheung. A Novel 2D Filter Design Methodology For Heterogeneous Devices. In *Proc. Field-Programmable Custom Computing Machines*, pages 1–10, 2005.

[5] A. Dempster and M. D. Macleod. Use of minimum-adder multiplier blocks in FIR digital filters. *IEEE Trans. Circuits Systems II*, 42:569 – 577, September 1995.

[6] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

[7] H.-C. Kim, D. Kim, and S. Y. Bang. A numeral character recognition using the pca mixture model. *Pattern Recogn. Lett.*, 23(1-3):103–111, 2002.

[8] I. Koren. *Computer Arithmetic Algorithms*. New Jersey: Prentice-Hall Inc., 2nd edition, 2002.

[9] R. Pasko, P. Schaumont, V. Derudder, S. Vernalde, and D. Durackova. A new algorithm for elimination of common subexpressions. *IEEE Transactions on computer-aided design of integrated circuit and systems*, 18(1):58–68, January 1999.

[10] J. Taur and C. Tao. Medical image compression using principal component analysis. In *International Conference on Image Processing*, volume 2, pages 903–906, 1996.

[11] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:72–86, 1991.