HETEROGENEITY EXPLORATION FOR MULTIPLE 2D FILTER DESIGNS

Christos-S. Bouganis, Peter Y.K. Cheung, and George A. Constantinides

Department of Electrical & Electronic Engineering,
Imperial College London,
Exhibition Road, London SW7 2AZ, U.K.
email:christos-savvas.bouganis@imperial.ac.uk

ABSTRACT

Many image processing applications require fast convolution of an image with a set of large 2D filters. Field - Programmable Gate Arrays (FPGAs) are often used to achieve this goal due to their fine grain parallelism and reconfigurability. This paper presents a novel algorithm for the class of designs that implement a convolution with a set of 2D filters. Firstly, it explores the heterogeneous nature of modern reconfigurable devices using a Singular Value Decomposition based algorithm, which orders the coefficients according to their impact to the filters' approximation. Secondly, it exploits any redundancy that exists within each filter and between different filters in the set, leading to designs with minimized area. Experiments with real filter sets from computer vision applications demonstrate up to 60% reduction in the required area.

1. INTRODUCTION

One of the fundamental operators in image processing is the two dimensional convolution. Early examples can be found in edge detection and smoothing operations, where the image is convolved with a specific 2D filter, or kernel, in order to produce the desired result. Early filter sizes used to be relative small, e.g. 3 3 pixels. In recent years, many image processing applications have appeared in the literature that require the use of much larger 2D filters. Moderate size examples can be found in face detection/recognition applications where kernels with size of 23 23 pixels are used, and some more extreme examples can be found in medical imaging where applications require kernels with size of up to 63 63 pixels. Moreover, it is usually required by the image processing algorithms to convolve the input image with a set of 2D filters rather than with a single filter [1], which increases the computational load. At the same time, real-time implementation is often required, making the use of hardware acceleration a necessity [1].

FPGAs are often used to achieve this goal due to their fine grain parallelism and reconfigurability. Modern FPGAs are heterogeneous devices, often targeting the DSP community, and thus providing a mixture of resources that can be used in DSP applications. The two main coarse grain cores that are usually included in the recent devices are embedded RAMs and embedded multipliers. The former provides fast localized memory access, while the latter provides high speed accurate multiplication.

Existing techniques for 2D filter optimization for a modern reconfigurable device, such as word-length optimization [2] and Singular Value Decomposition [3], do not take into account the heterogeneity of the device. However, research concerning the exploitation of heterogeneity for a particular application has recently started to appear in the literature. In [4], the authors propose an approach for exchanging embedded RAMs for multipliers, whereas in [5] the author proposes an algorithm that identifies part of the circuit that can be implemented in embedded RAMs.

In our previous work [6, 7] we addressed the problem of allocating in an efficient way the different resources of a modern reconfigurable device for implementation in an FPGA of a 2D convolution with a single 2D filter. However, most modern image processing algorithms require the convolution of the input image with a set of 2D filters rather with a single filter. The motivation behind this work is to explore the redundancy that is implied in the approximation of the filters as a set for 2D convolution using an FPGA, which according to the authors' knowledge, has not yet been addressed in the community.

The proposed algorithm extents our previous work to target the implementation of a set of 2D filters, by providing an approach that makes explicit use of the heterogeneity of the device, targeting designs that use less area, and exploiting any redundancy that exists within each filter and between different filters in the set. The novel contributions of this paper are:

- Using an extension of Singular Value Decomposition to approximate a set of 2D filters with a number of two 1D convolutions and a set of low complexity 2D convolutions, reducing the number of high precision multipliers required for implementation. Moreover,

0-7803-9362-7/05/$20.00 ©2005 IEEE

the proposed technique exploits any existing redundancy within each filter and between different filters of the input set, leading to designs with minimized resource requirements.

- A resource allocation algorithm that maps the decomposed filter design onto a given set of heterogeneous resources on an FPGA, including dedicated multipliers and LUTs, in order to minimize resource usage. The proposed method shows a significant reduction in used resources.

The paper is organized as follows. Section 2 describes the current related work regarding 2D filter designs for heterogeneous devices. High level and detailed descriptions of the proposed algorithm are given in Section 3. Section 4 focuses on the evaluation of the algorithm, and Section 5 concludes the paper.

2. RELATED WORK

The paper focuses on the case where designs with high throughput are required, but design latency is of secondary importance. For this reason, only pipelined techniques for implementation of a 2D convolution filter are considered.

Current design methods for implementing a 2D filter in an FPGA have two main stages. In the first stage, the filter's coefficients are approximated using finite word-length precision through optimization of an objective function. Depending on the application, the objective function can be the mean square error or the maximum absolute error and, can be applied in the spatial or frequency domain [8]. In the second stage, the constant coefficient multiplications are often implemented as shift/add combinations using 4-LUTs and, to further optimize the design, the coefficients are sometimes transformed using *canonic signed digit* recoding, which reduces the required logic [9]. Canonic signed digit recoding represents the coefficients in a way such that high-speed low area multiplication can be achieved. Techniques to further reduce the required resources when multiplying by several coefficients can also be found in the literature [10].

Another technique exploits the potential separability of a 2D filter into a set of two 1D filters employing the Singular Value Decomposition (SVD) [3] to express the original filter as a linear combination of separable filters. Using this technique, the initial filter can be implemented as a set of 1D filters where half of them are applied to the rows of the image, and the other half to the columns, leading to designs with fewer multiplications. The resulting coefficients are quantized to finite word-length.

In our previous work [6, 7], we proposed an algorithm to optimize a 2D convolution filter implementation in a heterogeneous device, given a set of constraints regarding the number of embedded multipliers and slices. The algorithm estimates an approximation of the original 2D filter which minimizes the mean square error and at the same time meets the user's constraints on resource usage.

In this paper, an extension to this algorithm is presented, which targets applications that require convolutions with a set of 2D filters rather than with a single 2D filter. The algorithm makes explicit use of the heterogeneity of the device to minimize area, and exploits any redundancy that exists within each filter and between different filters in the set. The proposed method alters the structure of the original filters in order to find a structure that can be mapped in a more efficient way to the targeted device. The exploration of the design space is performed at a higher level than the word-length optimization methods or methods that use common subexpressions to reduce the area, since they do not consider altering the computational structure of the filter. The proposed technique is thus complementary to these previous approaches.

3. ALGORITHM DESCRIPTION

The main idea behind the proposed method is to decompose a given set of 2D filters into a set of separable filters, which are common for all the given filters, and into a set of non-separable parts which are specific for each filter. The separable filters can potentially reduce the number of required multiplications from $m \quad n$ to $m+n$ for each filter with size $m \quad n$ pixels. The non-separable parts encode the trailing error of the approximation and still require $m \quad n$ multiplications. However, the coefficients are intended to need fewer bits for representation and therefore their multiplications are of low complexity. The decomposition that we are seeking should provide a ranking for the coefficients according to their impact in the overall filter approximation error. This ordering would enable the algorithm to assign the available resources in an optimum way. Moreover, the decomposition should remove any redundancy that exists within each filter and between different filters in the set leading to a more compact design in term of resource usage.

A 2D filter is called separable if its impulse response $f(n_1, n_2)$ is a separable sequence, *i.e.*

$$f(n_1, n_2) = f_1(n_1) f_2(n_2).$$

The important property is that a convolution with a separable filter can be decomposed into two one-dimensional convolutions.

In more detail, given a set of 2D filters $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_k$, the algorithm seeks a decomposition of the form given in (1), where $\hat{\mathbf{A}}_i$ is a linear combination of the separable filters $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_r$, and $\mathbf{E}_i$ is a non-separable 2D filter.

$$\mathbf{F}_i = \hat{\mathbf{A}}_i + \mathbf{E}_i \tag{1}$$

Since the $\mathbf{A}_j$ matrices are separable, they can be decomposed as $\mathbf{A}_j = \mathbf{u}_j \mathbf{v}_j^T$, where $\mathbf{u}_j$ and $\mathbf{v}_j$ are vectors. Thus,

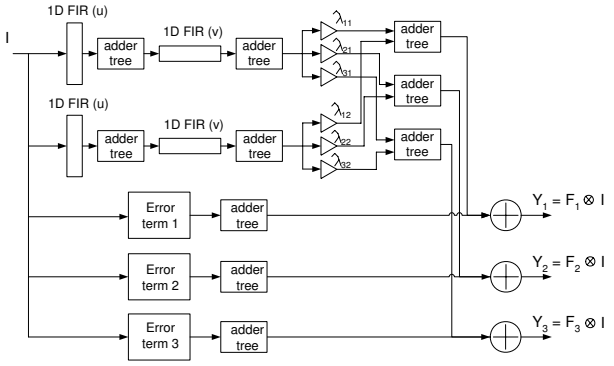


Fig. 1. Top level diagram

the approximation of each filter with r separable levels of decomposition is as follows.

$$\mathbf{F}_i = \sum_{j=1}^{r} \lambda_{ij} \mathbf{A}_j + \mathbf{E}_i = \sum_{j=1}^{r} \lambda_{ij} \mathbf{u}_j \mathbf{v}_j^T + \mathbf{E}_i \qquad (2)$$

The non-separable term $\mathbf{E}_i$ can be considered as the error term of the approximation of the filter from the separable set. This term is essential for achieving an adequate approximation of the filters using a relative small number of separable filters.

Given an input image $\mathbf{I}$ and a set of 2D filters $\mathbf{F}_i$, the resulting images of the convolutions are given by $\mathbf{Y}_i = \mathbf{I} \circledast \mathbf{F}_i$, where $\circledast$ denotes convolution. Using the filter decomposition in (2), the resulting images are given by:

$$\begin{aligned} \mathbf{Y}_i &= \mathbf{I} \circledast \left(\sum_{j=1}^{r} \lambda_{ij} \mathbf{A}_j + \mathbf{E}_i \right) \\ &= \sum_{j=1}^{r} \lambda_{ij} (\mathbf{I} \circledast \mathbf{u}_j) \circledast \mathbf{v}_j^T + \mathbf{I} \circledast \mathbf{E}_i \end{aligned} \qquad (3)$$

A top-level diagram of such design is illustrated in Figure 1. The figure shows a design regarding three filters, $k = 3$, and two decomposition levels, $r = 2$.

For a given set of filters there is an infinite number of possible decompositions described by (2). We are seeking the decomposition that has the minimum number of separable parts r for a given mean square error approximation, which corresponds to minimizing the number of required multipliers. Furthermore, the first few levels of the decomposition should have more impact to the mean square error approximation of the filters - a property that provides a ranking for the coefficients and is exploited by the proposed algorithm.

In the special case of one 2D filter, $k = 1$, the above decomposition can be achieved using the Singular Value Decomposition algorithm (SVD). The SVD algorithm decomposes a 2D filter into a linear combination of the fewest possible separable matrices [11]. In our case, we seek an extension of the SVD algorithm to many filters.

3.1. Rank-1 Decomposition algorithm

Under our framework, the problem can be formulated as follows. Given a set of m n matrices $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_k$, find the minimum set of rank-1 matrices $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_r$, such that linear combinations of them can span the matrices $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_k$ *i.e.*

$$\mathbf{F}_i = \sum_{j=1}^{r} \lambda_{ij} \mathbf{A}_j \qquad (4)$$

A matrix $\mathbf{A}$ is a rank-1 matrix if it can be decomposed as the product of two vectors *i.e.* $\mathbf{A} = \mathbf{u}\mathbf{v}^T$. The definition of the problem is the extension of a Singular Value Decomposition for one matrix to a collection of matrices. The special case where $r = 2$ is well researched [12]. The general case where $r > 2$ has been shown to be an NP-complete problem [13]. The proposed framework is based on the decomposition proposed in [14]. This algorithm is designed such that for the case of a single matrix ($k = 1$) the same decomposition as the Singular Value Decomposition algorithm is produced. This extends our previous work [6, 7] to the case of multiple matrices.

The rank-1 decomposition algorithm is given in Figure 2. It should be noted that the rank-1 decomposition algorithm is a greedy algorithm since previous selections of the rank-1 elements are not re-evaluated. Thus, the process is not guaranteed to converge to a global minimum. For more details and proof of convergence the reader is referred to [14].

3.2. Detailed Description of the Algorithm

Given a set of m n filters $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_k$, and a set of constraints regarding the available number of embedded multipliers and slices, the algorithm finds a design that minimizes the mean square error in the filters' approximation and meets the constraints on resource requirements. The algorithm can be divided into two main stages: the *slice allocation stage* and *multiplier allocation stage*.

3.2.1. Slice allocation stage.

In the slice allocation stage, the algorithm decomposes the input filters using the rank-1 decomposition algorithm and manifests all the constant coefficient multiplications using only slices. In the case of infinite precision, the matrices in (2) can be defined by one call of the rank-1 decomposition algorithm. However, due to the coefficient quantization in a hardware implementation, quantization error is inserted at

Algorithm: Decompose a set of m n 2D matrices $\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_k$ into rank-1 matrices $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_r$
FOR $j = 1 : r$
Calculate eigenvector $\mathbf{u}$ that corresponds to the largest eigenvalue of the n n matrix $\sum_{i=1}^{k} \mathbf{F}_i \mathbf{F}_i^T$
Form m k matrix $\hat{\mathbf{F}} = [\mathbf{F}_1^T \mathbf{u}, \mathbf{F}_2^T \mathbf{u}, \ldots, \mathbf{F}_k^T \mathbf{u}]$
Calculate the eigenvector $\mathbf{v}$ that corresponds to the largest eigenvalue of the m m matrix $\hat{\mathbf{F}}\hat{\mathbf{F}}^T$.
Repeat
Form the k n matrix $\hat{\mathbf{F}} = [\mathbf{F}_1 \mathbf{v}, \mathbf{F}_2 \mathbf{v}, \ldots, \mathbf{F}_k \mathbf{v}]^T$
Calculate the eigenvector $\mathbf{u}$ that corresponds to the largest eigenvalue of the n n matrix $\hat{\mathbf{F}}^T \hat{\mathbf{F}}$.
Form the m k matrix $\hat{\mathbf{F}} = [\mathbf{F}_1^T \mathbf{u}, \mathbf{F}_2^T \mathbf{u}, \ldots, \mathbf{F}_k^T \mathbf{u}]^T$
Calculate the eigenvector $\mathbf{v}$ that corresponds to the largest eigenvalue of the m m matrix $\hat{\mathbf{F}}\hat{\mathbf{F}}^T$.
until $\mathbf{u}$ and $\mathbf{v}$ vectors do not change significantly.
$\mathbf{A}_j = \mathbf{u}\mathbf{v}^T$.
$\lambda_{ij} = \mathbf{v}^T \mathbf{F}_i^T \mathbf{u}$.
Replace $\mathbf{F}_i$ with $\mathbf{F}_i$ $\lambda_{ij} \mathbf{u}\mathbf{v}^T$.
END

Fig. 2. Summary of the rank-1 decomposition algorithm

each level of the decomposition. The algorithm reduces the effect of the quantization error by propagating the error of each decomposition level to the next one and so on.

In more detail, the algorithm first determines the separable mask $\mathbf{A}_1 = \mathbf{u}_1 \mathbf{v}_1^T$ for the first level of the decomposition and the corresponding λ_{i1} for $i = 1, \ldots, k$ using the rank-1 decomposition algorithm. The algorithm quantizes the coefficients that correspond to the vectors $\mathbf{u}_1$ and $\mathbf{v}_1$ by using canonical signed digit recoding and using one non-zero bit per coefficient. Due to the fact that the coefficients for this level are quantized, the λ_{i1} terms are re-evaluated to correct for the quantization effects. The new λ_{i1} are calculated using the following system of linear equations $\mathbf{F}_i \mathbf{v}_1 = \lambda_{i1} \mathbf{u}_1$ and ${\mathbf{F}_i}^T \mathbf{u}_1 = \lambda_{i1} \mathbf{v}_1$ [11], where $\mathbf{u}_1$ and $\mathbf{v}_1$ have been quantized. The new λ_{i1} are quantized using the same quantization method as before.

The algorithm updates the input filters $\mathbf{F}_i$ as $\mathbf{F}_i \leftarrow \mathbf{F}_i$ $\lambda_{i1} \mathbf{u}_1 \mathbf{v}_1^T$, and repeats the above process until it estimates r levels of decomposition. The final error terms $\mathbf{E}_i$ are formed by the resulting filters $\mathbf{F}_i$ where all the coefficients are quantized as before.

Until now, the algorithm has assigned one non-zero bit per coefficient. In the next step, the algorithm allocates the remaining slices to the coefficients that have the largest impact to the mean square error approximation. A steepest descent optimization method is used, where all the possible coefficients are taken into consideration, and at each iteration of the steepest descent algorithm, the coefficient that minimizes the most the mean square error is selected to be updated. Moreover, for a given allocation of slices, the algorithm re-evaluates the whole decomposition from the beginning, since a change in a decomposition level affects the remaining levels of the decomposition. The first stage terminates when all the available slices have been used by the algorithm.

3.2.2. Multiplier Allocation stage.

In this stage, the algorithm determines the coefficients that will be placed to embedded multipliers. The coefficients that have the largest cost in terms of slices in the current design and reduce the filters' approximation error when are allocated to embedded multipliers, are selected. The second condition is necessary due to the limited precision of the embedded multipliers (18 bits in Xilinx devices), which in some cases may restrict the approximation of the multiplication and consequently of the final mean square error.

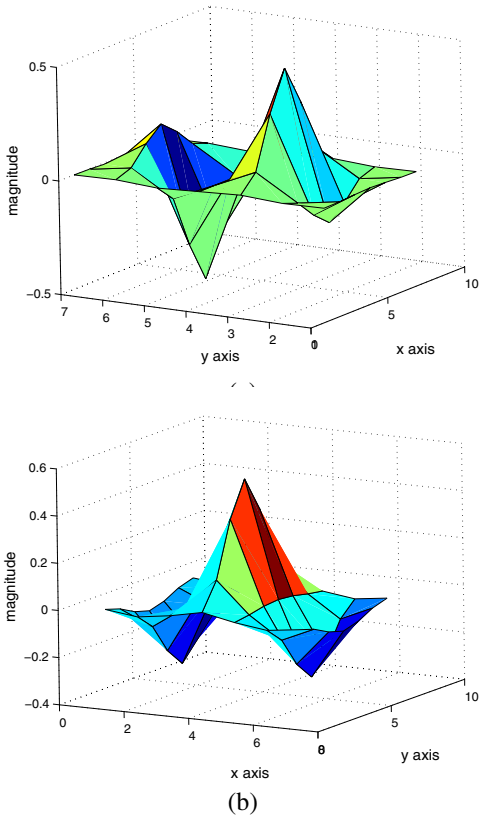
Finally, the algorithm allocates the slices that are available due to the allocation of the embedded multipliers to the rest of the design in order to achieve lower approximation error. This stage is similar to the slice allocation stage except that the coefficients that have been allocated to embedded multipliers are not considered for optimization.

3.3. Cost Model

In order to achieve an adequate estimation of the design's cost, accurate cost model for the constant coefficient multipliers and the adder trees are used. In all the cases, the exact precision and word-length of each register in the design is traced. Moreover, a fast algorithm for building an adder tree has been implemented, that minimizes the slices required through an intelligent selection of registers. The truncation of data path at different points in the design has been parameterized and can be set by the user. It should be noted that the cost calculation is specific for Xilinx devices where two one-bit full adders can be implemented in one slice, but it can be easily extended to different devices.

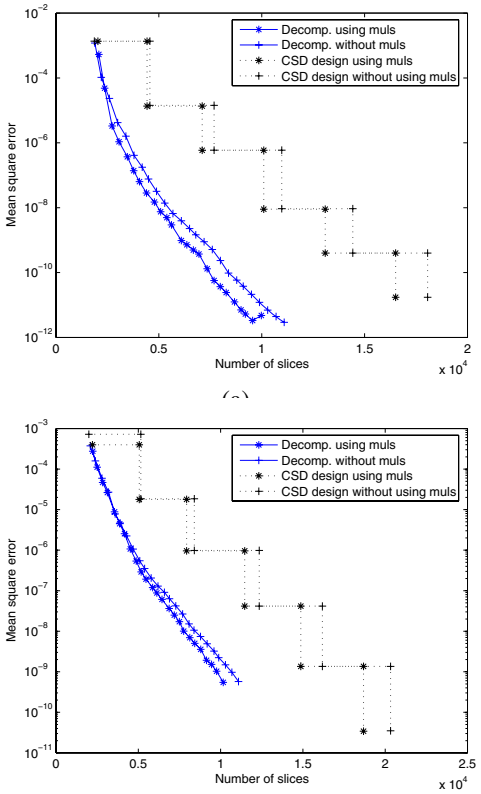
4. PERFORMANCE EVALUATION

The proposed algorithm is targeted to real computer vision applications. Thus, filters that are used in real computer vision applications are used to assess the performance of the algorithm. Two sets of filters are considered. The first set contains four 7 7 Gabor filters. This type of filters is selected because of their extensive use in computer vision applications. A convolution with Gabor filters yields images which are locally normalized in intensity and decomposed in terms of spatial frequency and orientation [15]. The Gabor kernels that are used are defined as $F(x, y) = \alpha \sin\theta \exp\left(\rho^2 \left(\frac{\alpha}{\sigma}\right)^2 \right)$, where $\rho^2 = x^2 + y^2$ and $\theta =$



(b)

Fig. 3. Impulse response of (a) a Gabor kernel and (b) a kernel from the steerable pyramid.



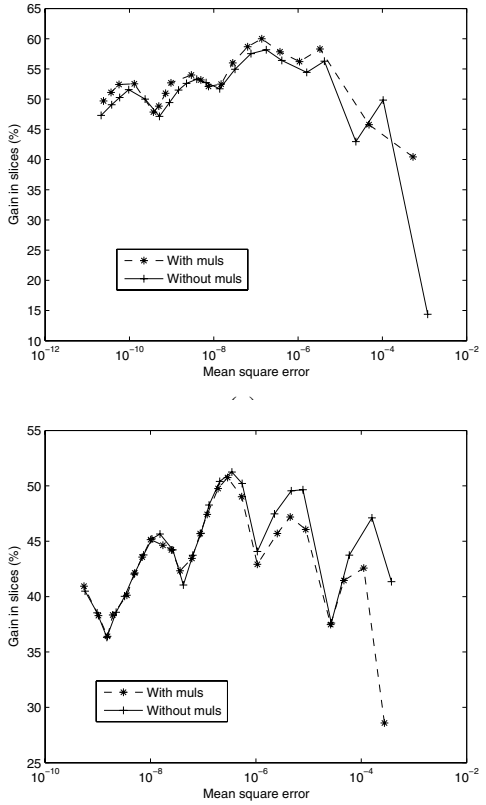
(b)

Fig. 4. Approximation mean square error versus number of slices using the proposed decomposition method and a reference algorithm for (a) four Gabor kernels (7x7) and (b) for four kernels from the steerable pyramid (7x7).

$[\alpha x, \alpha \quad \overline{2}(x+y), \alpha y, \alpha \quad \overline{2}(y \quad x)]$. The variance parameter, σ^2, controls the width of the Gaussian envelope and α controls the spatial frequency. Figure 3(a) illustrates a Gabor kernel for $\sigma = 6$ and $\alpha = 4$. The second set that is used for the evaluation procedure is taken from [1]. The authors use a set of $15 \quad 15$ bandpass filters to decompose an image into scale and orientation selective channels for scene analysis. The size of the proposed filters and the limited resources of the available target device lead the authors to truncate the filters to $7 \quad 7$ and to process each frame three times in order to calculate all the filter responses. In our experiment, four out of the eight filters were used. Figure 3(b) illustrates one filter of the set. Figure 4 shows the achieved mean square error in the approximation of the two above filter sets in relation with the number of required slices, when 20 embedded multipliers are used. The proposed method is compared against a reference algorithm that uses canonic signed digit recoding for the coefficients using the same number of non-zero bits for all them. Different points in the design space correspond to designs with different numbers of non-zero bits in canonic signed digit recoding. Two cases are illustrated in each graph. The first pair of lines corresponds to the case where the embedded multipliers have not been allocated to the coefficients, where the second pair of lines corresponds to a decomposition using the embedded multipliers. It can be seen that in both cases the proposed method leads to designs that use less area for a given error in the filters' approximation. Figure 5 demonstrates the achieved percentage gain in slice reduction between the proposed decomposition algorithm and the reference algorithm. Depending on the required approximation error, different levels of reduction can be achieved with a maximum of 60% for the set of Gabor filters and 51% for the set of steerable kernels. The achieved level of area reduction by the proposed algorithm depends on the nature of the filters, but it always performs better than the current techniques.

5. CONCLUSION

This paper presents a novel 2D filter design methodology for heterogeneous devices, by exploring the computational structure of the filter according to the different types of available resources in the device. Furthermore, it extends previ-



(b)

Fig. 5. Achieved percentage gain in slices for (a) four Gabor kernels (7x7) and (b) for four kernels from the steerable pyramid (7x7).

ous work to designs with multiple filters by exploiting any redundancy that exists within each filter and between different filters in the set. Experiments with filter sets from computer vision applications demonstrated a reduction of up to 60% in the required area. Future work will involve the use of word-length optimization techniques [2] and the use of current techniques for high-speed multiplication [10] to further enhance the algorithm's performance.

Acknowledgement

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

6. REFERENCES

[1] C.-S. Bouganis, P. Y. K. Cheung, J. Ng, and A. A. Bharath, "A Steerable Complex Wavelet Construction and its Implementation on FPGA," in *Proc. Field Programmable Logic and Applications*, September 2004.

[2] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, "Wordlength Optimization for Linear Digital Signal Processing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 10, October 2003.

[3] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*. Cambridge University Press, 1992.

[4] G. Morris, G. A. Constantinides, and P. Y. K. Cheung, "Migrating Functionality from ROMs to Embedded Multipliers," in *Proc. Field-Programmable Custom Computing Machines*, April 2004, pp. 287–288.

[5] S. Wilton, "SMAP: Heterogeneous Technology Mapping for Area Reduction in FPGAs with Embedded Memory Arrays," in *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, February 1998, pp. 171–178.

[6] C.-S. Bouganis, G. A. Constantinides, and P. Y. K. Cheung, "A Novel 2D Filter Design Methodology," in *Proc. International Symposium in Circuits and Systems*, 2005, pp. 532–535.

[7] ——, "A Novel 2D Filter Design Methodology For Heterogeneous Devices," in *Proc. Field-Programmable Custom Computing Machines*, 2005, pp. 1–10.

[8] D. Kodek, "Design of Optimal Finite Wordlength FIR Digital Filters Using Integer Linear Programming Techniques," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 28, pp. 304 – 308, June 1980.

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

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

[11] G. Strang, *Introduction to Linear Algebra*, 3rd ed. Wellesley-Cambridge Press, 1998.

[12] J. Ja'Ja, "Optimal evaluation of pairs of bilinear forms," in *Proc. of the 10th Annual ACM Sym. of Theory of Computing*, 1978, pp. 173 – 182.

[13] J. Hastad, "Tensor rank is NP-complete," *Journal of Algorithms*, vol. 11, no. 4, pp. 644 – 654, 1990.

[14] A. Shashua and A. Levin, "Linear image coding for regression and classification using the tensor-rank principle," in *Computer Vision and Pattern Recognition Conference*, vol. I. Kauai, HI, USA: IEEE, December 2001, pp. 42 – 49.

[15] S. Gong, S. McKenna, and A. Psarrou, *Dynamic Vision: From Images to Face Recognition*, 1st ed. Imperial College Press, 2000.